Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 9, No. 6, December, 2025

DML-IDS: Distributed Multi-Layer Intrusion Detection System for Securing Healthcare Infrastructure

Mohamed Sirajudeen Yoosuf [1*], P. Vijaya [1], Joseph Mani [1]

[1] *Department of Mathematics and Computer Science, Modern College of Business and Science, Bowshar, Muscat, Oman.*

Abstract

In recent years, the number of cyberattacks targeting healthcare resources has rapidly increased. Conventional IDSs rely heavily on predefined rules and attack signatures. However, modern zero-day attacks with unpredictable behavior and multi-vector attack patterns can still breach healthcare networks. When a new type of cyberattack targets a specific server, an existing IDS may fail to detect it because it depends on static, predefined rules. To address these issues, we propose DML-IDS: Distributed Multi-Layer Intrusion Detection System, designed to operate across multiple nodes in a network to collaboratively detect suspicious activities. The proposed approach employs a multi-layer ensemble strategy to improve detection accuracy while reducing computational overhead on a single machine. All incoming network packets are first analyzed by the Distributed Threat Analysis Module (DTAM), which runs a Random Forest-based model as the base classifier to distinguish between benign and malicious traffic. Based on the nature and severity of the threat, malicious packets are flagged as highAlert (HA) in the Threat Prioritization Layer (TPL) and then forwarded to the respective Confirmatory Ensemble Model (CEM) for further, attack-specific analysis. These CEM models are designed to scale efficiently and detect zero-day as well as multi-vector attacks. The proposed model was trained on the CICIDS-2017 dataset. DTAM achieved an accuracy of 98.5%, while the CEM models for DDoS, Patator, and Web Attack achieved 99.01%, 98.87%, and 98.91% accuracy, respectively. Furthermore, the computational overhead of the DML-IDS architecture was evaluated and compared with an existing ensemble learning-based IDS.

Keywords:

Network Security;
Intrusion Detection System;
Ensembled IDS;
Cyber Attacks;
Distributed IDS.

Article History:

Received:	07	July	2025
Revised:	28	October	2025
Accepted:	04	November	2025
Published:	01	December	2025

1- Introduction

With the rapid increase in the use of communication technologies and the advancement of Artificial Intelligence-based botnets, there has been a significant rise in cyber-attacks. Healthcare platforms have become prime targets for attackers, as they contain sensitive health information about individuals and often lack robust security measures. According to the Healthcare Data Breach Statistics 2023 from the HIPAA Journal, around 725 healthcare data breaches were reported, exposing the health records of approximately 133 million individuals [1]. Similarly, the Veriti Research Report 2025 states that around 400 healthcare organizations in the United States have experienced cyber-attacks, including ransomware attacks [2]. Meanwhile, the use of AI-based botnets has also grown rapidly. By leveraging advanced monitoring techniques, botmasters can initiate, deploy, and monitor Distributed Denial of Service (DDoS) attacks on healthcare platforms. GorillaBot 2024 cyber-attack has emerged as one of the most powerful DDoS threats. GorillaBot launched 300,000 DDoS attacks across 100 countries within just 24 days in 2024 [3].

Given these incidents and their consequences, implementing robust cybersecurity measures for healthcare infrastructure is essential. Traditional approaches—such as firewalls, rule-based Intrusion Detection Systems (IDS),

[*] **CONTACT**: mohamed.yoosuf@mcbs.edu.om

DOI: http://dx.doi.org/10.28991/ESJ-2025-09-06-016

Intrusion Prevention Systems (IPS), Virtual Private Networks (VPNs), and antivirus software—are often inadequate in defending against modern, sophisticated cyber threats.

Conventional Intrusion Detection Systems use machine learning algorithms to train models for classifying network traffic as normal or malicious. Conventional IDS are pre-trained models that are learned from historical attack patterns. Bhati et al. [4] presented an analytical study on Support Vector Machine and its variants for detecting network attacks and intrusions. The SVM-based IDS was trained using different SVM models such as Linear SVM, Quadratic SVM, Fine Gaussian SVM, and Medium Gaussian SVM. The dataset used to train these models is the NSL-KDD dataset. The SVM model achieved a maximum accuracy of 98.7%. However, the trained SVM models were incapable of detecting advanced and adversarial network attacks, resulting in high false negative errors. Similarly, Azam et al. [5] presented a comprehensive review on the challenges faced by the traditional Intrusion Detection Systems (IDS), such as signature-based (SIDS) and anomaly-based (AIDS) approaches, in proactively identifying the cyber-attacks. SIDS mostly rely on predefined attack signatures and struggle to identify zero-day attacks. On the other hand, AIDS can detect unknown threats; however, it suffers from a high false-positive rate.

The primary issue with respect to the conventional intrusion detection system is mostly relying on the preexisting rule sets and attack signatures, which makes it infeasible to detect multi-vector and zero-day attacks. This often results in an increased false positive rate. To overcome these challenges, ensemble learning was introduced in the Intrusion Detection System. It combines multiple algorithms to create a robust and accurate detection system. Ahmed et al. [6] proposed a HAEnID architecture that integrates three ensembled models, such as Stacking Ensemble, Bayesian Model Averaging, and the Conditional Ensemble Method, to enhance the detection accuracy. In the training process, the base classifiers are trained with Decision Trees, Random Forests, Multi-Layer Perceptron, Logistic Regression, LightGBM, and AdaBoost on the CIC-IDS2017 dataset. The hybrid model achieves a higher accuracy of 98.79%. The entire HAEnID architecture and multi-layer computation are executed on a centralized processing system, which greatly increases the computational overhead and poses a challenge in scalability.

Alsolami et al. [7] proposed an ensemble-based IDS architecture specifically designed for medical IoT devices, integrating stacking, bagging, and boosting techniques. It uses Random Forest and Support Vector Machine as base classifiers. The model is trained on the WUSTL-EHMS 2020 dataset, where stacking achieves the highest accuracy of 98.88%, followed by bagging at 97.83% and boosting at 88.68%. The main drawback of this technique is its high susceptibility to overfitting, as the continuous error-correction process in ensemble learning methods may produce highly specialized models that closely fit the training data. Furthermore, it does not address zero-day attack detection.

Doost et al. [8] proposed a hybrid IDS architecture integrating Convolutional Neural Networks for automated feature extraction with a Random Forest classifier. The model is trained on KDD99 and UNSW-NB15 datasets, achieving an accuracy of 97.36% and a precision of 98.46%, outperforming baseline methods such as NBTree and SVM. However, the approach incurs high computational overhead. Fares et al. [9] proposed an anomaly-based IDS for IoT environments by integrating a TabNet Transformer and Google Vizier for hyperparameter optimization. The model also incorporates SHAP (Shapley Additive Explanations) to enhance interpretability, achieving 98.29% accuracy with the NSL-KDD dataset. However, the evaluation mainly focused on traditional cyber-attacks and did not address zero-day threats. While SHAP improves transparency, it may introduce computational bottlenecks on IoT devices.

