A Comprehensive Analysis of Network Security Attack Classification using Machine Learning Algorithms

Abdulaziz Saeed Alqahtani[1], Osamah A. Altammami[2], Mohd Anul Haq[3]*
Department of Computer Science-College of Computer and Information Sciences, Majmaah University, Al Majmaah, 11952, Saudi Arabia[1, 2]
College of Business Administration, Majmaah University, Al Majmaah, 11952, Saudi Arabia[3]

***Abstract*—As internet usage and connected devices continue to proliferate, the concern for network security among individuals, businesses, and governments has intensified. Cybercriminals exploit these opportunities through various attacks, including phishing emails, malware, and DDoS attacks, leading to disruptions, data exposure, and financial losses. In response, this study investigates the effectiveness of machine learning algorithms for enhancing intrusion detection systems in network security. Our findings reveal that Random Forest demonstrates superior performance, achieving 90% accuracy and balanced precision-recall scores. KNN exhibits robust predictive capabilities, while Logistic Regression delivers commendable accuracy, precision, and recall. However, Naive Bayes exhibits slightly lower performance compared to other algorithms. The study underscores the significance of leveraging advanced machine learning techniques for accurate intrusion detection, with Random Forest emerging as a promising choice. Future research directions include refining models and exploring novel approaches to further enhance network security.**

***Keywords*—Machine learning; cyber security; intrusion detection; network security; cyber security**

I. Introduction

In recent years, cyber-attacks have become more sophisticated and frequent, posing significant challenges to cybersecurity efforts. As organizations increasingly rely on interconnected networks for their operations, they are exposed to a greater risk of malicious activities. Traditional security methods, such as firewalls and antivirus software, while still valuable, are struggling to keep pace with the evolving tactics of cybercriminals [1]. These attacks can take various forms, from relatively simple phishing emails to complex malware and DDoS attacks, resulting in operational disruptions, data breaches, and financial losses [2]. To effectively combat these threats, security professionals need to adopt more advanced techniques for threat detection and mitigation [3]. Machine learning algorithms offer a promising solution by leveraging data analysis to identify patterns and anomalies indicative of malicious activity [4]. By automating threat detection and response processes, ML can help organizations bolster their network security defenses in the face of evolving cyber threats.

*Corresponding Author

A. Research Objectives and Motivation

The main objective of this paper is to conduct a comprehensive examination of network security attack classification using ML algorithms. By exploring various ML techniques and evaluating their applicability to network security, the research aims to enhance precision and efficiency in identifying and categorizing network attacks [4]. The motivation behind this research lies in the critical need for adaptive and intelligent security measures to counter the dynamic tactics employed by cybercriminals [5].

B. Consequences of Cyber-Attacks

The introduction also underscores the significant consequences of successful cyber-attacks, ranging from financial losses to reputational damage and legal ramifications [6]. This [7] highlights the importance of enhancing security measures to safeguard sensitive data, ensure uninterrupted operations, and maintain trust in digital systems.

C. Transition to Proactive Security Strategies

Furthermore, the integration of ML into network security protocols facilitates a transition from reactive to proactive security strategies [8]. By preemptively addressing potential threats, organizations can enhance overall resilience and security posture.

This paper will include a detailed comparative analysis with state-of-the-art methods, including recent advancements in deep learning applied to intrusion detection. Additionally, recent research in deep learning for intrusion detection will be reviewed to identify advancements and opportunities for improvement. This comprehensive comparison will enhance the credibility and relevance of the research findings.

This study is structured to first explore the existing landscape of network security and the challenges posed by cyber-attacks. It will then delve into the application of ML algorithms in enhancing threat detection and response processes. Following this, the paper will evaluate the strengths and limitations of existing network intrusion detection systems, proposing innovative ML solutions to address emerging challenges. Finally, it will provide recommendations for developing stronger, more flexible, and smarter security systems to combat cyber threats effectively in today's digital age.

II. Related Works

This review of the existing literature offers an in-depth examination of the present state of research in the classification of network security attacks through the application of machine learning algorithms.

A. Network Security Attack Classification

Traditional cybersecurity methods rely on predefined rules and signatures to detect and mitigate threats, but they struggle to keep up with the rapidly evolving tactics of cybercriminals. This [9] limitation has prompted a shift towards more adaptive and intelligent systems, leading to the exploration of machine learning techniques. In their examination of machine learning algorithms, the focus is on their crucial role in intelligent data analysis and automation within the cybersecurity field [10]. They [11] highlight the ability of these algorithms to extract valuable insights from diverse cyber data sources, demonstrating their relevance in real-world scenarios and illustrating how data-driven intelligence contributes to proactive cybersecurity measures [12]. Furthermore, [13] their analysis explores current methodologies, their practical implications, and emerging research directions, aiming to provide a comprehensive understanding of the current state of machine learning in cybersecurity and its potential for transformative advancements in line with the goals of our research

B. Machine Learning in Network Security

Machine learning's role in network security extends far beyond just threat detection. It encompasses prevention, response, and recovery aspects as well. By leveraging machine learning, organizations can build systems that continuously adapt to emerging threats, effectively fortifying their defenses against evolving attack patterns [14]. This adaptability is particularly crucial in an environment where cyber threats are constantly evolving in sophistication and evasiveness.

Furthermore, a recent study introduces a comprehensive taxonomy of security threats, evaluating the potential of artificial intelligence (AI), including machine learning, to address a wide range of challenges. This study in [15] represents the first exhaustive examination of AI solutions across various security types and threats. It covers lessons learned, current contributions, future directions, open issues, and strategies for effectively countering advanced security threats [16]. This holistic approach underscores the significance of integrating machine learning techniques into network security frameworks to combat the diverse and evolving landscape of cyber threats effectively.

C. Existing Machine Learning Approaches

In addition to supervised learning methods like Support Vector Machines (SVM) and Random Forests, unsupervised learning approaches, particularly anomaly detection, have gained prominence in the realm of network security. Unlike supervised methods that rely on labeled datasets to classify attacks, anomaly detection techniques can identify deviations from normal network behavior without predefined attack signatures. This makes them particularly useful for detecting novel and previously unseen threats that may not be captured by traditional rule-based systems.

For instance, research conducted by [17] on intrusion detection exemplifies the application of machine learning in enhancing security measures. By leveraging machine learning algorithms, researchers have demonstrated the effectiveness of these techniques in discerning malicious activities within network traffic. This study showcases the potential of machine learning to augment traditional security measures by providing a more adaptive and proactive approach to threat detection and mitigation [18].

Furthermore, the exploration of machine learning approaches in network security continues to evolve, with researchers investigating new algorithms and methodologies to address emerging challenges. As cyber threats become increasingly sophisticated and diverse, the integration of machine learning techniques holds promise for enhancing the resilience of network defenses and mitigating the impact of cyber-attacks.

D. Feature Extraction

The success of machine learning models in network security heavily relies on the selection and extraction of relevant features. Features can include traffic patterns, packet content, and behavioral analysis [19]. The process of feature selection is critical in optimizing the performance of the machine learning model, as irrelevant or redundant features can lead to decreased accuracy and increased computational overhead. Researchers in [20] have explored various feature selection techniques to identify the most informative features for attack classification. The study in [21] employs machine learning models and feature selection techniques to detect DDoS attacks in SDN, achieving optimal accuracy (98.3%) with KNN.

