An Early Warning System for Proactive DDoS Attack Detection Using Ordinary Differential Equation-Based Scoring

Radhakrishna Vangipuram[1], Rajput Eswar Sai Singh[2], Ramesh Kumar Aytha[3],
Sravankiran Vangipuram[4]*, Sreenivasa Rao Annaluri[5]*
Department of Information Technology[1, 2, 4, 5]
Department of Electronics and Communication Engineering[3]
VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, Telangana, India[1, 2, 3, 4, 5]

***Abstract*—Distributed Denial of Service (DDoS) attacks are still a significant issue facing cybersecurity, flooding infrastructure with large packet torrents and disrupting networks across the globe. To mitigate against such volumetric attacks, the best approach is to intercept the threat long before any attack gets underway. However, conventional defense systems react primarily after the volume of attack peaks; thus, they leave critical assets vulnerable to disruption. To avoid this, it is essential to detect and analyze precursor signals within the attack preparation period. This work introduces a framework for generating an early warning signal based on computed scores and set thresholds to categorize network traffic alerts into low-, medium-, and high-impact alerts. In this study, a differential equation model is applied to the CICDDoS2019 dataset to generate these alert scores. The goal is to make a prediction well before the attack blows up in size and before the onset of the attack. The results exhibit an impressive capacity for proactive detection, from reactive mitigation to predictive defense. This framework was able to predict the intrusion with low, medium, and high alerts issued 46 minutes and 53 seconds, 46 minutes and 49 seconds, and 46 minutes and 55 seconds before the attack peak. The proposed framework obtained a balanced accuracy ranging from 99.43% to 99.70% for these three alert classifications.**

Keywords*—DDoS; early warning signals; ordinary differential equation; residual; low threshold; medium threshold; high threshold

I. Introduction

The rapid expansion of the Internet has introduced significant benefits to modern society and urban development, yet it has also created a complex network environment that is highly vulnerable to security threats, most notably Distributed Denial of Service (DDoS) attacks [9][10]. As network traffic increases sharply with the proliferation of IoT devices, traditional security measures often struggle with scalability and adaptability [7][8]. While paradigms like Software-Defined Networking (SDN) and Edge Computing offer improved resource management, they also introduce risks, such as centralised controllers becoming targets for malicious activity, potentially leading to single points of failure [4]. The evolution of DDoS attacks represents an ongoing challenge, with modern attacks reaching terabit-per-second traffic volumes, such as 1.7 Tb/s. Attacks of this magnitude can rapidly disrupt network services and pose serious risks to critical infrastructures, including healthcare systems and smart-city networks [4][9][10][12]. In this landscape, purely reactive defence is often insufficient; by the time anomaly-based detection flags an active flood, irreversible damage such as downtime or service degradation could have already occurred [8]. Consequently, there is a critical need for DDoS attack prediction, which identifies evidence of attack preparation before launch, providing network administrators with more time to implement countermeasures [1].

Early warning signals (EWS) for DDoS attacks can be found in precursor phases, and researchers distinguish these efforts across a 24-hour horizon, separating short-term predictions (seconds to hours) from long-term forecasts (days to months) [1]. Identifying these signals requires moving beyond simple signature-based methods to more intelligent, adaptive mechanisms. Recent advancements in machine learning (ML) and deep learning (DL) have opened new avenues for this purpose. Long Short-Term Memory (LSTM) networks are particularly effective for forecasting surges in daily attack activity by learning long-term temporal patterns in evolving traffic data [12][4]. The CoWatch framework leverages LSTM for collaborative prediction across distributed SDN controllers, utilising a Distributed Publish/Subscribe (DPS) mechanism to synchronise alerts efficiently [4]. Echo State Networks (ESN), integrated with nature-inspired algorithms like African Buffalo Optimisation (ABO-DT), facilitate detection and prediction in realistic multidimensional IoT environments [9]. Ensemble Online Machine Learning (OML) models, incorporating classifiers like BernoulliNB and SGD, enhance adaptability to concept drift and are resilient against zero-day and low-rate attacks [8]. Hybrid models, such as the CNN-Bidirectional LSTM for smart healthcare and Pyramid Depthwise Separable Convolution (PyConv) with Variational Gaussian Mixture (VGM) models, improve feature extraction and accuracy in cyber-physical systems [3][5]. Specialised implementations like the SS-LSACNN model address the unique resource constraints of Wireless Sensor Networks (WSN) [6].

Furthermore, classification frameworks optimised with Recursive Feature Elimination (RFE) and Grid Search on Random Forest algorithms have achieved near-perfect

*Corresponding author.

accuracy (99.99%) in identifying attack patterns [11]. Strategic deployment of these detection models at the network edge is a promising solution to minimise latency and computational overhead on centralised controllers [10]. While detection and mitigation strategies are becoming highly sophisticated, researchers emphasise that the field of prevention and forward-looking early warning remains an essential area for future development to protect against the spreading of malicious traffic [7].

II. Literature Survey

The survey by de Neira A. B. et al [1] provided a comprehensive classification of the state-of-the-art in DDoS attack prediction, categorising 27 shortlisted studies by time, architecture, methodology, and data type. It establishes a 24-hour frontier to distinguish between short-term predictions (made seconds to hours in advance) and long-term predictions (made days to months in advance). The work of He, J. et al [2] used a Pyramid Depthwise Separable Convolutional neural network intrusion detection system (PyDSC-IDS) model on NSL-KDD, UNSW-NB15, and CICIDS2017 datasets, achieving 81.63%, 80.63%, and 97.60% accuracy, respectively. The model effectively extracts multi-scale features and improves detection accuracy, especially for minority attack classes, compared to standard CNNs. Despite the reductions provided by DSC, the parameters and FLOPs are still considered "excessive" compared to some simpler models.

Zhou H. et al. [3] used Maximum Likelihood Estimation (MLE) as a feature selection method with the CoWatch Long Short-Term Memory (LSTM) model on the CIC-DDoS2019 dataset, which achieved a 3.10% false alarm rate in prediction and a 3.01% false alarm rate in detection. The CoWatch model allows distributed SDN controllers to collaboratively predict DDoS attacks before performance degradation occurs. The evaluation did not incorporate communication latency, as it varies significantly under different network conditions. Sharma et al. [4] proposed an efficient, lightweight hybrid CNN-Bidirectional LSTM model for smart healthcare networks. This model contributes by extracting temporal and spatial features from traffic flows to classify them as benign or malicious, specifically targeting DoS attacks in cyber-physical systems (CPS), and achieved 99% accuracy, 95.8% precision, with 98.9% detection rate on the BoT-IoT dataset.