Similarly, Torre et al. [10] proposed a Federated Learning Intrusion Detection System (FL-IDS) using a one-dimensional Convolutional Neural Network (CNN) for IoT environments. The system also integrates privacy-preserving techniques such as Differential Privacy (DP), the Diffie-Hellman Key Exchange (DHE) algorithm, and Homomorphic Encryption (HE) to perform computations on encrypted data. The FL-IDS achieved 97.31% accuracy using the TON-IoT dataset. Although the results are promising, the computational load may affect overall network latency, and the scalability of the IDS on resource-constrained IoT devices is not explored.

Nassreddine et al. [11] presented a Network Intrusion Detection System (NIDS) that employs ensemble machine learning techniques combined with a hybrid feature selection mechanism to accurately detect malicious packets. The system integrates Correlation-based Feature Selection (CFS) with embedded feature selection methods to identify the most relevant features. While the model achieved 99.99% accuracy with the NSL-KDD dataset, it does not address the risk of overfitting. Moreover, its accuracy may degrade in real-time scenarios where data is more volatile and less structured.

From the above studies, it is evident that modern-day cybersecurity practices use conventional, ensemble, and hybrid machine learning techniques in Intrusion Detection Systems to improve detection accuracy. However, several challenges remain unaddressed, such as limited adaptability, computational overhead, and inability to detect multi-vector and zero-day attacks.

1-1- Limitations

The limitations of existing conventional, ensemble, and hybrid machine learning techniques are as follows:

- Conventional Intrusion Detection Systems rely heavily on pre-existing rule sets, patterns, and signature-based detection, which make them incapable of identifying zero-day and multi-vector bot-based attacks.
- Most existing IDS architectures are not designed to adapt to evolving cyber threats, which limits their effectiveness in high-traffic and dynamic network environments.
- Many existing IDS models are deployed either on the firewall or at the network gateway (within the internal network), making them insufficient for distributed practices.
- Several IDS models use hybrid or ensemble machine learning techniques to improve threat detection accuracy. However, these heavy-weight models significantly increase the computational overhead on the machines running the IDS.
- Ensemble learning-based IDS that are implemented on a centralized multi-layer computation make it difficult to scale across distributed or high-traffic network environments.

To overcome the limitation of existing intrusion detection systems, a DML-IDS, or Distributed Multi-Layer Intrusion Detection System, is proposed. It facilitates a scalable, distributed, and collaborative IDS scheme to identify the zero-day and multi-vector attacks in healthcare networks. The proposed DML-IDS scheme uses a multi-layer approach and ensembled techniques to proactively detect cyber-attacks.

1-2- Contributions

The contribution of the proposed DML-IDS model is as follows:

- ***Distributed IDS Framework*****:** The proposed research introduces a distributed intrusion detection framework by deploying IDS across multiple network nodes, which facilitates collaborative detection and enhances scalability.
- ***Multi-Layer Approach*****:** The proposed DML-IDS framework uses a multi-layered approach, in which the first layer is (i) Distributed Threat Analysis Module (DTAM), the second layer is (ii) Threat Prioritization Layer (TPL), and the third layer is (iii) Confirmatory Ensemble Model (CEM), to ensure accurate and scalable detection. The first layer, DTAM, uses basic machine learning algorithms to perform initial screening of incoming packets. The second layer, TPL, analyzes the packets and checks for suspicious activities. If any are identified, the TPL denotes the corresponding packets as cyber threats. These suspicious packets are then further analyzed in the Confirmatory Ensemble Model (CEM).
- ***Ensembled Models to Improve Detection Accuracy*****:** To enhance threat detection accuracy, this research leverages ensemble techniques by combining multiple machine learning algorithms such as SVM, Random Forest, and Logistic Regression. Although the computation time of ensemble techniques is higher than traditional intrusion detection systems, the distributed nature of the proposed framework ensures that computational efficiency is not compromised.

The paper is organized as follows: Section 2 discusses recent research on intrusion detection systems, distributed IDS, and ensemble machine learning techniques. Section 3 presents the system model of the proposed DML-IDS framework. Section 4 presents the performance evaluation and comparative analysis of the proposed DML-IDS with existing IDS schemes. Section 5 discusses the conclusion and future work.

2- Related Works

In this section, previous studies on Intrusion Detection Systems (IDS) based on deep learning, ensemble, and hybrid machine learning models are briefly described. These approaches have been extensively explored to enhance detection accuracy, adaptability, and robustness against emerging cyber threats.

Xu et al. [12] presented an in-depth analysis of deep learning-based intrusion detection systems (DL-IDS), covering all phases including data collection, log analysis, graph summarization, and attack detection/analysis. The study explicitly mentions unsolved challenges in these systems, such as robustness and real-time constraints. Zhang et al. [13] reviewed deep learning applications in IDS with a focus on spatiotemporal feature extraction. They discussed issues related to class imbalance in IDS classification. The review emphasized that hybrid CNN–RNN architectures are more effective in capturing temporal correlations and spatial dynamics in packet-level data. However, due to class imbalance, these models perform poorly on minority attack types. To overcome this, the paper recommends using resampling methods and generative adversarial networks (GANs) to improve detection rates for rare attacks.

Mamatha et al. [14] proposed a Hybrid Ensemble Feature Engineering approach that combines Boruta, Relief, and Pearson correlation feature selection methods. Decision Tree, Random Forest, and Gradient Boosting were employed for Stacked Ensemble Classifiers. The hybrid ensemble model was trained and tested with the CICIDS-2017 dataset, specifically focusing on DoS and DDoS attacks. This hybrid feature selection method improved both training efficiency and detection performance due to dimensionality reduction. Accuracy was achieved above 98%, and precision was increased in DoS attacks. However, the processing time increased, and the setup process was complicated due to the multiple steps required for pre-processing and training. Ataa et al. [15] proposed a deep learning-based IDS for Software Defined Networks (SDN), focusing on LSTM, CNN, and hybrid models to secure SDN controllers and maintain control flows in the network. The research showed that hybrid models result in increased detection accuracy. However, implementing hybrid models on a single machine introduces significant latency and resource limitations, making real-time operation challenging. Therefore, the study recommends further development to balance computational efficiency and processing load.

Amouri et al. [16] presented a hybrid IDS that combines Kolmogorov-Arnold Networks (KAN) and XGBoost to improve intrusion detection in IoT environments. It uses KAN for feature transformation and representation learning and XGBoost as the final classifier. The method achieved accuracy above 99%, as well as high levels of Precision and Recall. Although it is robust in detecting various attacks, the computational load for training was higher than that of traditional ML classifiers, making it challenging for direct application on resource-constrained devices. Biber et al. [17] presented a comprehensive comparative study of individual ML models and ensemble strategies using two datasets, RoEduNet-SIMARGL2021 and CICIDS-2017. Ensemble methods such as Bagging, Stacking, Blending, Boosting, and individual models such as Decision Tree, Random Forest, SVM, and Neural Network were tested. The study found that ensemble methods consistently outperformed individual classifiers. In particular, the Stacking method achieved 99.1% accuracy on CICIDS-2017 and 98.7% accuracy on RoEduNet-SIMARGL2021. Bagging and Boosting methods were found to be better at reducing false positives. However, Stacking and Blending methods have challenges in implementing large-scale live IDS systems, as they require high computational loads

To enhance NIDS capabilities, Liu et al. [18] incorporated host telemetry data and network flow information. By utilizing a deep learning pipeline that integrates both sources, the system outperforms standalone network-based models in terms of detection accuracy and false positives. This demonstrates the effectiveness of integrating multi-source features to improve IDS reliability in various operational contexts.