Feature engineering is a critical step in the data preprocessing pipeline, aimed at transforming raw data into a format that enhances the performance of machine learning models. It encompasses various techniques, including feature extraction and feature selection, to optimize the dataset for analysis. Given our dataset's high dimensionality with 49 features, effective dimensionality reduction was essential to streamline the analysis and mitigate computational complexity. To achieve this, we opted for PCA as a feature extraction technique. PCA transforms the original features into a reduced set of principal components, capturing the dataset's essential variance while preserving valuable information. Unlike feature selection techniques, which may exclude potentially informative features, PCA retains underlying patterns and structures in the data. This approach not only enhances computational efficiency but also maintains the integrity of the dataset. Explained variance analysis revealed that 10 principal components accounted for 90% of the dataset's variance, striking an optimal balance between variance coverage and computational complexity in our study.

E. Related Articles and Cybersecurity Majors

Table I shows summary of literature reviews, the table major drawback from previous, write their accuracy values.

TABLE I. LITERATURE REVIEW

Cite Key	Security Threat & Attacks	Detection & Mitigation	Incident Response	Standards & Policy	Important Findings
[22]	✓	✓	✓	✓	Early identification and detection of TTPs using supervised machine learning
[23]	✓	✗	✓	✓	Use ML & DL algorithms
[24]	✓	✓	✓	✓	highlights the ease with which DDoS attacks can be executed using a network of infected bots under the control of a single botmaster
[25]	✓	✓	✗	✗	addresses the significant security concern of email phishing attacks in cloud computing
[26]	✓	✗	✓	✗	attack taxonomy and threat model, organizations can enhance their ability to anticipate, detect, and respond to cyber threats
[27]	✓	✓	✓	X	Proposed ABRC exhibits significant performance improvement compared to existing deep learning techniques for cyber-attack detection
[28]	✓	✓	✓	✓	ML & QML for attacks; calculate precision and recall despite decreased accuracy post-attack; Inter-model susceptibility to crafted adversarial samples underscores the need for robust defense strategies. Future research will delve deeper into model performance and resilience against attacks
[29]	✓	X	X	✓	Developed technique achieves 99.7% accuracy in multi-class classification for intrusion detection, surpassing existing algorithms significantly; Demonstrates the efficiency of auto-tuned hyper-parameters and dataset improvements in enhancing detection capabilities.
[30]	✓	✓	✓	✓	Intrusion detection model achieves 96.00% accuracy, outperforming other neural network models; Stable training and test times.Data transmission security performance shows over 80% data message delivery rate, less than 10% message leakage and packet loss rates, and stable average delay around 350 milliseconds. The model ensures high security and prediction accuracy, serving as an experimental basis for enhancing safety in smart city rail transit systems.

F. Research Gap Analysis

Addressing the identified research gaps holds paramount importance in advancing our understanding and fortification against privacy attacks in the realm of machine learning. Existing studies exhibit a propensity to focus on specific machine learning models, leaving a critical void in comprehending privacy threats across a broader spectrum of techniques. Furthermore, the proposed attack taxonomy provides a foundational framework, yet there exists a gap in grasping the nuanced impact of different adversarial knowledge levels on the severity of privacy attacks. Bridging this gap demands a contextual exploration of privacy attacks within real-world machine learning applications, considering the diversity of domains and their unique challenges. The scarcity of longitudinal studies underscores the need for a dynamic perspective, tracking the evolution of privacy attacks over time. Lastly, the burgeoning landscape of emerging machine learning paradigms, such as federated learning and edge computing, lacks adequate attention in current research, necessitating a focused effort to understand and mitigate privacy attacks in these evolving contexts. Addressing these gaps promises significant implications, fostering the development of more resilient privacy-preserving machine learning models, implementing enhanced security measures, and cultivating a holistic comprehension of privacy risks for the continued advancement of secure and ethical machine learning applications.

III. METHODS

The research methodology for this study adopts a quantitative approach leveraging empirical data to draw objective conclusions and make generalizations about the relationships between variables quantitative research is deemed suitable for this investigation as it enables the measurement and analysis of numerical data providing a statistical foundation for evaluating ML in network security attack classification.

A. System Design

The system design shown in Fig. 1, is designed to analyze network security attacks using a subset of the unsw nb 15 dataset it comprises two main steps aimed at enhancing the accuracy of attack detection in the initial step data preprocessing is conducted involving both standardization and normalization to ensure uniformity in the dataset given the datasets high dimensional nature some features may be irrelevant or redundant potentially impacting the accuracy of attack detection negatively to address this issue a feature selection process is implemented to identify and retain only the most relevant subset of features effectively eliminating useless and noisy elements from the multidimensional dataset moreover class imbalance is recognized as a potential challenge in the dataset to mitigate this specific measures are taken to balance the representation of different attack categories ensuring that the classifiers are trained on a more equitable distribution of data moving on to the second step various classifiers are trained using the selected and refined features these classifiers are designed to detect all categories of attacks thereby aiming for maximum accuracy in the identification of security threats the utilization of multiple classifiers allows for a comprehensive assessment of the dataset considering the nuanced characteristics of different attacks finally the model's performance is evaluated using key

metrics such as accuracy precision recall and f 1 score these measures provide a thorough understanding of how well the classifiers are performing in terms of correctly identifying and classifying network security attacks the combination of these performance metrics ensures a comprehensive evaluation taking into account various aspects of the model's effectiveness in summary the proposed methodology begins with meticulous data preprocessing addressing issues of standardization normalization and feature selection it then tackles the challenge of class imbalance before training classifiers to detect diverse attack categories the evaluation phase employs a set of performance metrics to gauge the overall effectiveness of the framework in accurately identifying and classifying network security threats this methodological approach provides a systematic and robust foundation for analyzing network security attack data.

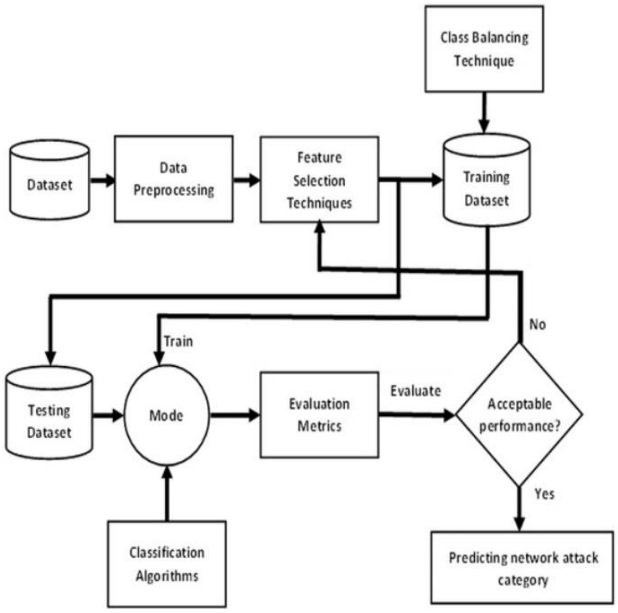

Fig. 1. System design.