V. K. Singh et al. [5] focused on WSN security using Soft Swish Linear Scaling-centred Adam Convolutional Neural Network (SS-LSACNN) for classification and Two's Complement Shift Reverse (TCSLR) for data prevention. It integrates fuzzy logic and neural networks to reduce energy usage and discover attacks rapidly. The survey by H. Wang and Y. Li [6] provided a systematic review of SDN architecture and the evolution of DDoS detection within this environment. It categorizes detection techniques into statistical analysis (entropy-based), machine learning, and deep learning, noting that machine learning-based methods have garnered more attention due to their accuracy. The review also points out that the field of DDoS prevention remains largely unexplored compared to detection and mitigation.

The work of A. A. Alashhab et al [7] utilised an Ensemble Online Machine Learning (OML) model with Chi-squared (Chi^2) as a feature selection method, which selected 14 features, on a Custom/proprietary dataset containing 1,45,614 instances and 22 initial features. The study reported 99.26% accuracy with 99.10% precision. The system effectively blocked malicious traffic with minimal impact on legitimate users. But it needs to be validated with real-world network topologies. The study by Rao G. S. et al. [8] proposed an Echo State Network integrated with African Buffalo Optimization and a Decision Tree. The dataset used is the DARPA dataset containing 12,478 instances and 45 features. The approach achieved an attack detection rate of 98.62% and a precision of 98.27%. The combination of African Buffalo Optimization and Decision Tree-based feature selection and Echo State Network-based classification has demonstrated strong detection performance with reduced computational requirements. However, this study was evaluated in a simulated environment without real-world hardware validation. In addition, the use of dataset-specific experimental settings and evaluation measures may limit direct comparison with other DDoS detection approaches.

The work of H. M. Belachew et al. [9] used the SDN-Edge-IoT model with Random Forest Feature Importance (RFFI) on CICIDS2017 and Edge-IIoTset datasets with 10 selected features. They achieved an accuracy of 99.9987% and achieved near real-time DDoS detection and effective mitigation by enforcing high-priority flow rules through the SDN controller. The gap in this work was limited to a single edge server and does not account for multiple edge servers to handle larger IoT traffic volumes.

Sawah M. S. et al. [10] utilised Recursive Feature Elimination (RFE) on Random Forest (RF), Naïve Bayes (NB), K-Nearest Neighbours (KNN), Linear Discriminant Analysis (LDA), and Support Vector Machine (SVM). The dataset used for the experimentation was the DDoS-SDN dataset, having 1,04,345 rows and 23 features. The study demonstrated that feature engineering and parameter tuning are critical for high-accuracy cybersecurity applications. However, further evaluation is needed to assess its ability to detect zero-day attacks and generalise to diverse and evolving attack patterns that are not represented in the training dataset.

The study by Yeen K. M. et al. [11] proposed a CRISP-DM (Cross Industry Standard Process for Data Mining) model on the COVID-19 period (2019–2020) dataset of 1,92,525 samples and 9 features. It predicted spikes in attack counts and throughput with reasonable accuracy, though predicted spike magnitudes were often up to 50% lower than actual values. But the model exhibited relatively high error rates and struggled to detect spikes in attack duration. In the work by J. F. Canola Garcia and G. E. T. Blandon [13], a functional Intrusion Detection and Prevention System (IDS/IPS) named Dique is proposed. It uses a Multi-layered Deep Feed Forward Neural Network (DFNN) model with 22 features on the CICDDoS2019 dataset to classify the incoming web server traffic into either malicious or benign categories to mitigate probable DDoS attacks. An accuracy of 99.94% is obtained, but some attack types (SYN flood, TCP flood) were misclassified due to feature overlap with benign traffic.

The work by Y. Al-Dunainawi et al. [14] contributed a one-dimensional convolutional neural network (1D-CNN) with an automatic feature extraction method, achieving 99% accuracy with 15 features on a Simulated SDN traffic dataset. However, simulated environments and datasets may not be able to fully capture the complexities and dynamics of real-world networks. Shieh, C. S. et al. [15] proposed the CNN-Geo framework, which uses a geometrical metric calculation module to identify unknown attack patterns. It highlights the need for an IDS that can promptly notify technicians of unfamiliar traffic at the onset of an attack. Y. Wang et al. [16], The research used a binary convolutional neural network (Binary CNN) with direct selection of features based on the fixed-length properties of the IPv6 protocol on the CICDDoS2019 dataset. The work contributed to a lightweight monitoring mechanism (BCDM) that utilizes a two-dimensional traffic matrix to aggregate packet-level data. It used binarized convolutional filters and hierarchical pooling to create a model with high sensitivity and minimal resource consumption in IPv6 networks. It is limited to a single 100x82 traffic matrix, which may miss longer-term sequence dependencies.

E.-U.-H. Qazi et al. [17] contributed a deep learning-based network Intrusion Detection System (DL-NIDS-ZTN) explicitly designed for zero-touch networks. It utilized a 5-layer 1-D convolutional neural network to autonomously learn features and identify malicious behaviours with a high accuracy of 99.80% on CIC-IDS2018. The study by S. Dasari and R. Kaluri [18] proposed a hierarchical machine learning framework that uses the LASSO approach for feature selection and hyperparameter-optimized models to classify intrusions. It determines the LGBM algorithm as particularly effective, achieving 99.77% accuracy, 100% precision, and recall on the CIC-IDS 2017 dataset. M. Rashid Minhas et al. [19] proposed a multi-pronged fog-level system using Zero-Shot Learning (ZSL) and Contractive Autoencoders (CAE) with autonomously extracted meaningful features. The fog layer brings intelligence closer to end devices, providing low-latency and real-time defence against zero-day threats. It achieved 99.72% accuracy on CIC-IDS2017 and 99.96% accuracy on the CIC-DDoS2019 dataset. The work by Reed, A. et al [20] addressed the risk of application-layer Slow DoS attacks in IoT networks. The study introduced a flexible Single-Attribute Intrusion Detection System, SA-IDS, that used the delta time (packet Inter-Arrival Time) attribute to differentiate between malicious traffic and legitimate slow-node activity. The framework involves the addition of a separate Guardian Node to support the border routers, which, while efficient, still represents an extra hardware requirement.

Vashistha, A. et al. [21] proposed a multi-staged AI-driven Intrusion Prevention System (IPS) deploying mathematical foundations to recognize patterns and anomalies missed by security systems. The framework combined a supervised learning approach with a reinforcement learning approach to analyse the network traffic and achieved an accuracy of 94.77%; while effective, the reward-based system carries a risk that unintended effects of an action can result in negative rewards, potentially affecting the learning process in highly volatile environments. The work by M. Srinivasan and N. C. Senthilkumar [22] contributed a novel hybridized Intrusion Detection and Prevention System (IDPS) for Industrial Internet of Things (IIoT) environments. The integration uses deep learning for detecting threats and Blockchain-assisted Reinforcement Learning (RL) for prevention. With 96.8% detection accuracy, the model scales well, but the traditional cloud-based components of the system still face data overload when dealing with the rapid growth of IIoT devices.