Lansky et al. [19] presented a fundamental review of deep learning-based intrusion detection systems. They classified IDS approaches based on network types, such as autoencoders, convolutional neural networks (CNNs), recurrent neural networks (RNNs), and restricted Boltzmann machines (RBMs), and evaluated their performance on datasets such as KDD-Cup, NSL-KDD, and UNSW-NB15. They described the stages in feature discovery and classification and highlighted challenges such as model generalization and interpretability.

In another study, Gao et al. [20] applied ensemble machine learning techniques to build an adaptive IDS model and emphasized its significance in intrusion detection development. Bringer et al. [21] conducted a review on honeypots in cybersecurity, analyzing recent advancements and future trends. Titarmare et al. [22] provided a detailed overview of honeypot systems, including their functions, types, and benefits. Verma & Dubey [23] discussed the development and real-time deployment of honeypots in network environments. Sharafaldin et al. [24] introduced the CICIDS-2017 dataset and compared it with existing datasets such as DARPA98, KDD99, ISC2012, and ADFA13 used for evaluating IDS and intrusion prevention approaches. They also evaluated network traffic features and applicable machine learning algorithms. Abbas et al. [25] developed an ensemble machine learning model for the Internet of Things and discussed the benefits of ML ensembling. Zhou et al. [26] proposed a distinctive method for model assembly and feature selection, explaining various algorithm combinations and recommending the most effective model.

Das et al. [27] conducted a comparative analysis highlighting the advantages of ensemble ML models, also using the CICIDS-2017 dataset. Thockchom et al. [28] introduced a novel ensemble model trained on the CICIDS-2017 dataset, demonstrating performance improvements over individual models. Mhawi et al. [29] proposed an advanced feature selection mechanism to extract optimal features for training ensemble ML models. Maseer et al. [30] benchmarked various ML algorithms using the CICIDS-2017 dataset and compared their performance metrics.

3- System Model

The proposed DML-IDS: Distributed Multi-Layer Intrusion Detection System introduces a distributed and multi-layer approach to detect cyber threats. The proposed DML-IDS framework consists of: (i) a Master Node for coordinating the ensemble models running on different networks, (ii) a Firewall in which the proposed multi-layer IDS is implemented, and (iii) Healthcare Resources. Figure 1 depicts the overall architecture of the proposed DML-IDS framework.

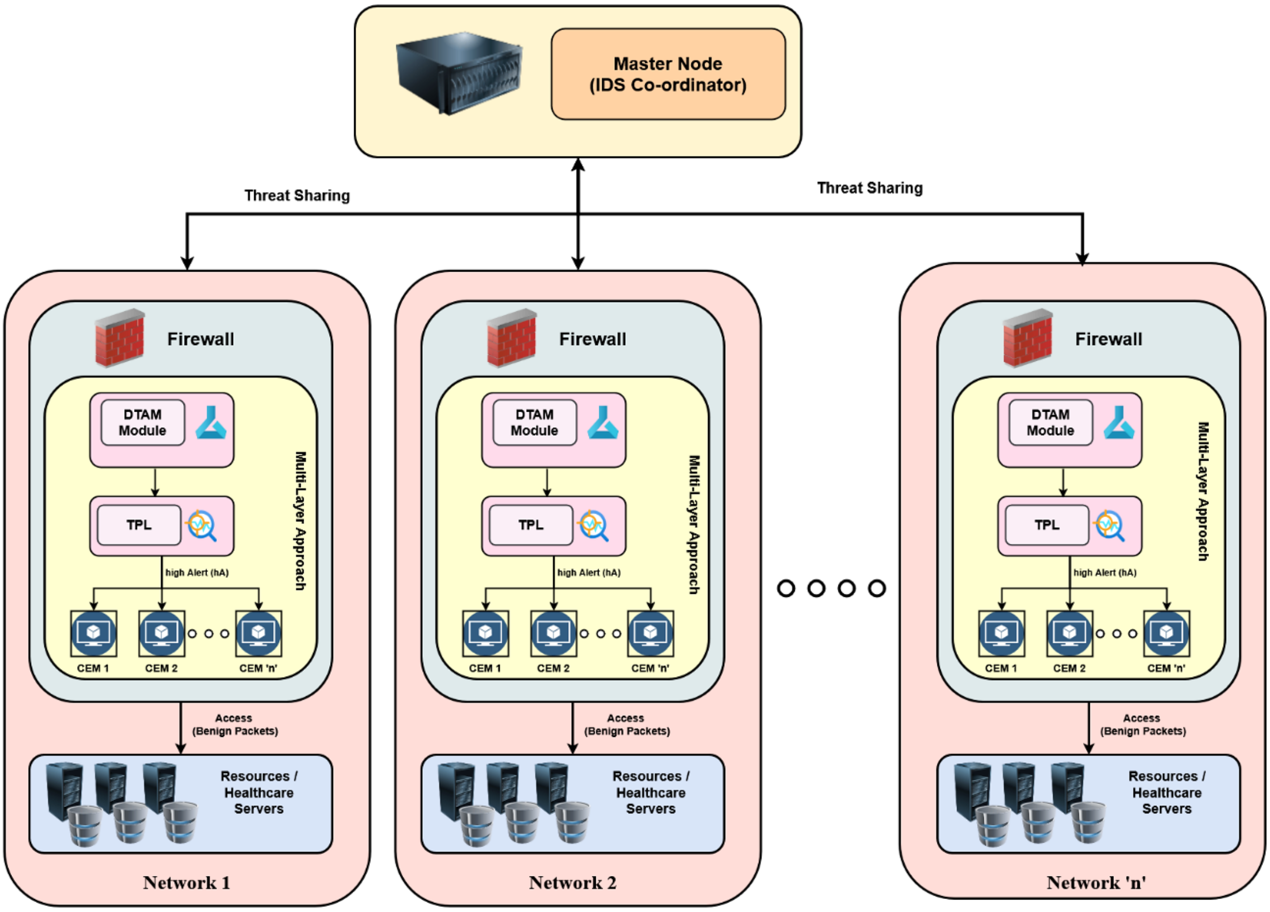


Figure 1. Overall architecture of the proposed DML-IDS framework

Master Node (IDS-Co-Ordinator): Acts as a trusted supervisory entity in the distributed IDS framework. The primary responsibilities of the Master Node are to: (i) coordinate the firewalls deployed in distributed and heterogeneous healthcare networks, (ii) issue alerts immediately to connected firewalls upon detecting suspicious activities in the network, and (iii) synchronize IDS rules and detection models across interconnected firewalls to maintain uniform security standards.

Firewall: Responsible for executing the multi-layer intrusion detection model, which detects harmful or highAlert packets entering the network. The firewall consists of: (i) the Distributed Threat Analysis Module (DTAM), which runs base classifiers including SVM, Random Forest, and Logistic Regression for preliminary threat evaluation, (ii) the Threat Prioritization Layer (TPL), which identifies highAlert packets, and (iii) the Confirmatory Ensemble Model (CEM), which performs attack-specific analysis.