B. Data Collection

1) Data sources: Network traffic and attack data will be sourced from Kaggle. This includes both real-world and simulated datasets to ensure a comprehensive evaluation.

The unsw nb 15 dataset crafted by researchers in 2015 stands as a comprehensive resource specifically tailored to address advanced network intrusion techniques comprising an extensive collection of 25 million records this dataset provides a rich and diverse landscape for the study of network security threats to the dataset encapsulates the complexity of modern cyber threats by encompassing 49 distinct features facilitating a nuanced analysis of network activity the 49 features in the dataset encapsulate various aspects of network traffic creating a multidimensional representation of cyber activities these features serve as essential variables for understanding and classifying different types of network security attacks researchers and practitioners benefit from the detailed and granular information embedded in each record enabling a thorough exploration of advanced intrusion techniques one noteworthy characteristic.

Nine different classes of attack families each representing a unique category of network security threat these classes encompass a wide spectrum of attack methodologies providing a holistic view of the diverse challenges faced in contemporary cybersecurity the dataset employs two label values for classification normal and attack enabling the categorization of network activities into either benign or malicious classes the dataset s utility extends shown in Fig. 2. The unsw nb 15 dataset serves as a vital resource in the field of cybersecurity research offering a rich and diverse collection of network activity records that enable in depth investigations into advanced intrusion techniques and the development of effective security solutions its comprehensive nature and well defined class structure make it an invaluable tool for researchers practitioners and educators alike in advancing the understanding and mitigation of network security threats.

2) Data preprocessing: Raw data will undergo preprocessing to handle missing values normalize features and address any anomalies this step is crucial for the effective application of machine learning algorithms.

a) Data standardizations: Data standardization, also known as data normalization, is a crucial preprocessing step in data analysis, particularly when working with machine learning algorithms sensitive to input feature scales. This process transforms the values of different variables to a common scale, ensuring that no particular feature dominates the learning process due to differences in their original scales. By rescaling the variables to have a mean of 0 and a standard deviation of 1, standardizing the data aids in maintaining consistency and improving algorithm performance. Formula is:

$$Z'' = X' - M' / \sigma'$$

Here:-

- Z" is the standardized value
- X' is the original value of the variable
- M' is the mean of the variable
- σ' is the standard deviation of the variable.

b) Data normalization: Data normalization is a preprocessing method employed to adjust numerical variables to a standardized range, usually between 0 and 1. This practice aims to ensure that all variables equally contribute to the analysis, preventing any single feature with larger magnitudes from dominating. One frequently used technique for normalization is min-max scaling, which involves a formula for normalizing a variable.

$$X\text{normalized} = X\text{max} - X\text{min} / X - X\text{min}$$

C. Machine Learning ML Classification Algorithm

Machine learning classification algorithms are computational tools created to classify input data into predefined categories or labels by analyzing their underlying patterns and characteristics. These algorithms learn from labeled training data, identifying patterns and relationships to predict the class labels of new instances. Various classification

algorithms, each with unique methodologies such as rule-based decision-making or probabilistic modeling, are utilized to effectively categorize data points into different classes. These algorithms are versatile tools used for tasks like detecting spam, recognizing images, and diagnosing medical conditions. Their performance is typically assessed using metrics such as accuracy, precision, recall, and F1 score, ensuring their efficacy across diverse applications.

This is a Classification problem where we want to detect whether there is an attack or not.

1) KNN: This is Ml algorithm proficient in both classification and regression assignments. Unlike traditional methods, KNN doesn't undergo a conventional training phase but rather memorizes the entire training dataset. During prediction, it relies on the proximity of data points within the feature space [31]. To classify a new data point, KNN computes distances, often employing Euclidean distance, from the point to all other instances in the training set. The k-nearest neighbors, identified by the smallest distances, then engage in a majority voting mechanism to allocate the class to the new data point. Alternatively, a weighted voting system can be utilized, granting closer neighbors greater influence. In regression duties, KNN forecasts the target value through averaging (or weighted averaging) the target values of the k-nearest neighbors. The selection of the hyper-parameter 'k' is pivotal, as it shapes the algorithm's sensitivity and generalization capability. KNN showcases its adaptability across various domains like image recognition and recommendation systems. Nonetheless [32], its performance hinges on meticulous 'k' selection, the choice of a distance metric, and understanding the dataset's traits. Employing efficient data structures such as KD-trees can enhance scalability, while thoughtful parameter tuning ensures its efficacy across diverse contexts.

2) Random forest: It is an ensemble learning method widely used for classification and regression tasks, particularly in intrusion detection, the algorithm operates through bootstrapped sampling creating diverse subsets of the dataset by randomly selecting instances with replacement and training individual decision trees on these subsets key to its robustness is the random select of features at each node split tree construction preventing overemphasis on specific features in classification random forest employs a majority voting mechanism aggregating predictions from multiple trees to make [33] the final decision this approach not only yields high accuracy but also enhances the models resilience to noise and variability the algorithm s adaptability and effectiveness make it a valuable tool in cybersecurity and various other domains.

3) Naïve bayes: Naive Bayes is a probabilistic classification algorithm based on Bayes theorem with the naive assumption of feature independence it s particularly effective for text classification and spam filtering the algorithm calculates the probability of a given instance belonging to a specific class by considering the conditional probabilities of each feature given the class despite its simplicity naive bayes often performs well and is computationally efficient the naive assumption simplifies calculations making it suitable for high dimensional datasets despite its success naive bayes might struggle with correlated features violating the independence assumption nevertheless its speed simplicity and respectable performance in various applications make it a popular choice for tasks involving categorical or text based data.

4) Logistic regression: It is used linear model for binary and multiclass classification problems despite its name. Also sigmoid the logistic function transforms the output into a range between 0 and 1 interpreting it as the probability of the positive class the algorithm optimizes its parameters through maximum likelihood estimation regularization techniques like l 1 or l 2 regularization can be applied to prevent overfitting logistic regression is interpretable and its coefficients provide insights into feature importance it s suitable for linearly separable problems but may struggle with complex relationships ensemble methods like random forest often outperform logistic regression on more intricate datasets but its simplicity interpretability and efficiency make it a valuable tool in various classification tasks.

D. Evaluation Metrics

Evaluation metrics serve as the compass for navigating the landscape of machine learning model performance accuracy the bedrock metric quantifies the models overall correctness precision zooms in on the models ability to avoid false positives while recall encapsulates its prowess in capturing all actual positive instances the f 1 score harmonizes precision and recall into a single metric striking a balance between precision oriented and recall oriented scenarios the confusion matrix a comprehensive tableau breaks down a models predictions into true positives true negatives false positives and false negatives these metrics collectively illuminate the multifaceted facets of a models effectiveness providing practitioners with a versatile toolkit to gauge and enhance performance across diverse applications shown in Fig. 2.