III. METHODOLOGY

The detailed methodology is depicted using Fig. 1 The initial step is data gathering, which involves collecting a dataset *D* that contains both *BENIGN* and *DDoS* instances. From the dataset, all the features that are non-time-based are dropped. So, only the time-based features and the label column from the complete dataset are retained. The time-based features are important in detecting and mitigating attacks; these features help in identifying patterns that indicate the initiation of the attack over a period of time.

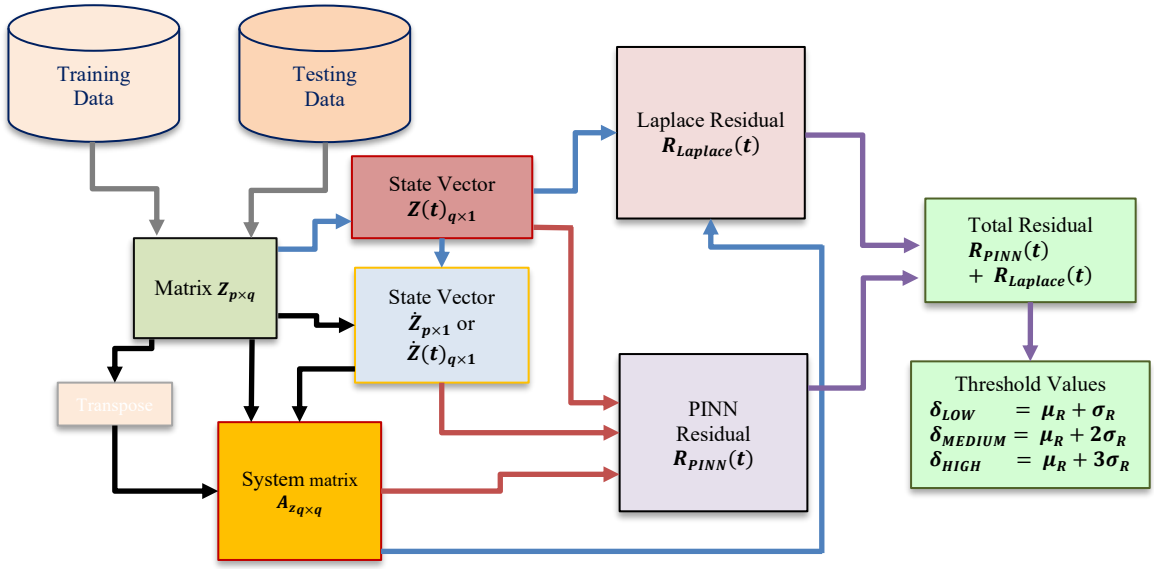


Fig. 1. Proposed system architecture.

After eliminating the non-time-based features, replace the INF values in the dataset with NaN, then remove all the instances with NaN values to make the dataset efficient for further computation. After removal of all the NaN value instances, the timestamps in the dataset can be in milliseconds or a different format; for efficiency, the timestamps are converted to seconds. The timestamp of each instance is converted to seconds, then the instances of the same second are grouped based on the label of each instance in that second. Instances are aggregated into one-second intervals and labelled as DDoS when the attack-instance proportion exceeds a configurable threshold; otherwise, they are labelled BENIGN. In this study, the threshold Δ=7% was experimentally selected. Like this, the complete dataset timestamps are standardized to one-second intervals. The dataset $D_{i\times j}$ is the refined dataset that is used for this methodology, where i denotes the number of rows and j denotes the number of columns. The training dataset $d_{k\times l}$ consists of k rows with l number of columns, and the testing dataset $d_{m\times n}$ consists of m rows with n number of columns, where $k+m=i$; $l+n=j$. For the initial experimentation, only the *BENIGN* instances were taken for the training phase. The training is done on $d_{p\times l}$ where p is the number of rows of *BENIGN* instances, and p is a subset of k. The detailed procedure is presented in Algorithm 1.

Algorithm 1:

1. $D=\{f_0, f_1, f_2, \dots, f_{j-1}, f_j\}$
   $\forall\ j = Number\ of\ features$
   $f_j = All\ the\ features\ of\ D$
2. $D[Label] = "DDoS"\ \&\&\ "BENIGN"$
3. $D = D[f_x]\ \&\&\ Label.$
   $\forall\ f_x = set\ of\ time-based\ features\ from\ f_j$
4. $if\ D = Any\ instance\ with\ NaN\ value;$
   $\quad then\ drop\ the\ instance$
5. $D[timestamp] = Converted\ into\ seconds\ \&$
   $grouped\ instance\ based\ on\ second$
6. $For\ timestamp\ t;$
   $\quad if\ count(D[Label] = "DDoS") \geq \Delta\%;$
   $\quad\quad then\ the\ second\ is\ labelled\ as\ DDoS$
   $\quad else\ labelled\ as\ BENIGN$
7. $D_{i\times j} = \{f_0, f_1, f_2, \dots, f_{j-1}, f_j\}$
   $\forall\ i = Number\ of\ instances$
   $j = Number\ of\ time-based\ features$
   $f_x = Set\ of\ all\ time-based\ features$
8. $d_{k\times l} = \{f_0, f_1, f_2, \dots, f_{j-1}, f_j\}$
   $\forall\ k = number\ of\ rows\ for\ training\ data$
   $l = number\ of\ columns\ for\ training\ data$
9. $d_{m\times n} = \{f_0, f_1, f_2, \dots, f_{j-1}, f_j\}$
   $\forall\ m = number\ of\ rows\ for\ testing\ data$
   $n = number\ of\ columns\ for\ testing\ data$
   $k, m \subset i\ and\ l, n \subset j$
10. $d_{p\times l} = d_{k\times l}[Label = "BENIGN"]$

After that, to capture network traffic dynamics in a suitable form for dynamical systems modelling, all the time-based features from $d_{p\times l}$ are taken in a matrix Z of size $p \times q$, where q is the number of time-based features and q is a subset of l shown in Eq. (1).

$$Z_{p\times q} = d_{p\times l}[f_x] \tag{1}$$

where, f_x denotes the set of time-based features.