Healthcare Servers: Centralized systems within the network that host medical services and store confidential healthcare data. These servers are prime targets for cyber attackers and require continuous monitoring.

3-1- Dataset Information

In this research work, the CICIDS-2017 dataset is used for training and testing the proposed DML-IDS: Distributed Multi-Layer Intrusion Detection System. CICIDS-2017 is widely used in cybersecurity research, particularly for the development and evaluation of Intrusion Detection Systems (IDS). It contains real-life network traffic events, including various types of attacks such as DoS, DDoS, brute force, botnets, web attacks, and infiltration. The dataset consists of 80 features and approximately 3 million network transaction records.

The CICIDS-2017 dataset includes network traffic from cyber-attacks such as: (i) DDoS Attacks, (ii) Brute Force Attacks (SSH and FTP), and (iii) Web-based Attacks (XSS, SQL Injection, and Command Injection). Details of each cyber-attack and the corresponding network information are presented in Table 1.

The CICIDS-2017 dataset, which includes diverse attack types, serves as a reliable and validated source for training and testing the proposed DML-IDS system.

Table 1. Dataset Information

Dataset Info	Records focusing on DDoS Attack	Records focusing on Brute Force Attacks (SSH and FTP)	Records focusing on Web based attacks (XSS, SQL injection and command injections)
Total Records	225752	445910	170365
Number of Training data	180682	356781	136453
Number of Testing data	45142	89123	33917
Total features	79	79	79
Features chosen for Training	24	20	20
Number of Attack Categories	1	2	3
Percentage of MD+IA	1.42	0.25	1.09

3-2- Dataset Information

To reduce inconsistencies and achieve better accuracy in the proposed DML-IDS system, preprocessing techniques such as: (i) handling missing, infinite, and large values, (ii) categorical encoding, and (iii) feature scaling are applied.

Missing, infinite, and large values are identified and removed from the dataset. Later, label encoding is performed to convert categorical values into a numerical format. The StandardScaler library is then used to standardize the data by removing the mean and scaling to unit variance.

After preprocessing, the CICIDS-2017 dataset is split into an 80:20 ratio, where 80% is used for training the model and 20% for testing.

3-3- Feature Extraction

To achieve better accuracy, it is important to select the most relevant features from the dataset. As the proposed DTAM model handles multiple types of cyber-attacks, it is crucial to identify suitable features for training. In the proposed work, the SelectKBest algorithm is applied, and for each specified cyber-attack, the top 20 features are identified and extracted.

The SelectKBest method finds the k most important features with the highest scores, as assessed by statistical measurements. Each feature is evaluated using a specific statistical test, such as f_regression for regression tasks or chi-square for classification tasks.

Let, $\mathrm{D_i} = \left\{ \left(\mathrm{x}_1^{(\mathrm{i})}, \mathrm{y}^{(\mathrm{i})}\right), \left(\mathrm{x}_2^{(\mathrm{i})}, \mathrm{y}^{(\mathrm{i})}\right), \ldots, \left(\mathrm{x}_\mathrm{n}^{(\mathrm{i})}, \mathrm{y}^{(\mathrm{i})}\right) \right\}$ be the dataset for the i^{th} cyber-attack category where $\mathrm{x}_\mathrm{j}^\mathrm{i} \in \mathbb{R}^\mathrm{m}$ is the feature vector and $\mathrm{y}_\mathrm{j}^{(\mathrm{i})}$ is the corresponding label. $\mathrm{F_i} = \{ \mathrm{f}_1^\mathrm{i}, \mathrm{f}_2^\mathrm{i}, \ldots, \mathrm{f}_\mathrm{m}^\mathrm{i} \}$ be the set of all features in dataset D_i

A statistical scoring function $S(f)$ is used to assign a relevance score to each feature,

$$S\left(f_j^i\right) = \; score\ of\ feature\ f_j^i based\ on\ D_i \tag{1}$$

To choose the k features using SelectKBest algorithm,

$$F_{selected}^i = \arg top - k\ \ S\left(f_j^i\right), \text{for } \mathrm{f_j} \in \ \mathrm{F_i} \tag{2}$$

For each cyber-attack, the dataset D_i is, $\mathrm{F}_{\mathrm{Selected}}^{\mathrm{i}} = \{\mathrm{f}_1^\mathrm{i}, \mathrm{f}_2^\mathrm{i}, \ldots, \mathrm{f}_\mathrm{k}^\mathrm{i}\}$

SelectKbest feature extraction algorithm selects the ‘k’ features that score the highest. The primary objective of this algorithm is to improve the performance of the machine learning model by reducing the dimensionality of the data. This helps to control overtraining, increase training speed, and reduce the curse of dimensionality. To develop a DTAM model, which detects all types of cyber-attacks, a common feature in each dataset is extracted and combined as a unified dataset.

Let, $\mathrm{A} \; = \left\{ \mathrm{A}_1, \mathrm{A}_2, \ldots, \mathrm{A}_\mathrm{p} \right\}$ be the set of all types of cyber-attacks, $\mathrm{F}_{\mathrm{Selected}}^{\mathrm{I}}$ be the top k features from each A_i. Then, the common feature set used to train the combined DTAM model is,

$$\mathrm{F}_{\mathrm{unified}} = \; \cap_{\mathrm{i}=1}^{\mathrm{p}} \mathrm{F}_{\mathrm{selected}}^{\mathrm{i}} \tag{3}$$

The DTAM model is trained using $\mathrm{F}_{\mathrm{unified}}$ set to efficiently identify all types of cyberattacks. Figure 2 shows the extracted features from the CICISD 2017 data using the SelectKbest algorithm.

The ANOVA F-value quantifies the ratio of variance between the groups to the variance within the groups. For a target variable Y and a feature X:

$$F \; = \frac{variance\ between\ groups}{variance\ within\ groups} = \frac{\frac{1}{k-1}\sum_{k=1}^{k} n_k(\bar{X}_k - \bar{X})^2}{\frac{1}{N-K}\sum_{k=1}^{k}\sum_{i=1}^{n_k}(X_{ik} - \bar{X}_k)^2} \tag{4}$$

where, k is the number of groups, n_k is the number of samples in the group k, $\bar{X}_k$ is the mean of the group k , and $\bar{X}_k$ is the overall mean. Figure 2 presents the features that are extracted from CICIDS-2017 dataset for training using SelectKbest method.