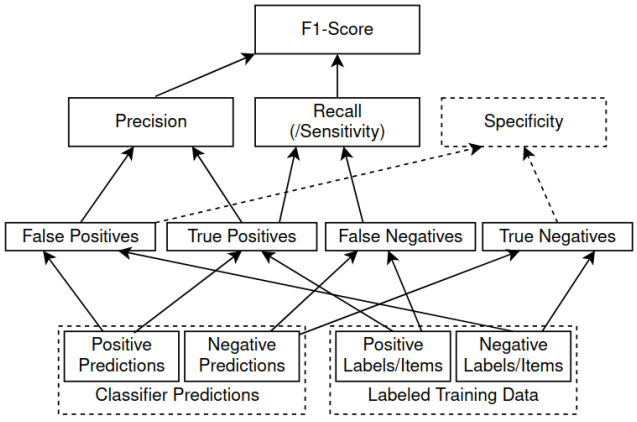

Fig. 2. Performance evaluation.

IV. IMPLEMENTATION

The experimental setup encompasses the selection and preparation of datasets the configuration of machine learning algorithms and the establishment of a controlled environment for rigorous testing the unsw nb 15 dataset consisting of 2 5 million records with 49 features was chosen for its relevance to advanced network intrusion techniques to ensure a diverse representation of attacks the dataset was partitioned into training and testing sets.

A. Tools and Techniques

This experimentation involved the implementation of various machine learning algorithms to evaluate their performance in network security attack classification python leveraging popular libraries such as scikit learn and tensor flow served as the primary programming language for algorithm implementation the choice of algorithms includes decision trees support vector machines neural networks and ensemble methods each configured with appropriate hyperparameters.

B. Implementation

The machine learning algorithms were implemented using a modular and scalable approach allowing for easy integration of new algorithms and flexibility in experimenting with different configurations the jupyter notebook was version-controlled using git to track changes and ensure reproducibility.

1) Import dataset: In the dataset preparation phase, the unsw nb 15 datasets were employed consisting of both training set unsw nb 15 training set csv and a testing set unsw nb 15 testings set csv the dataset was loaded into a python environment using the pandas library the training set as read from the unsw nb 15 training set csv file comprised 82 332 records while the testing set obtained from the unsw nb 15 testing set csv file included 175 341 records to verify the integrity of the dataset and ensure the appropriate division between training and testing data the lengths of the training and testing sets were checked the training set exhibited a length of 82 332 records and the testing set comprised 175.

2) Data visualization: The data visualization code utilizes the seaborn library to create informative plots depicting the distribution of attacks and normal traffic in both the training and testing sets the first two count plots in the top row display the overall distribution of labels attack or normal in the training and testing datasets meanwhile the bottom row illustrates the distribution of attack categories in both sets with the order specified based on the frequency of attack categories these visualizations provide a clear overview of the class distribution and the prevalence of different attack categories within the datasets such insights are crucial for understanding the imbalance between attack and normal instances and guide subsequent steps in the analysis such as addressing class imbalances and selecting appropriate evaluation metrics for machine learning models show in Fig. 3.

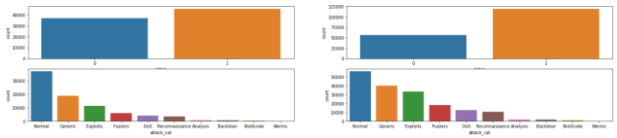

Fig. 3. Data visualization.

Next distribution of classes in the target variable showcases the balance or imbalance between normal and attack instances in Fig. 4. This is crucial for assessing the dataset's class distribution and potential class imbalance, which can impact machine learning model training.

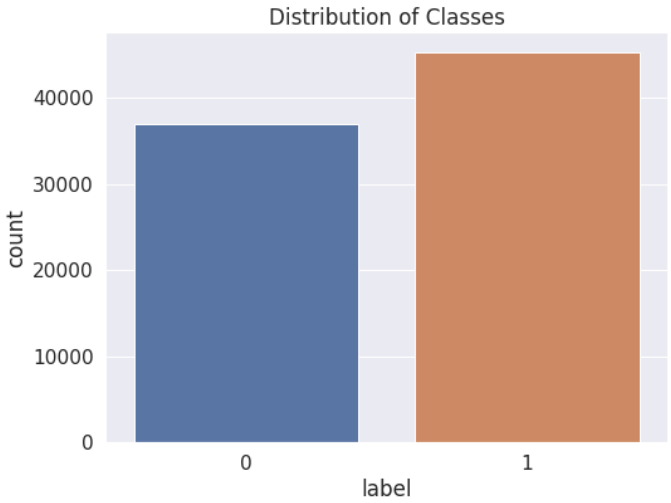

Fig. 4. Distribution of classes.

The below in Fig. 5, visualization presents a correlation heatmap, offering a comprehensive overview of the numerical features' relationships. This heatmap aids in identifying potential multicollinearity and understanding feature interdependencies.

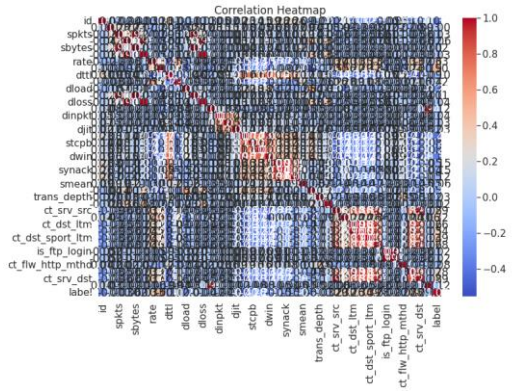

Fig. 5. Correlation of heat map.

In Fig. 6, a boxplot of the sttl feature is depicted showcasing its distribution across different classes this graphical representation allows for a quick assessment of the feature's potential discriminative power in distinguishing between normal and attack instances together these visualizations contribute to a holistic understanding of the dataset's characteristics guiding subsequent steps in the analysis and model development.

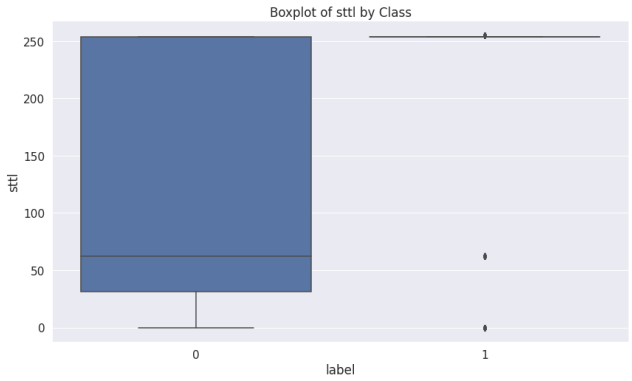

Fig. 6. Boxplot of sttl by class.

3) Data preprocessing: The data preprocessing phase involved a comprehensive examination and cleaning of the dataset the initial step focused on identifying and handling missing values and the results showed that there were no null values in any of the features this indicates a well maintained dataset without missing information ensuring the integrity of the subsequent analysis furthermore a closer look at categorical variables including proto service state and attack cat revealed the nature of these attributes the attack cat variable which represents the attack category is a crucial element for classification tasks the categorical variables were encoded appropriately for machine learning algorithms and their unique values and distribution were inspected this preprocessing step ensures that the dataset is ready for model training with categorical variables appropriately handled and missing values addressed the cleanliness and encoding of categorical variables contribute to the robustness of the subsequent machine learning analysis and enhance the interpretability of the results.