Now, using the Z matrix, all the selected traffic features are combined into a single state vector $Z(t)$ as in Eq. (2) for each time t from 1 to the l^{th} second by applying the transpose of the matrix. The resulting state vector has dimension $q \times 1$.

$$Z(t) = \left(Z_{1\times q}\right)^T = [f_0(t)\ f_1(t)\ \dots f_j(t)]^T = \begin{bmatrix} f_0(t) \\ f_1(t) \\ \vdots \\ f_j(t) \end{bmatrix} \tag{2}$$

The rate of change of each state vector is then computed using the finite-difference relation represented by Eq. (3). The resulting derivative is denoted by $\dot{Z}(t)$. In Eq. (3), Z(t+1) represents the state vector at the next second, while Z(t) represents the state vector at the current second. Since the dataset is aggregated at one-second intervals, i.e., Δt =1 s, the derivative reduces to the difference between two consecutive state vectors. The derivative is computed up to the penultimate state vector. For the final state vector, the derivative cannot be calculated because Z(t+1) is not there.

$$\dot{Z}(t) = \frac{Z(t+1) - Z(t)}{\Delta(t)} \tag{3}$$

Next, Network traffic is modelled as a linear, time-invariant dynamical system using an ordinary differential equation (ODE) formulation in Eq. (4), where Z represents the traffic state vector, $\dot{Z}$ represents its rate of change, and A_z is the system matrix describing the relationships among the selected traffic features. The system matrix A_z, can be obtained using Least Squares Analysis. As denoted in Eq. (5), the model is rearranged to relate the observed state matrix to its derivative. Thus, solving this least-squares problem yields Eq. (6), where Z^T denotes the transpose of the matrix. The predicted A_z therefore represents the linear temporal interactions among the selected network-traffic features.

$$\dot{Z} = A_z \cdot Z \tag{4}$$

$$\dot{Z} = Z \cdot A_z{}^T \tag{5}$$

$$A_z = \left((Z^T \cdot Z)^{-1} \cdot Z^T \cdot \dot{Z}\right)^T \tag{6}$$

After computing the system matrix A_z value, next to get the PINN Residual, first compute the model's predicted derivatives at time t with Eq. (7), then use the resulting vector to find the residual between the observed derivatives and model-predicted derivatives by Eq. (8) where $\dot{Z}$ (observed value) is subtracted from the resulting vector (predicted value). Then, for each time $'t'$ till $p-1$ seconds, the summation of all the values after subtraction gives R_{PINN}. The R_{PINN} gives how much the real, observed traffic dynamics deviate from what the learned linear model expects at time t as shown in Eq. (9). After the system matrix A_z is obtained, the PINN residual is computed to quantify the difference between the observed and model-predicted traffic dynamics. First, the predicted derivative Z_{PINN} at time 't' is obtained using Eq. (7). The residual vector $Residual(t)$ is then calculated by subtracting the predicted

derivative Z_{PINN} from the observed derivative $\dot{Z}$, as given in Eq. (8). This procedure is repeated for each time step up to p-1. The residual components are then summed to obtain $R_{PINN}(t)$, as defined in Eq. (9). Thus, $R_{PINN}(t)$ represents the deviation of the observed network-traffic dynamics from the behaviour predicted by the estimated dynamical model.

$$Z_{PINN} = A_z \cdot Z(t) \quad (7)$$

$$Residual\ (t) = \dot{Z} - Z_{PINN} \quad (8)$$

$$R_{PINN}(t) = \sum Residual\ (t) \quad (9)$$

Next, Laplace ($\mathcal{L}$) is applied with respect to the differential form of the Z matrix, which transforms traffic dynamics into the Laplace domain (S) as defined in Eq. (10). So, by further computing the equations with respect to the estimated system matrix, achieved $Z_{Laplace}$ in Eq. (11), where I denotes the identity matrix and the value of S is 1. For each time t till $p-1$ seconds, the values are computed, and the resulting vector at each t is used to calculate the residual by $\dot{Z}$ (observed value) subtracted from the resulting vector (predicted value), as in Eq. (12). For each t second, the scalar Laplace residual value $R_{Laplace}$ is obtained by summing all residuals as shown in Eq. (13).

$$\mathcal{L}(\dot{Z}) = SZ_{(s)} - Z_{(0)} = A_z.Z(s) \quad (10)$$

$$Z_{Laplace} = (SI - A_z)^{-1}.Z(t) \quad (11)$$

$$Residual\ (t) = \dot{Z} - Z_{Laplace} \quad (12)$$

$$R_{Laplace}(t) = \sum Residual\ (t) \quad (13)$$

The two residual signals (R_{PINN}) and ($R_{Laplace}$) are combined into a single anomaly score as total Residual R for time 't', as shown in Eq. (14). These R values are the scores for each instance to analyze how far the observed traffic at time $'t'$ deviates from the expected behavior of the learned normal-traffic model, to predict if it is *BENIGN* or a *DDoS* instance.

$$R(t) = R_{PINN}(t) + R_{Laplace}(t) \quad (14)$$

To define a threshold value, the mean (μ) and standard deviation (σ) of these R are computed using Eq. (15) and Eq. (16). Here, the mean and standard deviation were used as they capture the natural spread of residual values under normal traffic. From the μ_R and σ_R values, the threshold value is defined for three-level alerts: low, medium, and high thresholds, making ($\mu_R + k\sigma_R$) is a statistically grounded basis for differentiating benign deviations from actual anomalies, as shown in Eq. (17) to Eq. (19).

$$\mu_R = \bar{x}_R = \frac{1}{N}\sum_{i=1}^{n} x_i \quad (15)$$

$$\sigma_R = s_R^2 = \frac{1}{N}\sum_{i=1}^{n}(x_i - \mu)^2 \quad (16)$$

$$\delta_{LOW} = \mu_R + \sigma_R \quad (17)$$

$$\delta_{MEDIUM} = \mu_R + (2 \cdot \sigma_R) \quad (18)$$

$$\delta_{HIGH} = \mu_R + (3 \cdot \sigma_R) \quad (19)$$

After the threshold values are defined, the same procedure is repeated from Eq. (1) to Eq. (14) on the complete training dataset $d_{k\times l}$ and testing dataset $d_{m\times n}$ to get the R values for every instance of *BENIGN* and *DDoS*. Based on the threshold values, the Early Warning Signals (EWS) are generated for low, medium, and high alerts for each instance, as shown below Eq. (20) to Eq. (23),

$$BENIGN = R \leq \delta_{LOW} \quad (20)$$

$$LOW\ EWS = \delta_{LOW} < R \leq \delta_{MEDIUM} \quad (21)$$

$$MEDIUM\ EWS = \delta_{MEDIUM} < R \leq \delta_{HIGH} \quad (22)$$

$$HIGH\ EWS = R > \delta_{HIGH} \quad (23)$$

IV. Experimentation and Results Discussion

A. Dataset

For the experimentation, the benchmark CICDDoS2019 dataset is utilized to evaluate the performance of the proposed method. The CICDDoS2019 dataset is an updated IDS dataset and has all the recent DDoS attack patterns. From all the attack files of CICDDoS2019, only the BENIGN and UDP instances are used for training, while in the testing data, only BENIGN and UDPLag instances are selected for experimentation in this study. This scenario was selected specifically to enable direct comparison with A. B. d. Neira et al [12] COOPRED DDoS system, the most closely related prior work, which uses the same attack types and dataset split. Table I shows that the dataset size is 31,91,508 instances for training from training day data of 01-12-2018 between 09:17 and 17:16 and 58,838 instances for testing from testing day data of 03-11-2018 between 09:18 and 17:36. After this, all the feature columns from the dataset are dropped except time-based features, Timestamp, and Label for the experimentation. The 27 time-based features are shown in Table II. Now, the dataset contains only 29 columns, from which INF values were replaced with NaN, and then all instances with a NaN value were removed from the dataset. After that, the data instances are grouped into fixed time intervals (in seconds) by converting the timestamp data values into seconds and aggregating the values of time-based features by summing all the instances for that second.