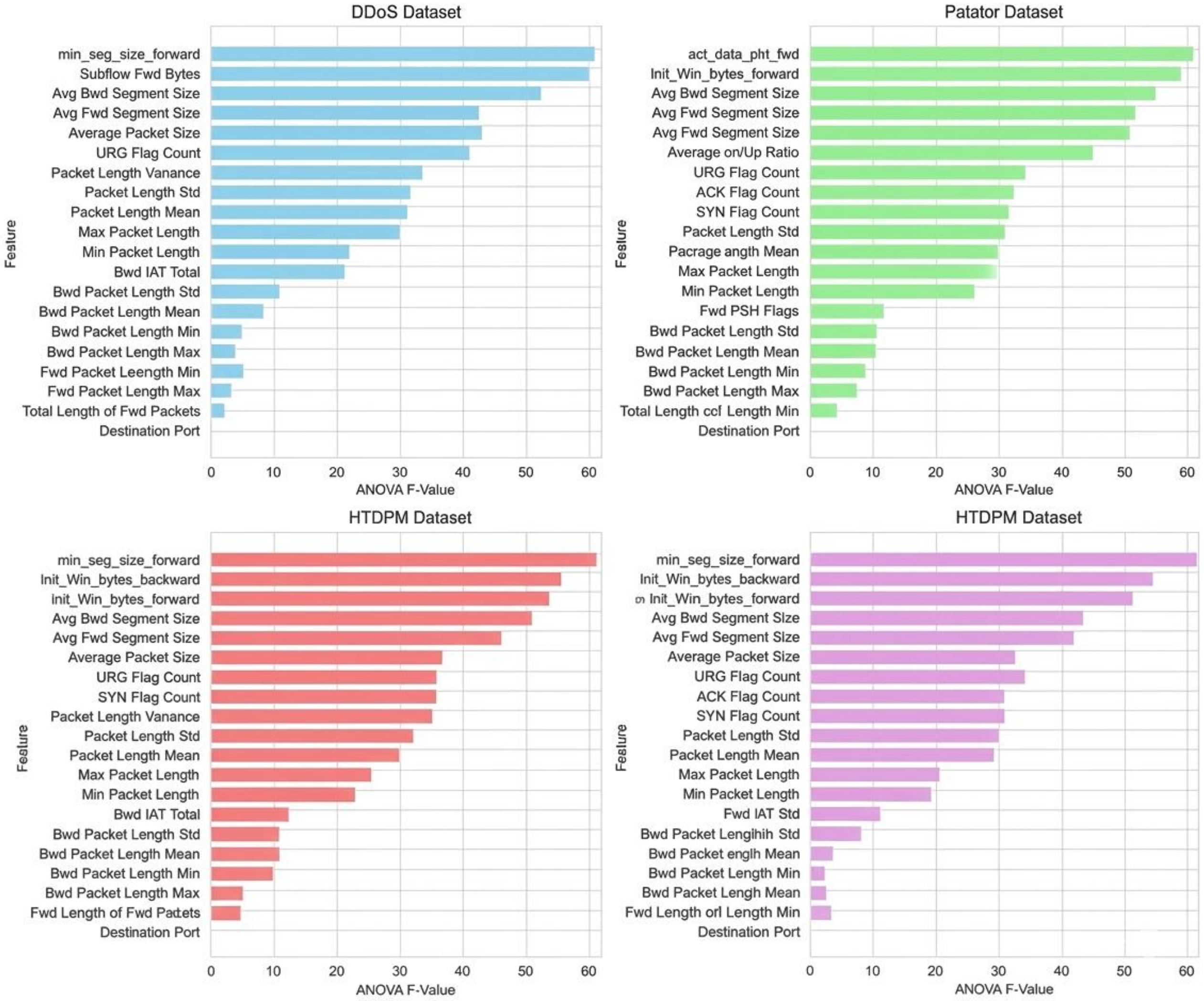


Figure 2. Features extracted for training using SelectKbest method

By identifying the most important features for each type of cyber-attack, the SelectKBest algorithm reduces the dimensionality of the data, which improves both accuracy and performance of the DTAM model in detecting various cyber threats.

3-4- DML-IDS Multi-layer Approach

The proposed DML-IDS: Distributed Multi-Layer Intrusion Detection System includes three distinct stages: (i) Distributed Threat Analysis Module (DTAM), (ii) Threat Prioritization Layer (TPL), and (iii) Confirmatory Ensemble Models (CEM). The DTAM, located in the firewall's first layer, processes all incoming network packets and performs an initial threat assessment using the Random Forest machine learning algorithm. If any suspicious activity is detected, the DTAM module forwards the packets to the Threat Prioritization Layer (TPL). In this layer, malicious packets are flagged as highAlert (hA) packets based on the type and severity of the detected threat.

These hA packets are then sent to the appropriate Confirmatory Ensemble Models (CEMs) for in-depth, attack-specific analysis. The CEMs are designed to detect zero-day and multi-vector threats and are built to be scalable. This layered approach enables efficient detection of various types of cyber-attacks without the need to deploy separate models for each attack type, thereby reducing the computational load.

3-4-1- Distributed Threat Analysis Module (DTAM)

The DTAM module examines all incoming packets entering the network that contains healthcare resources. To classify packets as malicious or benign, the DTAM is trained using the Random Forest machine learning algorithm. A unified dataset, consisting of common features across all attack types, is used to train the model so that it can detect multiple types of cyber-attacks rather than being restricted to a single attack category.

$$D_{unified} = \bigcup_{i=1}^{P} F_{selected}^{(i)} \tag{5}$$

where, $F_{selected}^{(i)}$ is the set of top features selected from the dataset $D_i$corresponding to the attack type i and p is the total number of different types of attack type considered.

The shared characteristics of all the individual attack datasets are extracted and consolidated into a unified dataset. This dataset is then used to train the model, enabling it to anticipate and detect various forms of web attacks rather than being limited to a single attack type. The model is trained using the unified dataset as follows,

$$M_{DTAM} = Train_{RF}\ (D_{unified}) \tag{6}$$

where, M_{DTAM} represents the trained model, and $Train_{RF}$ denotes the training process of Random Forest algorithm. The Random Forest algorithm used in the DTAM model combines multiple decision trees to achieve better accuracy and to reduce the overfitting issues in the trained model. From the unified dataset $D_{unified}$, T bootstrap samples $D_1, D_2 \dots D_T$ are generated by the sampling with replacement. Each sample is used to train one decision tree.

At each node, a random subset of features $F_t \subset \{1,2,\dots,m\}$ is selected. The best feature and threshold to split the node are determined using a criterion such as Gini Impurity, calculated as,

$$G(N) = 1 - \sum_{c=1}^{C} p_c^2 \tag{7}$$

where p_c is the proportion of instances belonging to class c at node N, and C is the number of classes. Each trained decision tree $h_t$provides a prediction for a given input x, $h_t(x) = \in \{0,1\}$, where 0 denotes benign and 1 denotes malicious. The final DTAM model us majority vote among all decision trees,

$$H(x) = mode\{h_1(x), h_2(x), \dots, h_T(x)\} \tag{8}$$

The majority voting mechanism increases the model's robustness and accuracy by reducing the impact of individual tree errors.

3-4-2- Threat Priority Layer

The second layer in the proposed DML-IDS: Distributed Multi-Layer Intrusion Detection System is the Threat Prioritization Layer (TPL). Network packets identified as malicious by the DTAM model are forwarded to the TPL for further evaluation based on their severity and potential impact. The primary goal of the Threat Prioritization Layer is to assign priority levels to the identified network packets, enabling the system to respond more quickly to high-risk attacks.

The TPL uses a severity scoring function $S(x)$ to calculate the threat score for each network packet that is flagged as a highAlert packet by the DTAM model. The score is calculated from the key threat indicators such as, packet size, source reputation, port access pattern, frequency of attack signature and type of protocol.