Also, in [36], data preprocessing focuses on numeric variables and involves a thorough exploration of statistical summaries initially it showcases a selection of numeric features from the dataset such as id dur spkts and others highlighting their characteristics subsequently statistical summaries are generated including count mean standard deviation minimum 25th percentile median 50th percentile 75th percentile and maximum values for each numeric variable this summary provides valuable insights into the distribution and variability of these features furthermore an additional exploration of the unsw nb 15 testings set csv file is conducted to understand its structure and dimensions revealing that it contains 1 000 rows and 45 columns this step is crucial for gaining an overview of the testing set which will be utilized in the subsequent stages of model evaluation.

4) EDA: The dataset comprising features related to network security attack classification was initially examined for its dimensions and the presence of relevant attributes descriptive statistics were computed to understand the distribution and variability of numeric variables and class distribution analysis provided insights into the balance between normal traffic and different attack categories correlation matrices were employed to explore relationships between features aiding in the identification of potential multicollinearity visualization of categorical variables and the distribution of attack categories within them further enriched our understanding of the dataset s structure the eda process also encompassed data cleaning and preprocessing steps addressing missing values and encoding categorical variables scatter plots and density plots were generated to visualize relationships between numeric features facilitating the detection of patterns shown in Fig. 7 and Fig. 8.

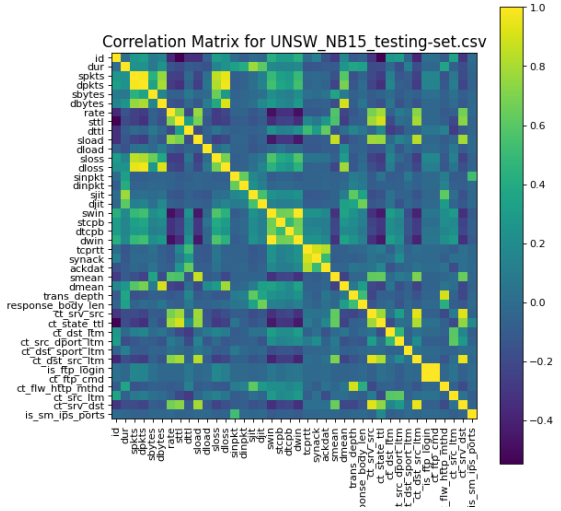

Fig. 7. Correlation matrix for test data.

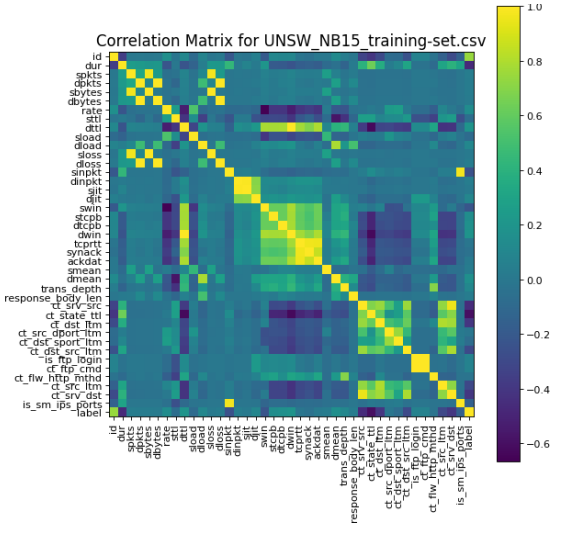

Fig. 8. Correlation matrix for train data.

5) Testing and training: The initial split was performed based on the index with the first 175 341 records designated for training and the remaining 82 332 records for testing the labels was extracted and assigned to y train and y test for training and testing respectively subsequently the label column was dropped from the feature sets to standardize the feature values a min max scaler was applied it s crucial to note that the scaler was fitted only on the training data to avoid data

leakage the training data x train was transformed using the fitted scaler and the testing data x test was scaled accordingly the final dataset dimensions were confirmed showcasing 175 341 samples for training each comprising 196 features and 82 332 samples for testing additionally categorical columns such as proto-state and service underwent one hot encoding for inclusion in the analysis these preprocessing steps ensure that the machine learning models are trained and tested on standardized and appropriately formatted data.

C. ML Model Classifications

1) Random forest: In Fig. 9, RF classification algorithm was implemented on a network security attack dataset, achieving an accuracy of 90%. The classification report details the model's performance in distinguishing normal and attack instances, with precision, recall, and F1 score metrics providing insights. For normal instances (label 0), precision and recall are 0.77 and 0.98, resulting in an F1 score of 0.86. For attack instances (label 1), precision, recall, and F1 score are higher at 0.99, 0.86, and 0.92 respectively. The weighted average F1 score is 0.90, indicating balanced performance. The confusion matrix shows the model correctly identifying 54,699 normal instances and 102,950 attack instances while misclassifying 1,301 normal instances as attacks and 16,391 attack instances as normal. Despite these misclassifications, the 90% accuracy highlights the random forest model's robustness in network security attack classification, contributing valuable insights to the research.

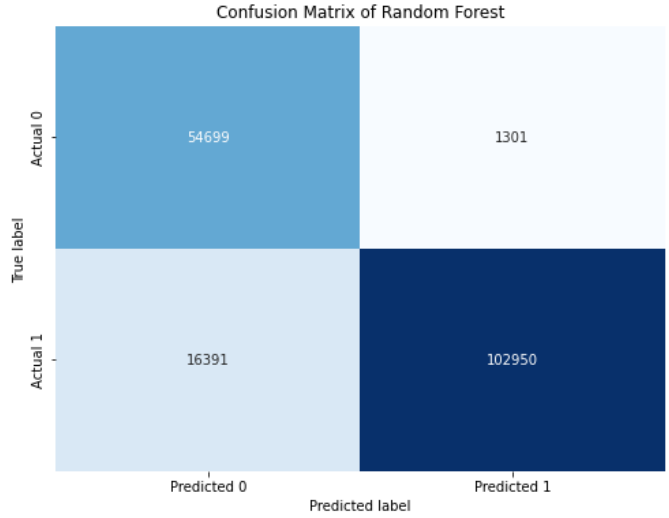

Fig. 9. Confusion matrix of random forest.

2) KNN: In Fig. 10, the K-Nearest Neighbors (KNN) classification algorithm was implemented with k=5 on the network security attack dataset, yielding an accuracy of 87%. The classification report provides detailed insights into the model's performance, showcasing its ability to discriminate between normal and attack instances.

For normal instances (label 0), the precision and recall are 0.72 and 0.96, respectively, resulting in an F1-score of 0.82. Similarly, for attack instances (label 1), the precision, recall, and F1-score are notably higher at 0.98, 0.83, and 0.90, respectively. The weighted average F1-score is reported as 0.87, indicating a balanced performance across both classes.