The labelling is done by taking a threshold value of 7%. If more than 7% of the instances in a given second are DDoS instances, then that second is labelled as DDoS or else BENIGN. To determine the threshold, a robustness analysis was performed to determine the 7% threshold. This is done by varying the proportion of DDoS instances per second from near zero to 50%. Then, the train–test accuracy gap at each alert level is evaluated. We observed that below the 7% threshold, the generalization gap was relatively high, reaching 10–12% at the medium-alert level, indicating sensitivity to training-specific noise. At 7%, the gap decreased to below 3% and remained low up to 10%. Beyond 15%, the gap increased again, which indicates that higher thresholds may exclude relevant attack-related traffic. Based on this analysis, a 7% threshold was selected as a suitable threshold in this study.

TABLE I. DATASET DESCRIPTION FOR TRAINING AND TESTING DATA

TRAINING		TESTING	
Labels	***No. of Instances***	***Labels***	***No. of Instances***
UDP	31,34,645	UDPLag	1,873
BENIGN	56,863	BENIGN	56,965

Therefore, 7% is the smallest threshold at which the model generalized reliably, to avoid both overfitting at lower thresholds and signal loss at higher threshold values, as shown in Table III. Now, after grouping and aggregating, the dataset contains 9,237 instances in training and 7,416 instances in the testing dataset.

TABLE II. SELECTED LIST OF TIME-BASED FEATURES

Sno.	Feature Name	Sno.	Feature Name
1	Flow Bytes/s	15	Bwd IAT Mean
2	Flow Packets/s	16	Bwd IAT Std
3	Flow Duration	17	Bwd IAT Max
4	Flow IAT Mean	18	Bwd IAT Min
5	Flow IAT Std	19	Bwd Packets/s
6	Flow IAT Max	20	Active Mean
7	Flow IAT Min	21	Active Std
8	Fwd IAT Total	22	Active Max
9	Fwd IAT Mean	23	Active Min
10	Fwd IAT Std	24	Idle Mean
11	Fwd IAT Max	25	Idle Std
12	Fwd IAT Min	26	Idle Max
13	Fwd Packets/s	27	Idle Min
14	Bwd IAT Total		

Table IV shows the instances of each label in the training and testing datasets used in this experiment.

TABLE III. ANALYSIS OF THRESHOLD VALUE FOR LABELLING AGGREGATED DATASET BASED ON TRAIN-TEST GENERALIZATION GAP

Threshold	MEDIUM-Alert Gap	Average Gap (All Alerts)
1% to 6%	10.4% - 12.5%	20.1% – 22.4%
7%	2.94%	16.18%
8% to 10%	2.88% - 3.27%	16.14% - 16.36%
15%	10.50 %	19.17%
20%+	11.13%+	~16% - 17%

TABLE IV. DATASET DESCRIPTION FOR TRAINING AND TESTING DAY TRAFFIC AGGREGATED IN SECONDS

TRAINING		TESTING	
Labels	***No. of Instances***	***Labels***	***No. of Instances***
DDoS	1505	DDoS	460
BENIGN	7732	BENIGN	6956

B. Scores and EWS Threshold Values

From the training day dataset, only the BENIGN instances are used during training to build the model for experiments. Thus, the matrix Z is of size 7732×27, indicating 7732 instances and 27 features, from which Z^T is obtained as a matrix of size 27×7732. The next step is the computation of temporal derivation, $\dot{Z}(t)$ by finding the derivatives between $Z(t+1)$ and $Z(t)$ where $Z(t)$ is the current value and $Z(t+1)$ is the next value. The value of 't' is varied from $t = 1\ to\ 7731$. As in the last instance $\dot{Z}(7732)$ cannot be computed due to no next instance to find the derivative, the last sample is excluded. The resulting data and derivative matrices are subsequently used in the ordinary differential equation formulation to estimate the system matrix A_z according to Eq. (4) to Eq. (6). From the A_z matrix, PINN residual, R_{PINN} and Laplace residual, $R_{Laplace}$ are determined using Eq. (7) to Eq. (13), for each $t = 1\ to\ 7731$. The total residual (R) is calculated by the sum of R_{PINN} and $R_{Laplace}$ as defined by Eq. (14). The resulting R values are the scores based on which the alerts are generated for the individual instances. The resulting R values serve as warning scores for generating alerts for individual instances. The residual values are shown in Table V for the R values. The mean and standard deviation values of the residuals, R, are calculated using Eq. (15) and Eq. (16), yielding $\mu_R = -140.1766$, $\sigma_R = 1190671971.4852$.

The threshold values are determined based on these statistics, resulting in $\delta_{LOW} = 1190671831.3086$, $\delta_{MEDIUM} = 2381343802.7938$ and $\delta_{HIGH} = 3572015774.2789$, using Eq. (17) to Eq. (19). After defining threshold values, the procedure is repeated on the complete training and testing dataset containing both BENIGN and DDoS instances to generate the alerts based on the R values for each instance. The predicted alerts are classified as: BENIGN if $R \leq 1190671831.3086$, LOW if $1190671831.3086 < R \leq 2381343802.7938$, MEDIUM if $2381343802.7938 < R \leq 3572015774.2789$ and HIGH if $R > 3572015774.2789$.