Let, $x = \{f_1, f_2, \dots, f_k\}$ be the feature vector of a flagged packet. The severity scores $S\ (x)$ is computed as,

$$S(x) = \sum_{j=1}^{k} w_j . f_j \tag{9}$$

where f_j is the j^{th} selected feature of the packet, w_j the weight assigned to feature f_j and k is the total number of features used for priority classification.

A threshold θ is defined to classify the identified packets as highAlert (hA) packets,

$$If\ S(x) \geq \theta, x \in highAlert\ (hA)\ set \tag{10}$$

Network packets that exceed the threshold value are designated as high-priority packets and forwarded to the third layer, called the Confirmatory Ensemble Models (CEMs), for advanced and attack-specific analysis.

3-4-3- Confirmatory Ensemble Models

The third layer in the proposed DML-IDS: Distributed Multi-Layer Intrusion Detection System is the Confirmatory Ensemble Models (CEMs), in which multiple models are trained as an ensemble. The primary task of the CEM is to analyze the highAlert packets received from the Threat Prioritization Layer (TPL) in a targeted, attack-specific manner. These models are developed as specialized ensembles that achieve high accuracy against specific types of cyberattacks, such as web-based attacks, DDoS attacks, and Patator attacks. Each Confirmatory Ensemble Model includes the following base classifiers: (i) Random Forest (RF), (ii) Support Vector Machine (SVM), and (iii) Naïve Bayes (NB), as shown in Figure 3.

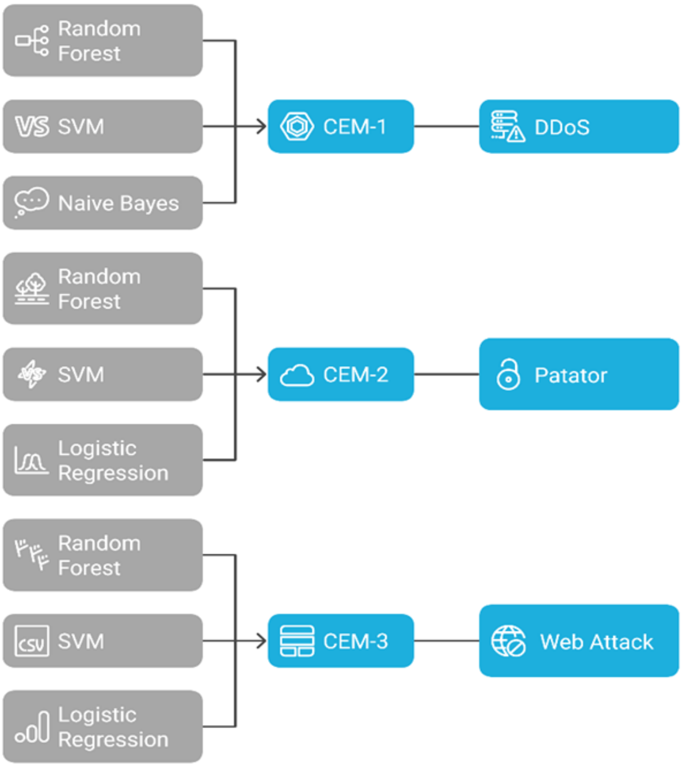


Figure 3. **Confirmatory Ensembled Model to detect different types of cyber attack**

The outputs of these classifiers are combined using majority voting or another fusion strategy. This process ensures that the system can confirm whether an incoming packet is truly malicious and determines its specific attack type.

In the third layer, CEM-1 is responsible for verifying DDoS attacks by combining Random Forest, SVM, and Naive Bayes. CEM-2 handles Patator attacks using Random Forest, SVM, and Logistic Regression and CEM-3 focuses on Web Attacks, using the same base classifiers as CEM-2.

Let the prediction of classifier c_j in CEM_i for input x be $c_j^i(x)$, where $j = 1,2, \ldots n$ and $i = 1,2,3$. Then the final CEM decision is computed as,

$$CEM_i^{(x)} = mode\{c_1^i(x), c_2^i(x), \ldots, c_n^i(x)\} \tag{11}$$

where, $CEM_i^{(x)}$ gives the final label (benign or malicious for the specific attack type) and mode represents majority voting among the classifier predictions.

4- Performance Evaluation

The proposed DML-IDS: Distributed Multi-Layer Intrusion Detection System was trained on an HPC machine with the following configuration: Intel Xeon 4210 processor, 32 GB RAM, running on Ubuntu 22.04 LTS. Python 3.11 was used for model training. The experimental setup validated the process of training parallel ensemble models and efficiently managing large datasets. Table 2 shows the hyperparameters that were used for training the ensemble learning models.

Table 2. **Hyperparameters used for training the model**

Model	Tuning Parameters	Values
Random Forest	n_estimators	100
	max_depth	10
	min_samples_split	2
SVM	Kernel	rbf
	Regularization Parameter (C)	1
	Gamma	'scale'
Logistic Regression	Regularization parameter	1
	Solver	ibfgs
	Max Iterations	100
Naïve Bayes	Default Parameters	Default Parameters

In the proposed work, Random Forest, Support Vector Machine, Logistic Regression, and Naïve Bayes algorithms are used for training the model. For the Random Forest algorithm, the number of trees is set to 100, the tree depth to 10, and the minimum sample split to 2. For the Support Vector Machine algorithm, the kernel type is set as RBF, the regularization parameter CC is set to 1, and the gamma value is set to 'scale'. For the Logistic Regression algorithm, the regularization parameter CC is set to 1, the solver is set as 'lbfgs', and the maximum number of iterations is set to 100. For the Naïve Bayes algorithm, the default parameter settings are used.

4-1- Evaluation of Distributed Threat Detection Model (DTAM)

The first layer of the proposed DML-IDS: Distributed Multi-Layer Intrusion Detection System is the Distributed Threat Analysis Module (DTAM). In this layer, incoming network packets are classified as either malicious or benign. The model is trained using the $D_{unified}$ dataset with the Random Forest algorithm. The trained model is evaluated using standard performance metrics such as accuracy, precision, and F1-score.

The accuracy of the DTAM is calculated through, $Accuracy_{DTAM} = \frac{Correct\ Detection}{Total\ Packets} = \frac{TP+TN}{TP+TN+FP+FN}$

Similarly, the precision of the DTAM is calculated from, $Precision_{DTAM} = \frac{Correct\ Detected\ Malicious}{All\ Predicted\ Malicious} = \frac{TP}{TP+FP}$

F1 Score is calculated as, $F1_{DTAM} = \frac{2*Precision*Recall}{Precision+Recall}$

The accuracy, precision and F1 score of the proposed DTAM model is shown in Figure 4.