The confusion matrix further shows the classifier's effectiveness, correctly identifying 53,638 instances of normal traffic and 98,636 instances of attacks. However, the model misclassified 2,362 normal instances as attacks and 20,705 attack instances as normal. Despite these misclassifications, the overall accuracy of 87% underscores the robustness of the KNN model in distinguishing between benign and malicious network activities.

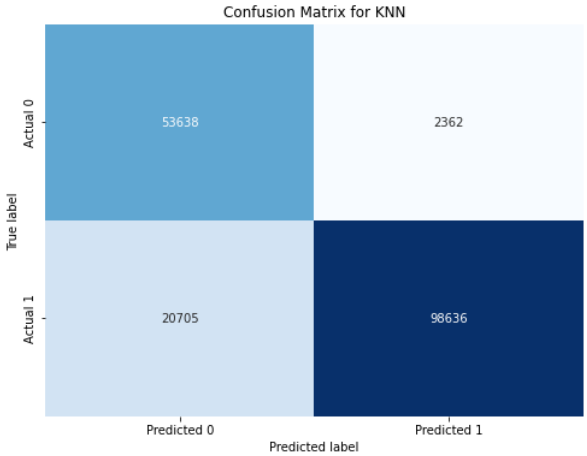

Fig. 10. Confusion matrix for KNN.

These findings contribute valuable insights to the research, emphasizing the efficacy of the K-Nearest Neighbors algorithm in the context of network security attack classification.

3) Naïve bayes: The Gaussian Naive Bayes classification algorithm was employed for network security attack classification, resulting in an accuracy of 79%. The classification report reveals that the model achieved a precision of 62% and recall of 87% for normal instances (label 0), yielding an F1-score of 0.72. For attack instances (label 1), the precision and recall are notably higher at 92% and 75%, contributing to an F1-score of 0.83. The weighted average F1-score is reported as 0.80, indicating a balanced performance across both classes shown in Fig. 11.

The confusion matrix provides additional insights, indicating that the model correctly identified 48,706 instances of normal traffic and 89,663 instances of attacks. However, there were misclassifications, with 7,294 normal instances being erroneously identified as attacks and 29,678 attack instances mistakenly labeled as normal. Despite these challenges, the Gaussian Naive Bayes algorithm demonstrates a commendable accuracy, emphasizing its suitability for network security attack detection in this context.

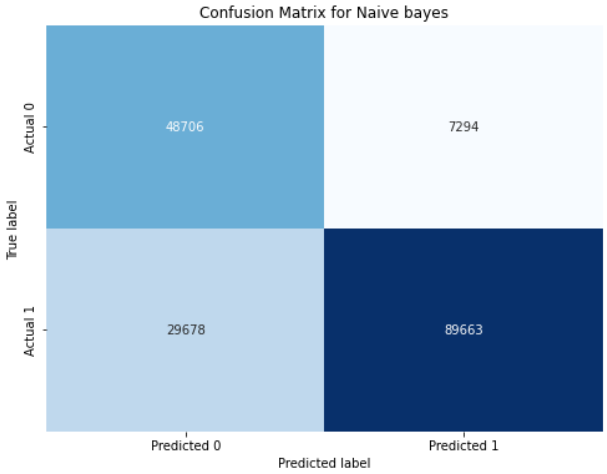

Fig. 11. Confusion matrix for naive bayes.

This analysis in Fig. 12 contributes valuable findings to the research, highlighting the strengths and limitations of the Gaussian Naive Bayes classifier in the domain of network security.

4) Logistic regression: The Logistic Regression classifier was implemented for network security attack classification, yielding an accuracy of 87.25%. The precision, recall, and F1-score for normal instances (label 0) are 74%, 92%, and 82%, respectively. For attack instances (label 1), the classifier achieved higher precision (96%) and slightly lower recall (85%), resulting in an impressive F1 score of 90.06%. The weighted average F1 score stands at 88%, indicating a balanced performance across both classes.

The confusion matrix and classification report provide detailed insights into Fig. 12, the classifier's performance. It correctly identified 51,592 instances of normal traffic and 101,719 instances of attacks. However, there were misclassifications, with 8,408 normal instances being erroneously identified as attacks and 17,622 attack instances mistakenly labeled as normal.

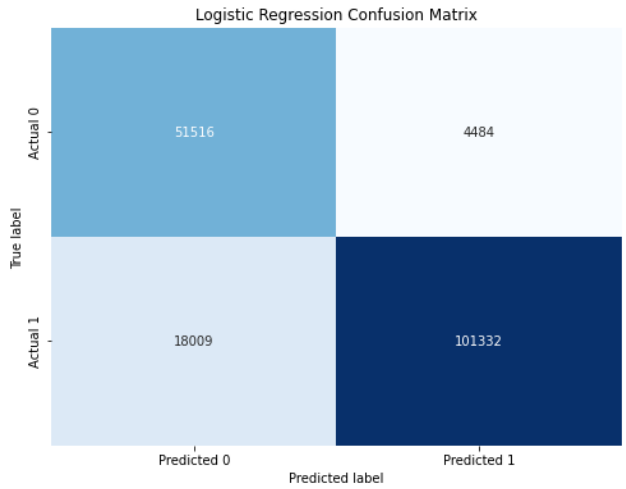

Fig. 12. Confusion matrix for logistic regression.

V. Results

The outcomes of the machine learning models including accuracy precision recall and f 1 score will be systematically analyzed a comparative study will be conducted to identify the algorithm that best suits the requirements of network security attack detection additionally insights gained from the analysis will be used to draw meaningful conclusions about the performance of each algorithm in handling diverse patterns present in the network traffic data.

TABLE II. Results of Classification Models without Feature Selection

Classifier	Accuracy	Precision	Recall	F1
Random Forest	0.90	0.92	0.90	0.90
K-Nearest Neighbors	0.87	0.90	0.87	0.87
Naive Bayes	0.79	0.83	0.79	0.80
Logistic Regression	0.87	0.85	0.96	0.90

In the result analysis Table II, the Random Forest classifier demonstrated superior performance with a high accuracy of 90%, effectively balancing precision and recall at 0.92 and 0.90, respectively. K-Nearest Neighbors (KNN) showcased strong predictive capabilities with an accuracy of 87% and a well-balanced precision-recall trade-off at 0.90 and 0.87. Naive Bayes exhibited a decent accuracy of 79%, with a precision of 0.83 and a balanced F1-Score of 0.80. Logistic Regression delivered an accuracy of 87%, with a commendable precision of 0.85 and a high recall of 0.96, resulting in a robust F1-Score of 0.90.

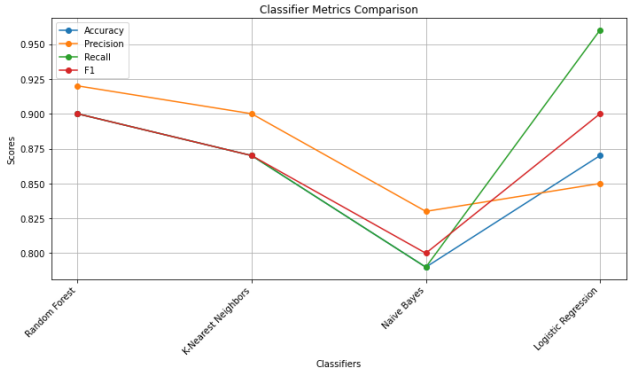

Fig. 13. ML classifiers metrics comparisons.