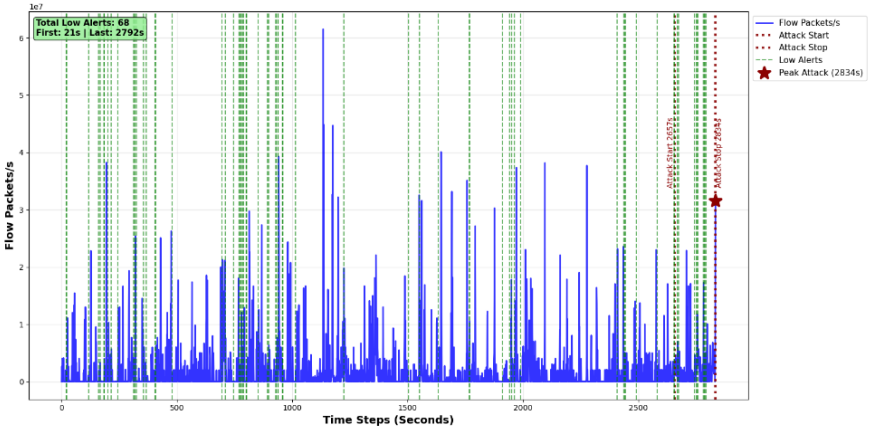
TABLE V. RESULTING RESIDUAL VALUES FOR THE DATASET

Time	PINN Residual	Laplace Residual	Total Residual
1	477500352	-2204316.58	475296036
2	332947127	3091892.24	3.3604
3	-179883629	-346281337	-526164966
:	:	:	:
7729	336106100	33283018.7	369389119
7730	19467270	-427148653	-407681383
7731	-25775345.1	-1758958.43	-27534303.5

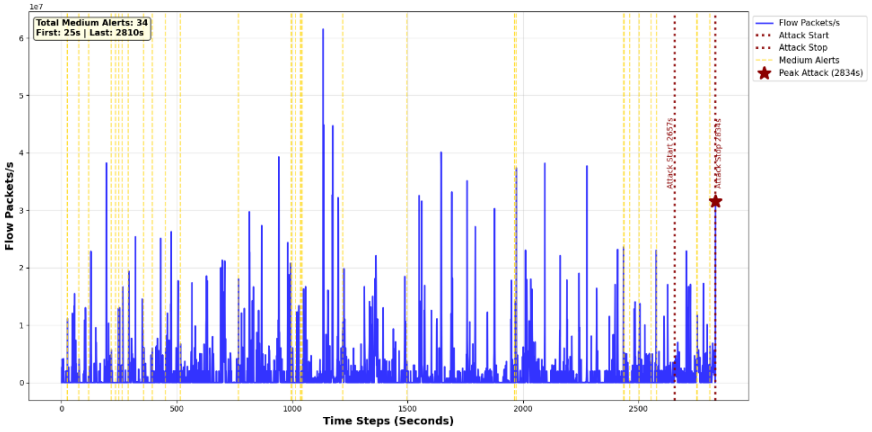
C. Results

The results of the experimentation are shown to demonstrate how well the model is able to generate an early warning signal (EWS) before both the onset of the attack and the peak of the DDoS attack. The testing dataset for the experimentation contains both attack and benign instances. As shown in Fig. 2(a), 2(b), and 2(c), the attack started at 2657 seconds (11:13:03), with the attack reaching its peak at 2834 seconds (11:17:41), and the attack ended at 3256 seconds (11:27:53). The early warning signals are generated based on the low, medium, and high alert conditions. The total alerts generated are 134, of which 68 are low alerts, 34 are medium alerts, and 32 are high alerts. The first low alert was generated at 21 seconds (09:20:41), which is 46 minutes 53 seconds before the peak of the attack, and the last low alert was at 2792

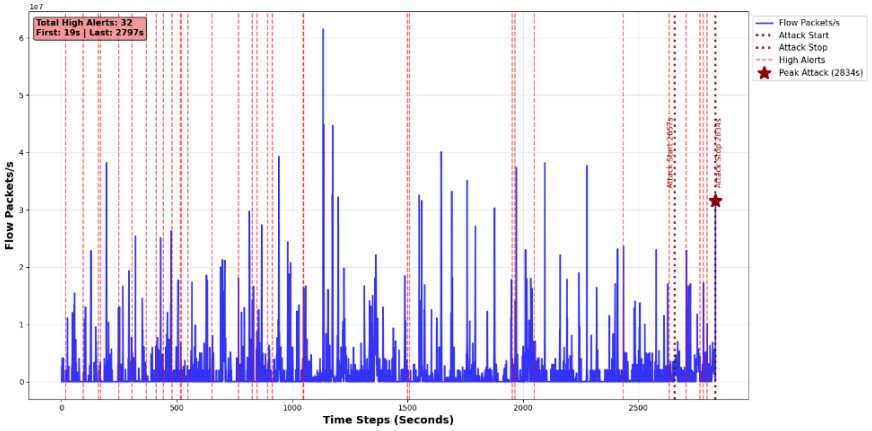
seconds. The first medium alert was generated at 25 seconds (09:20:51), and the last medium alert was at 2810 seconds. The first medium alert is 46 minutes 49 seconds before the peak of the attack. The first high alert at 19 seconds (09:20:31) is 46 minutes 55 seconds before the peak of the attack, and the last high alert is at 2797 seconds. In Fig. 3(a), the confusion matrix of the testing dataset containing both attack and benign instances is shown. The confusion matrix represents the classification performance of early warning signals across various alert levels.



(a)



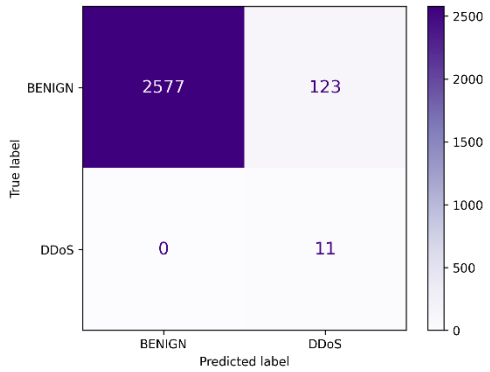
(b)



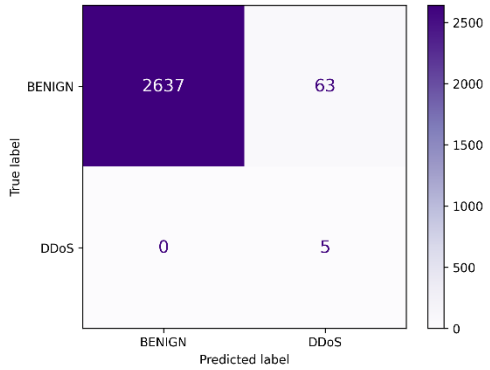
(c)

Fig. 2. EWS generated for testing data for: (a) low alerts; (b) medium alerts; (c) high alerts.

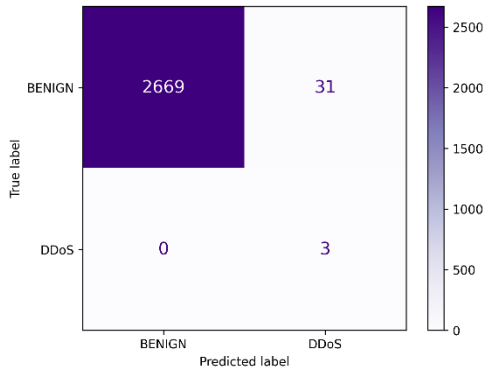
Fig, 3(a) represents the confusion matrix obtained for testing data. As shown in Fig. 3(a), 2,577 BENIGN instances were correctly classified, while 123 BENIGN-labelled instances triggered early warning alerts. All eleven DDoS instances were correctly identified, with no false negatives. Of the 123 nominal false positive alerts, 117 (95.12%) alerts occurred before the first labelled DDoS instance. This temporal concentration of alerts before the attack onset indicates the potential of the proposed framework to provide warning of emerging abnormal network conditions.