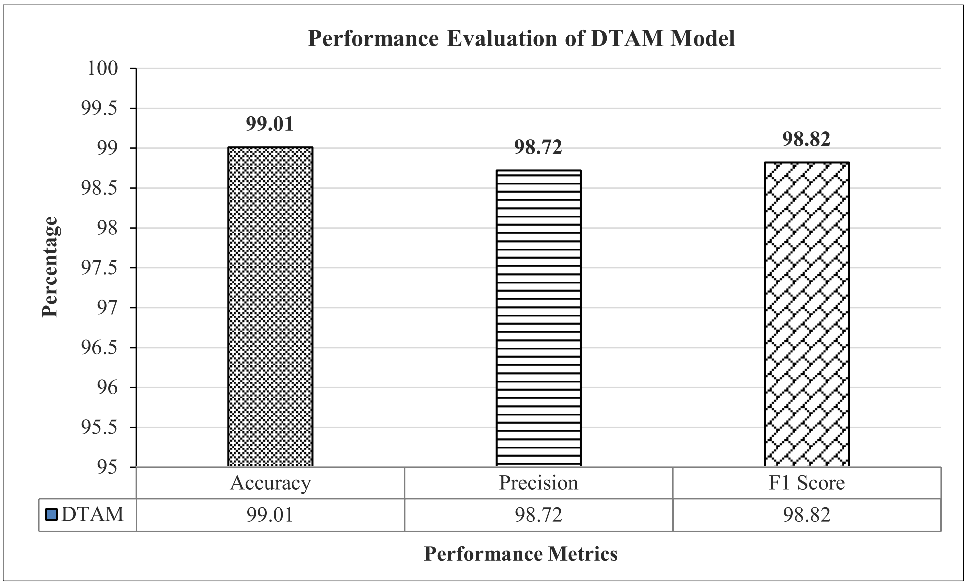


Figure 4. Performance evaluation of DTAM model with $D_{unified}$

The DTAM model, trained using the Random Forest algorithm on the unified dataset $D_{unified}$, directly monitors traffic passing through the network tunnel (firewall). It classifies traffic as normal or malicious. If an attack is suspected, the traffic is forwarded to the Threat Prioritization layer (TPL) for higher-level assessment.

4-2- Threat Prioritization Layer (TPL)

The DTAM layer acts as a primary filter to detect potential attacks. It applies a Random Forest (RF) algorithm to classify the malicious internet network packets and forwards it to the Threat Prioritization Layer (TPL), where the severity of the packets is evaluated.

Let a detected threat TPL_i be evaluated by the Threat Prioritization Layer based on multiple parameters such as, Severity score (S_i), Frequency of Occurrence (F_i), Classifier confidence score (C_i) and Risk impact score (R_i).

Let weight be assigned to each factor:

$$w_s, w_f, w_c, w_r \in [0,1] \text{ and } w_s + w_f + w_c + w_r = 1 \tag{12}$$

Then the TPL of threat TPL_i is defined as,

$$TPL_i = w_s \cdot S_i + w_f \cdot F_i + w_c \cdot C_i + w_r \cdot R_i \tag{13}$$

Here, the value of S_i, F_i , C_i, R_i are normalized to [0,1],

$0 \leq S_i, F_i, C_i, R_i \leq 1$

Combining with weights $w_s + w_f + w_c + w_r = 1$, the convex combination be,

$$TPL_i = \sum_{j\in\{s,f,c,r\}} w_j \cdot X_j \,, where\ X_j \in [0,1] \tag{14}$$

Using the properties of convex combination of bounded values

Lower bound: if all $X_j = 0$, then $TPL_i = 0$ and

Upper bound: if all $X_j = 1$, then $TPL_i = w_s + w_j + w_c + w_r = 1 \Rightarrow 0 \leq TPL_i \leq 1$

4-2-1- Analysis of Threat Prioritization Layer (TPL)

In the proposed DML-IDS architecture, the second layer, Threat Prioritization layer analyzes network packets that have been classified as benign packets by the DTAM layer. For each such packet, the TPL_i is computed using the weighted combination of severity score, frequency of occurrence, classifier confidence, and risk impact, as defined in Equation 12. To evaluate the performance of the TPL layer, the Friday DDoS Day subset of CICIDS-2017 dataset was used. It consists of 225,745 records, in which 128,027 are DDoS attack packets and 97,718 benign packets.

The first layer, DTAM correctly identified 126,761 packets as malicious and 98,984 packets as benign with the accuracy of 99.01%. However, to further tighten the security and to improve the accuracy of the detection system, the second layer, Threat Prioritization Layer calculates the threat value TPL_i of each benign packet.

The TPL layer evaluated all DTAM benign packets using the threshold value τ = 0.70. This process identified 5,867 benign-classified packets with threat score exceeding the threshold value. Table 3 presents the detail of the analysis of Threat Prioritization Layer.

Table 3. Analysis of Threat Prioritization Layer

Metric	Count
Total Packets	225,745
DTAM – Malicious (auto-HighAlert)	126,761
DTAM-benign	98,984
TPL-promoted (Score > 0.70)	5,867
Recovered true attacks	1,726
Benign promoted as hA packet	4,141
Final High Alert packets	131,989
Residual undetected attacks	179
Total Packets	225,745
DTAM – Malicious (auto-HighAlert)	126,761

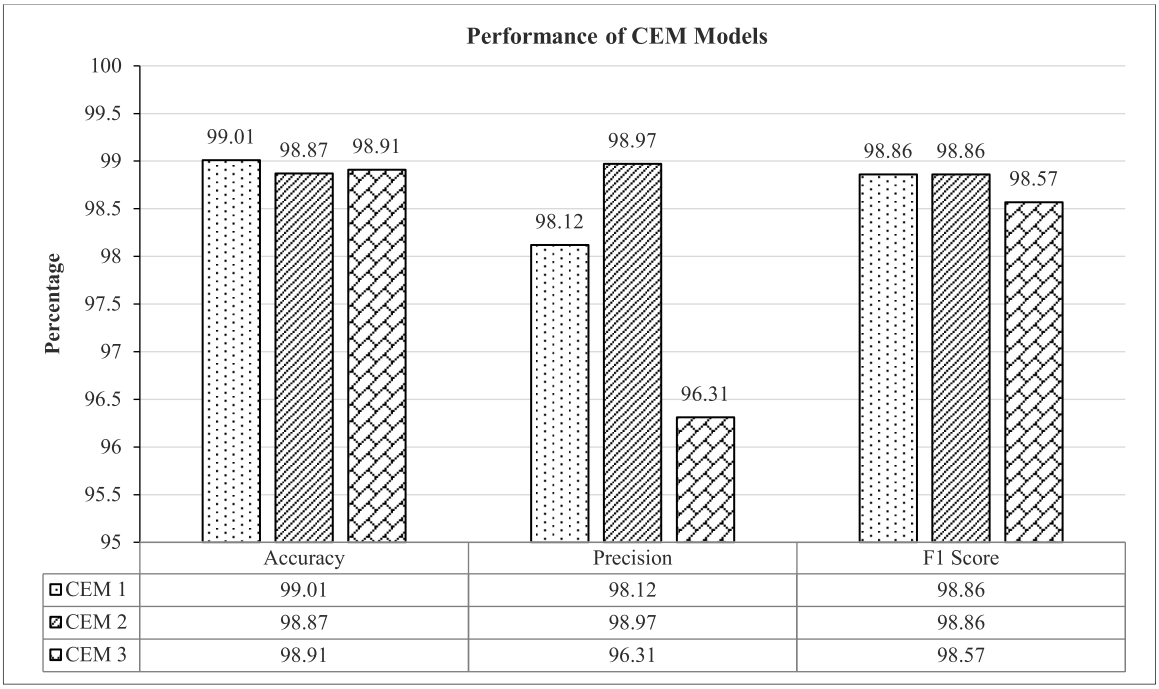
The malicious packets classified by the DTAM model and the packets that have threshold value of more than 0.70 are flagged as high Alert (hA) packets and are fed to Confirmatory Ensemble Models (CEMs) for further analysis.