These classifiers play a crucial role in our network security context, offering effective means of identifying and classifying instances of security attacks based on the observed performance metrics in Fig. 13. The Random Forest model, in particular, emerges as a promising choice for its overall high accuracy and balanced precision-recall scores, making it well-suited for robust intrusion detection in network security applications.

A. Discussions

Building upon the result analysis the discussion section will explore the implications of the findings in the context of network security consideration will be given to the practicality efficiency and robustness of the algorithms the discussion will

also address potential challenges and limitations observed during the analysis providing a comprehensive perspective on the feasibility of deploying these algorithms in real-world scenarios furthermore comparisons with existing literature and benchmarks will be made to contextualize the significance of the results.

Deep learning has revolutionized intrusion detection, offering unparalleled accuracy and efficiency. In a study, [12] introduced the Principal Component-based Convolution Neural Network (PCCNN) approach for IDS, specifically targeting DoS and DDoS attacks on IoT devices. This approach boasts impressive accuracies of 99.34% for binary and 99.13% for multiclass classification on the NSL-KDD dataset. Utilizing a sophisticated architecture of 13 layers of Sequential 1-D CNN and feature reduction through Principal Component Analysis (PCA), it showcases exceptional promise for cutting-edge IoT intrusion detection.

Furthermore, the IDSGT-DNN framework, presented by [37], elevates cloud security by seamlessly integrating an attacker-defender mechanism using game theory and deep neural networks. This framework outperforms traditional methods in accuracy, detection rate, and various metrics on the CICIDS-2017 dataset. Remarkably, the defender's detection rate spans from 0 to 0.99, with gains strategically set at -5, 0, and 5. While the present study may not achieve the accuracies of the PCCNN approach (99.34% for binary and 99.13% for multiclass) and the IDSGT-DNN framework presented in previous works, it excels in computational efficiency. Our machine learning classifiers—Random Forest (RF) with an accuracy of 0.90, K-Nearest Neighbors (KNN) at 0.87, Naive Bayes with 0.79, and Logistic Regression (LR) also at 0.87—demonstrate competitive performance. Importantly, these classifiers deliver these results in significantly less time, underscoring the trade-off between accuracy and computational speed in intrusion detection systems.

Additionally, promising results in the random forest model showcased notable improvements achieving a commendable balance between precision and recall k nearest neighbors demonstrated strong predictive capabilities aligning with its suitability for identifying patterns in network traffic although naive bayes presented a lower accuracy its performance remains consistent with the algorithm s inherent assumptions logistic regression emerged as a reliable choice showcasing a balanced precision recall trade off collectively our findings contribute to the existing body of research by highlighting the effectiveness of these classifiers in the specific context of intrusion detection offering valuable insights for the development of robust and accurate network security systems.

The performance of the proposed methodology will be compared with existing approaches, highlighting the advancements achieved in Table III.

TABLE III. COMPARISON WITH EXISTING APPROACHES

Paper	Classifiers	Accuracy	Precision	Recalls
[34]	SGD	80%	82.1%	82.1%
This study	Random forest	90%	90.2%	90%
[35]	Neural network	87%	87.2%	87.8%
[3]	XGBoost	88%	88.3%	88.8%
[8]	SVM	76%	77%	77%
[23]	Random forest	80%	81%	8.9%
[14]	KNN	82%	82%	82%

B. Limitations

Though our study presented promising results, it is crucial to recognize the limitations. The effectiveness of machine learning models heavily relies on the dataset's quality and representativeness. The utilization of the unsw nb 15 dataset in our research may not adequately cover all real-world network traffic scenarios and variations. The chosen features and preprocessing techniques could impact model performance, suggesting further exploration of feature engineering methods to improve classifier efficacy. The selection of classifiers was based on established algorithms, but future research could investigate new approaches or DL methods for better outcomes. Evaluation metrics mainly focused on accuracy, precision, recall, and f1 score, potentially overlooking variations in performance among different attack types. These restrictions highlight the importance of continuous refinement and exploration in intrusion detection to combat evolving cyber threats effectively.

The ML models used in the present investigation have been practically implemented and tested using the real intrusion detection dataset, which is recognized for its relevance to real-world network intrusion scenarios. This approach leverages the dataset to demonstrate the models' practical applicability in a real-world network environment. By conducting experiments on the dataset, the effectiveness of the models in detecting a variety of attacks, including novel and sophisticated ones, was evaluated. This hands-on validation allows for the identification of operational challenges and fine-tuning of the models for improved performance in real-world scenarios. The practical testing provides valuable insights into the models' robustness, scalability, and applicability, thereby reinforcing their effectiveness and reliability in real-world network intrusion detection applications. Future research could consider novel approaches or DL methods for better results [38]. Evaluation metrics focused on overall accuracy, precision, recall, and f1 score, neglecting performance variations across different attack types. These limitations highlight the importance of continuous refinement and exploration in intrusion detection to address evolving cyber threats.

VI. CONCLUSIONS

The detailed examination and discussion of the outcomes provide valuable insights into the effectiveness of different machine learning classifiers for detecting intrusions in network security. The Random Forest classifier showed the best performance, with high accuracy, precision, recall, and F1 score. K-Nearest Neighbors and Logistic Regression also had good results, while Naive Bayes had a slightly lower performance. These results highlight the importance of using advanced machine-learning techniques for accurate intrusion detection. Choosing the right algorithm based on the specific characteristics of the cybersecurity task is crucial. However, it's important to recognize the limitations and future research should focus on improving models, exploring new approaches, and incorporating more data to enhance the strength and applicability of intrusion detection systems. In conclusion, this study adds to the conversation on strengthening cybersecurity defenses through machine learning methods. In concluding this study, it is essential to highlight future research possibilities for advancing intrusion detection and network security. One potential avenue is to enhance existing models through hyperparameter tuning and ensemble methods. Moreover, utilizing diverse datasets could broaden the adaptability of models to various network scenarios. Exploring deep learning approaches like neural networks could help uncover complex patterns in network traffic data. Additionally, addressing limitations like dataset reliance and biases may require more comprehensive datasets and real-world scenarios for model validation. Lastly, research could focus on creating hybrid models that combine multiple classifiers' strengths for increased resilience.

REFERENCES

[1] S.-W. M. M. S. R. A. M. R. M. M. a. M. H. Lee, "Towards secure intrusion detection systems using deep learning techniques: Comprehensive analysis and review," Journal of Network and Computer Applications 187, 2021.

[2] H. a. G. K. Alqahtani, "Machine learning for enhancing transportation security: A comprehensive analysis of electric and flying vehicle systems.," Engineering Applications of Artificial Intelligence 129, 2024.

[3] T. S. S. C. D. T. D. C. a. M. A. K. Saranya, "Performance analysis of machine learning algorithms in intrusion detection system: A review," Procedia Computer Science 171, 2020.