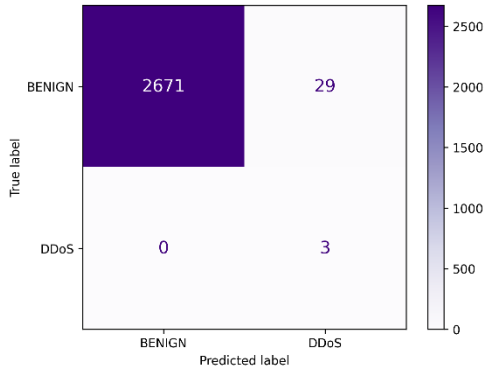
(a)



(b)



(c)



(d)

Fig. 3. Confusion matrix of Testing data for: (a) all alerts; (b) low alerts; (c) medium alerts; (d) high alerts.

The three confusion matrices in Fig. 3(b) to 3(d) successively record the classification performance of the three EWS warning levels: LOW, MEDIUM, and HIGH EWS. The true negative rates (TNRs) of the three groups are 97.67%, 98.85%, and 98.93%, respectively. The true negative rate can be understood as the proportion of cases that are originally normal, correctly identified as normal by the system, and do not trigger an unnecessary warning; the higher this value, the better. The corresponding false positive rates (FPRs) are 2.33%, 1.15%, and 1.07%. The false positive rate is the proportion of cases that are originally normal, misjudged by the system as problematic, and incorrectly trigger a warning; the lower this value, the better. The Low-level warning was triggered a total of 68 times, of which 63 times were false alarms triggered by normal samples marked as BENIGN, and the remaining 5 times correctly identified DDoS attacks. The MEDIUM-level warning was triggered 34 times; 31 times were false alarms triggered by normal samples marked as BENIGN, and 3 times accurately caught DDoS attacks. The High-level warning was triggered 32 times; 29 times were false alarms triggered by normal samples marked as BENIGN, and another 3 times correctly identified DDoS attacks. It is worth noting that not a single DDoS attack was missed at any warning level. In all tested configurations, the number of false negatives was 0 — a false negative refers to a missed alarm where an attack was actually occurring, but the system failed to detect it and did not trigger a warning. No missed alarm problem occurred at all in this test.

Table VI summarizes the performance of the proposed EWS framework on the training and testing datasets up to the attack peak.

TABLE VI. PERFORMANCE OF THE EWS FRAMEWORK ON TRAINING AND TESTING DATASETS UP TO THE ATTACK PEAK

	Alerts	Accuracy	Precision	Recall	F1-Score	Balanced Accuracy
Training Dataset	All Alerts	95.98	74.60	100	85.45	97.72
	LOW Alerts	97.95	45.52	100	62.56	98.96
	MEDIUM Alerts	98.77	41.25	100	58.41	99.38
	HIGH Alerts	98.88	89.69	100	94.57	99.38
Testing Dataset	All Alerts	95.46	8.21	100	15.17	97.72
	LOW Alerts	97.67	7.35	100	13.70	98.83
	MEDIUM Alerts	98.85	8.82	100	16.22	99.43
	HIGH Alerts	98.93	9.38	100	17.14	99.46

For the training dataset, all alert levels achieved 100% recall, with a balanced accuracy of 97.72% for the combined alerts. The LOW, MEDIUM, and HIGH alert levels achieved balanced accuracies of 98.96%, 99.38%, and 99.38%, respectively. On the testing dataset, the framework also achieved 100% recall across all alert levels, indicating that no labelled DDoS instances were missed. The balanced accuracy was 97.72% for all alerts and increased to 98.83%, 99.43%,

and 99.46% for LOW, MEDIUM, and HIGH alerts, respectively. Although testing precision was low because several BENIGN-labelled instances triggered early warnings, their temporal occurrence before the labelled attack onset is separately considered when evaluating the advance-warning capability of the proposed EWS framework.

The objective of the proposed framework is to provide warnings before DDoS attack onset rather than only identify labelled DDoS attack instances. Hence, the low testing precision between 7.35% and 9.38% across alerts should be interpreted together with the temporal distribution of the alerts. The proposed framework attained a high true negative rate, which is between 95.44% and 98.93%, and 100% recall, indicating that most of the BENIGN traffic is correctly identified while no DDoS instances are missed. The low precision results are due to the strong class imbalance and warnings generated during the BENIGN-labelled time period.

Importantly, out of the 123 false positive alerts, 117 alerts have occurred before the first DDoS instance, which amounts to 95.1%. Thus, the temporal concentration of alerts before the attack onset is to be considered, which supports the relevance to the early-warning objective of the proposed framework, although they remain false positives under conventional classification. The LOW, MEDIUM, and HIGH alert levels further support prioritization, with higher alert levels showing improved precision and lower false positive rates.

TABLE VII. COMPARISON OF DDOS EARLY-WARNING LEAD TIME AND PRE-ATTACK ALERTS WITH AN EXISTING APPROACH

S. No.	Method	Warning Lead Time before Attack Onset	No. of Alerts before Attack Onset
1	de Neira, et al. [12]	2 minutes 08 seconds	4
2	Proposed (Low alert)	43 minutes 56 seconds	58
3	Proposed (Medium alert)	43 minutes 52 seconds	31
4	Proposed (High alert)	43 minutes 58 seconds	28

Table. VII shows a comparison of the proposed method with the existing work of de Neira et al. [12] COOPRED DDoS system (Experiment 2), as both evaluations use the same CICDDoS2019 UDP attack (training) and UDPLag attack (testing) scenario. Neira et al. [12] computed statistical early-warning signals (kurtosis, skewness, and spectral density) on the packet-count time series and classified each second as normal or attack-warning using a supervised classifier, which predicted an alert 2 minutes 08 seconds before the start of the attack. The proposed method instead scores every timestamp using a combined residual across multiple traffic features, producing continuous three-level alerts. The medium alerts were generated 43 minutes 52 seconds, low alerts generated 43 minutes 56 seconds, and high alerts generated 43 minutes 58 seconds before the start of the attack. The proposed method is a fundamentally more continuous detection mechanism rather than waiting for a rare statistical spike. The trade-off is precision: Neira et al. work reports higher precision (33%-100%) because their method triggers rarely and conservatively, while our method is a continuous, multi-feature approach that produces more frequent but lower-precision alerts in much earlier and more consistent warning coverage, consistent with maximizing early-warning time rather than minimizing alert counts. With a better prediction time, the proposed method gives a network administrator more time to make better use of it and act more effectively to mitigate attacks before they reach their peak.