[4] M. D. M. a. K. R. Karthikeyan, "Firefly algorithm based WSN-IoT security enhancement with machine learning for intrusion detection," Scientific Reports 14, no. 1, 2024.

[5] X. X. Z. Z. Y. L. Y. B. Z. Q. L. a. X. L. Xiao, "A comprehensive analysis of website fingerprinting defenses on Tor," Computers & Security 136, 2024.

[6] K. M. M. M. K. F. K. a. I. G. Aygul, "Benchmark of machine learning algorithms on transient stability prediction in renewable rich power grids under cyber-attacks," Internet of Things 25, 2024.

[7] G. MeeraGandhi, "Machine learning approach for attack prediction and classification using supervised learning algorithms.," Int. J. Comput. Sci. Commun 1, no. 2 (2010): 247-250, 2010.

[8] M. a. M. M. Zamani, "Machine learning techniques for intrusion detection.," arXiv preprint arXiv:1312.2177, 2013.

[9] Z. G. A. Y. Y. a. Y. L. Sun, "Optimized machine learning enabled intrusion detection 2 system for internet of medical things," Franklin Open 6, 2024.

[10] A. L. a. E. G. Buczak, "A survey of data mining and machine learning methods for cyber security intrusion detection," IEEE Communications surveys & tutorials 18, no. 2, 2015.

[11] X.-S. Y. S. S. N. D. A. Joshi, "Fourth International Congress on Information and Communication Technology," ICICT 2019, London, Volume 2, 2019.

[12] M. A. M. A. R. K. a. T. A.-H. Haq, "Development of PCCNN-Based Network Intrusion Detection System for EDGE Computing," Computers, Materials & Continua 71, no. 1, 2022.

[13] I. H. Sarker, "Machine learning for intelligent data analysis and automation in cybersecurity: current and future prospects," Annals of Data Science 10, no. 6, 2023.

[14] A. A. A. D. V. a. S. S. Mahfouz, ""Ensemble classifiers for network intrusion detection using a novel network attack dataset.," Future Internet 12, no. 11, 2020.

[15] M. S. T. Z. H. S. U. R. G. A. a. Z. H. A. Waqas, "The role of artificial intelligence and machine learning in wireless networks security: Principle, practice and challenges," Artificial Intelligence Review 55, no. 7, 2022.

[16] H. M. a. P. S. Prachi, "Intrusion detection using machine learning and feature selection.," International Journal of Computer Network and Information security 11, no. 4, 2019.

[17] A. a. M. A. R. Alotaibi, "Enhancing the Sustainability of Deep-Learning-Based Network Intrusion Detection Classifiers against Adversarial Attacks," Sustainability 15, no. 12 , 2023.

[18] D. M. A. F. A. A. R. a. R. M. M. Musleh, "Intrusion Detection System Using Feature Extraction with Machine Learning Algorithms in IoT.," Journal of Sensor and Actuator Networks 12, no. 2, 2023.

[19] M. A. a. M. A. R. K. Haq, "DNNBoT: Deep neural network-based botnet detection and classification.," Computers, Materials & Continua 71, no. 1, 2022.

[20] E. S. R. R. N. Z. A. A. A. H. J. M. N. S. S. M. I. E. a. B. A. M. Alomari, "Malware detection using deep learning and correlation-based feature selection," Symmetry 15, no. 1 , 2023.

[21] H. O. P. a. A. C. Polat, "Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models," Sustainability 12, no. 3 , 2020.

[22] M. H. U. R. S. A. R. M. A. R. F. R. a. I. A. Imran, "A performance overview of machine learning-based defense strategies for advanced persistent threats in industrial control systems.," Computers & Security 134, 2023.

[23] Z. M. M. I. a. M. N. H. Azam, "Comparative analysis of intrusion detection systems and machine learning based model analysis through decision tree.," IEEE, 2023.

[24] M. A. S. M. a. M. A. Al-Shareeda, "DDoS attacks detection using machine learning and deep learning techniques: Analysis and comparison," Bulletin of Electrical Engineering and Informatics 12, no. 2, 2023.

[25] U. A. R. A. H. A. S. M. B. A. a. A. A. Butt, "Cloud-based email phishing attack using machine and deep learning algorithm," Complex & Intelligent Systems 9, no. 3 , 2023.

[26] M. a. S. G. Rigaki, "A survey of privacy attacks in machine learning," ACM Computing Surveys 56, no. 4, 2023.

[27] B. M. M. AlShahrani, "Classification of cyber-attack using Adaboost regression classifier and securing the network," Turkish Journal of Computer and Mathematics Education (TURCOMAT) 12, no. 10, 2021.

[28] M. S. H. S. I. I. M. D. H. M. A. K. V. C. a. R. V. Akter, "Exploring the Vulnerabilities of Machine Learning and Quantum Machine Learning to Adversarial Attacks using a Malware Dataset: A Comparative Analysis," arXiv preprint arXiv:2305.19593 , 2023.

[29] H. Z. S. Y. C. a. H. B. Xu, "A data-driven approach for intrusion and anomaly detection using automated machine learning for the Internet of Things," Soft Computing 27, no. 19 (2023): 14469-14481., 2023.

[30] Z. X. X. L. C. S. S. a. Z. W. Wang, ""Intrusion detection and network information security based on deep learning algorithm in urban rail transit management system," IEEE Transactions on Intelligent Transportation Systems 24, no. 2, 2023.

[31] K. ". Alnowaiser, "Improving Healthcare Prediction of Diabetic Patients Using KNN Imputed Features and Tri-Ensemble Model.," IEEE Access, 2023.

[32] A. R. X. Y. C. a. M. G. Huang, "Research on multi-label user classification of social media based on ML-KNN algorithm.," Technological Forecasting and Social Change 188, 2023.

[33] B. D. J. A. a. S. H. L. He, "Assessment of tunnel blasting-induced overbreak: A novel metaheuristic-based random forest approach.," Tunnelling and Underground Space Technology 133 , 2023.

[34] G. a. G. K. Kocher, "Analysis of Machine Learning Algorithms with Feature Selection for Intrusion Detection Using UNSW-NB15 Dataset," Available at SSRN 3784406, 2021.

[35] M. S. L. a. K. G. K. Beechey, ""Evidential classification for defending against adversarial attacks on network traffic," Information Fusion 92 (2023): 115-126., 2023.

[36] G. MeeraGandhi, "Machine Learning Approach for Attack Prediction and Classification using supervised learning algorithms," Int. J. Comput. Sci. Commun 1, no. 2, 2010.

[37] E. Balamurugan, A. Mehbodniya, E. Kariri, K. Yadav, A. Kumar, and M. Anul Haq, "Network optimization using defender system in cloud computing security based intrusion detection system withgame theory deep neural network (IDSGT-DNN)," Pattern Recognit. Lett., vol. 156, pp. 142–151, 2022, doi: https://doi.org/10.1016/j.patrec.2022.02.013.

[38] H. Mohd Anul, "DBoTPM : A Deep Neural Network-Based Botnet," Electronics, vol. 12, no. 1159, pp. 1–14, 2023.