V. CONCLUSION

Anticipating cyberattacks is a fundamentally harder problem than detecting them post-hoc, and increasing the integrity of connected systems requires an architecture of defense in depth. Traditional reactive methods degrade quickly as attack vectors evolve, especially when it comes to new attack vectors. Thus, the fundamental goal of threat prediction is to separate harmful staging operations and generate alert signals in a timely manner, where alert generation is done long before an intrusion even executes. The proposed architecture achieves this by generating multi-level Early Warning Signals (EWS) such as low, medium, and high, well ahead of the peak of the attack and ahead of attack execution. In order to achieve this, the proposed architecture generates multi-level EWS signals divided into low, medium, and high threat classes. We then use the ODE formulation to obtain a Physics-Informed Neural Network and Laplace residuals from discrete data samples. These residual vectors then determine the empirical boundaries required for partitioning network traffic into the respective risk levels. The system obtained a combined balanced accuracy of 97.72% over the alert spectrum. On a more granular scale, the low-alert generation reported 98.83% balanced accuracy, sounding an alarm 46 minutes and 53 seconds before peak attack severity, and 43 minutes and 56 seconds before first impact. The medium-alert level achieved a balanced accuracy of 99.43%, with warnings sent out 46 minutes and 49 seconds before the peak and 43 minutes and 52 seconds ahead of onset, respectively. In parallel, the high-level alert class achieved a balanced accuracy of 99.46%, which shifted critical alerts to 46 minutes and 55 seconds prior to the peak and to 43 minutes and 58 seconds before the attack began. Future work will focus on extending the lead time of this prediction as well as improving the accuracy and robustness of the EWS generation pipeline for a wider variety of network topologies and threat vectors with respect to distributed denial of service network traffic.

REFERENCES

[1] De Neira, A. B., Kantarci, B., & Nogueira, M. Distributed denial of service attack prediction: Challenges, open issues and opportunities. Computer Networks, 222, 109553, 2023.

[2] He, J., Wang, X., Song, Y., & Xiang, Q. A multiscale intrusion detection system based on pyramid depthwise separable convolution neural network. Neurocomputing, 530, 48–59, 2023.

[3] Zhou, H., Zheng, Y., Jia, X., & Shu, J. Collaborative prediction and detection of DDoS attacks in edge computing: A deep learning-based approach with distributed SDN. Computer Networks, 225, 109642, 2023.

[4] Sharma, A., Rani, S., Shah, S. H., Sharma, R., Yu, F., & Hassan, M. M. An efficient hybrid deep learning model for denial of service detection in cyber-physical systems. IEEE Transactions on Network Science and Engineering, 10(5), 2419–2428, 2023.

[5] Singh, V. K., Sivashankar, D., Kundan, K., & Kumari, S. An efficient intrusion detection and prevention system for DDoS attack in WSN

using SS-LSACNN and TCSLR. Journal of Cyber Security and Mobility, 13(1), 135–159, 2024.

[6] Wang, H., & Li, Y. Overview of DDoS attack detection in software-defined networks. IEEE Access, 12, 38351–38381, 2024.

[7] Alashhab, A. A., et al. Enhancing DDoS attack detection and mitigation in SDN using an ensemble online machine learning model. IEEE Access, 12, 51630–51649, 2024.

[8] Rao, G. S., Patra, P. S. K., Narayana, V. A., Reddy, A. R., Reddy, G. N. V. V., & Eshwar, D. DDoSNet: Detection and prediction of DDoS attacks from realistic multidimensional dataset in IoT network environment. Egyptian Informatics Journal, 27, 100526, 2024.

[9] Belachew, H. M., Beyene, M. Y., Desta, A. B., Alemu, B. T., Musa, S. S., & Muhammed, A. J. Design a robust DDoS attack detection and mitigation scheme in SDN-Edge-IoT by leveraging machine learning. IEEE Access, 13, 10194–10214, 2025.

[10] Sawah, M. S., Elmannai, H., El-Bary, A. A., et al. Distributed denial of service (DDoS) classification based on random forest model with backward elimination algorithm and grid search algorithm. Scientific Reports, 15, 19063, 2025.

[11] Yeen, K. M., Noor, R. M., Shah, W. M., Hassan, A., & Munir, M. U. Forecasting future DDoS attacks using Long Short Term Memory (LSTM) model. International Journal of Wireless & Mobile Networks, 17(4), 2025.

[12] De Neira, A. B., De Araujo, A. M., & Nogueira, M. An intelligent system for DDoS attack prediction based on early warning signals. IEEE Transactions on Network and Service Management, 20(2), 1254–1266, 2023.

[13] Cañola Garcia, J. F., & Blandon, G. E. T. A deep learning-based intrusion detection and preventation system for detecting and preventing denial-of-service attacks. IEEE Access, 10, 83043–83060, 2022.

[14] Al-Dunainawi, Y., Al-Kaseem, B. R., & Al-Raweshidy, H. S. Optimized artificial intelligence model for DDoS detection in SDN environment. IEEE Access, 11, 106733–106748, 2023.

[15] Shieh, C. S., Nguyen, T. T., & Horng, M. F. Detection of unknown DDoS attack using convolutional neural networks featuring geometrical metric. Mathematics, 11(9), 2145, 2023.

[16] Wang, Y., Wang, X., Ni, Q., Yu, W., & Huang, M. BCDM: An early-stage DDoS incident monitoring mechanism based on Binary-CNN in IPv6 network. IEEE Transactions on Network and Service Management, 21(5), 5873–5887, 2024.

[17] [Qazi, E.-U.-H., Zia, T., Faheem, M. H., Shahzad, K., Imran, M., & Ahmed, Z. Zero-touch network security (ZTNS): A network intrusion detection system based on deep learning. IEEE Access, 12, 141625–141638, 2024.

[18] Dasari, S., & Kaluri, R. An effective classification of DDoS attacks in a distributed network by adopting hierarchical machine learning and hyperparameters optimization techniques. IEEE Access, 12, 10834–10845, 2024.

[19] Minhas, M. R., et al. F-OSFA: A fog level generalizable solution for zero-day DDoS attacks detection. IEEE Access, 13, 75157–75170, 2025.

[20] Reed, A., Dooley, L., & Mostefaoui, S. K. SA-IDS: A single attribute intrusion detection system for Slow DoS attacks in IoT networks. Internet of Things, 30, 101512, 2025.

[21] Vashistha, A., Alghamdi, T. A., Almalki, S. A., et al. AI-driven intrusion prevention system: Leveraging mathematical foundations for advanced threats detection. International Journal of Information Technology, 17, 4365–4370, 2025.

[22] Srinivasan, M., & Senthilkumar, N. C. Intrusion detection and prevention system (IDPS) model for IIoT environments using hybridized framework. IEEE Access, 13, 26608–26621, 2025.