4-3- Confirmatory Ensemble Models

The Voting Classifier technique is applied to Confirmatory Ensemble Models (CEMs). Each base model is combined using a hard voting mechanism, where the predictions of multiple models are aggregated, and the majority vote determines the final output. This approach enhances both the accuracy and reliability of the model. As a result, the CEMs can detect and prevent attacks with greater accuracy and precision. Table 4 summarizes the overall performance results of the CEMs models based on the type of attack and Figure 5 depicts the accuracy, precision of F1 Score of the CEM model.

Table 4. Performance evaluation of CEM Models proposed in the DML-IDS Framework

Metric	CEM 1 DDoS	CEM 2 Patator	CEM 3 Web Attacks
Accuracy	99.01	98.87	98.91
Precision	98.12	98.97	98.31
F1-Score	98.86	98.86	98.57
Missed Detection (MD)	15	15	309
Incorrect Alarm (IA)	875	210	59
Total Test Values	45143	89120	33912
Percentage of MD+IA	1.43	0.26	1.09



	Accuracy	Precision	F1 Score
□ CEM 1	99.01	98.12	98.86
▨ CEM 2	98.87	98.97	98.86
□ CEM 3	98.91	96.31	98.57

Figure 5. Performance evaluation of CEM Models

The accuracy, precision and F1-score of each CEMs are measured. The results are promising with good accuracy and only a low level of false alarms and missed detection is observed. The detailed results of the proposed CEM models with respect to each attack type and algorithm used are denoted in Table 5.

Table 5. Performance evaluation of CEM Models proposed in the DML-IDS Framework

Attack	Algorithm Used	Accuracy	Precision	F1-score	Incorrect Alarm (IA)	Missed Detection (MD)	Total IA+MD	Total Values	Percentage
DDoS	Random Forest Classifier	99.89	99.99	99.96	1	18	19	45143	0.04
	SVM	98.32	97.32	98.62	708	11	719		1.59
	Naïve Bayes Classifier	98.12	96.85	98.38	835	9	844		1.87
Patator	Random Forest Classifier	99.05	99.99	99.99	0	0	0	89129	0
	SVM	98.90	98.9	98.91	230	8	238		0.27
	Logistic Regression	98.87	98.89	98.77	286	9	295		0.33
Web Attacks	Random Forest Classifier	99.24	99.33	99.3	52	49	101	33912	0.3
	SVM	98.73	98.2	98.12	61	309	370		1.09
	Logistic Regression	98.56	97.82	98.05	69	309	378		1.11

4-4-Comparison of the Proposed DML-IDS System with Existing Work

To improve the accuracy of cyber threat detection, several researchers have explored ensemble-based machine learning approaches using the CICIDS-2017 dataset. Abbas et al. [25] achieved 88.96% accuracy, indicating limitations in performance. Meanwhile, Zhou et al. [26] and S. Das et al. [27] achieved higher accuracies of 97.89% and 98.50%, respectively; however, their models can be considered limited, as both employed single-layer detection approaches with a small number of features.

Similarly, Thokchom et al. [28] and Mhawi et al. [29] tested their models using a limited set of attack types, achieving 99.48% and 99.7% accuracy, respectively, but the lack of comprehensive attack coverage remains a limitation. Maseed et al. [30] achieved 98.9% accuracy using a Random Forest classifier; however, their approach focused on only a few specific attack types and did not incorporate an ensemble method. Table 6 presents a comparison between the existing works and the proposed DML-IDS: Distributed Multi-Layer Intrusion Detection System.

Table 6. Comparative Analysis of the proposed work with existing work

Article	Algorithm Used	Accuracy	Remark
Abbas et al. [25]	Ensemble	88.96	Limited Accuracy
Zhou et al. [26]	Ensemble	99.89	Single Layer Approach and Lower Number of Features
Das et al. [27]	Ensemble	99.50	Less features used for training, single layer approach
Thockchom et al. [28]	Ensemble	99.48	Accuracy Persistent to limited attacks, some attacks are not detected
Mhawi et al. [29]	Ensemble	99.7	Single Layer Approach, only limited attacks taken into consideration
Maseet et al. [30]	Random Forest Classifier	98.9	No Ensembling and model limited to few types of attacks only
Proposed Research	Random Forest Classifier and Ensemble	99.01	Multi-Layer approach, high Accuracy produced, and can detect various attacks, integrated honeypot mechanism, also works as in attack prevention

The present study improves upon previous work by combining Random Forest and ensemble models within a multilayer architecture. This design achieves a high accuracy of 99.01% and enhances the model’s capability to detect a wide range of attacks.

4-5- Analysis of Computational Overhead of Proposed DML-IDS System with Existing Work

One of the primary objectives of the proposed DML-IDS architecture is to reduce the computational overhead of threat detection. The proposed model uses a multi-layer and distributed approach to reduce the computational overhead. To evaluate the computational efficiency of the proposed architecture, system resource usage and network packet processing time are measured.

4-5-1- Analysis of Network Packet Processing Time

The processing time of the DML-IDS framework was evaluated by measuring the average time taken to handle incoming network packets. A total of 200,000 network packets is taken into consideration to measure the processing time. Initially, the first 10,000 packets were processed in 0.259 seconds, and it gradually increased to 4.042 seconds for 160,000 packets. Table 7 presents the processing time of 200,000 network packets.

Table 7. Processing Time Taken to implement 200,000 network packets

Packets	DTAM Time (s)	TPL Time (s)	CEM Time (s)	Total Time (s)
10000	0.092	0.048	0.119	0.259
20000	0.184	0.096	0.211	0.491
30000	0.276	0.144	0.317	0.737
40000	0.368	0.192	0.427	0.987
50000	0.46	0.24	0.499	1.199
60000	0.552	0.288	0.678	1.518
70000	0.644	0.336	0.732	1.712
80000	0.736	0.384	0.967	2.087
90000	0.828	0.432	0.981	2.241
100000	0.92	0.48	1.189	2.589
110000	1.012	0.528	1.208	2.748
120000	1.104	0.576	1.427	3.107
130000	1.196	0.624	1.546	3.366
140000	1.288	0.672	1.664	3.624
150000	1.38	0.72	1.783	3.883
160000	1.472	0.768	1.802	4.042
170000	1.764	0.816	2.631	5.211
180000	1.756	0.864	2.994	5.614
190000	1.948	0.912	3.759	6.619
200000	2.484	0.96	3.95	7.394

After reaching 1,60,000 packets, there was a noticeable rise in the processing time. The Confirmatory Ensemble Models (CEMs) layer, where several machine learning models run simultaneously, handles high-alert (hA) packets, introducing computational cost that is responsible for this abrupt spike. The total time taken to process 200,000 network packets is shown in Figure 6.

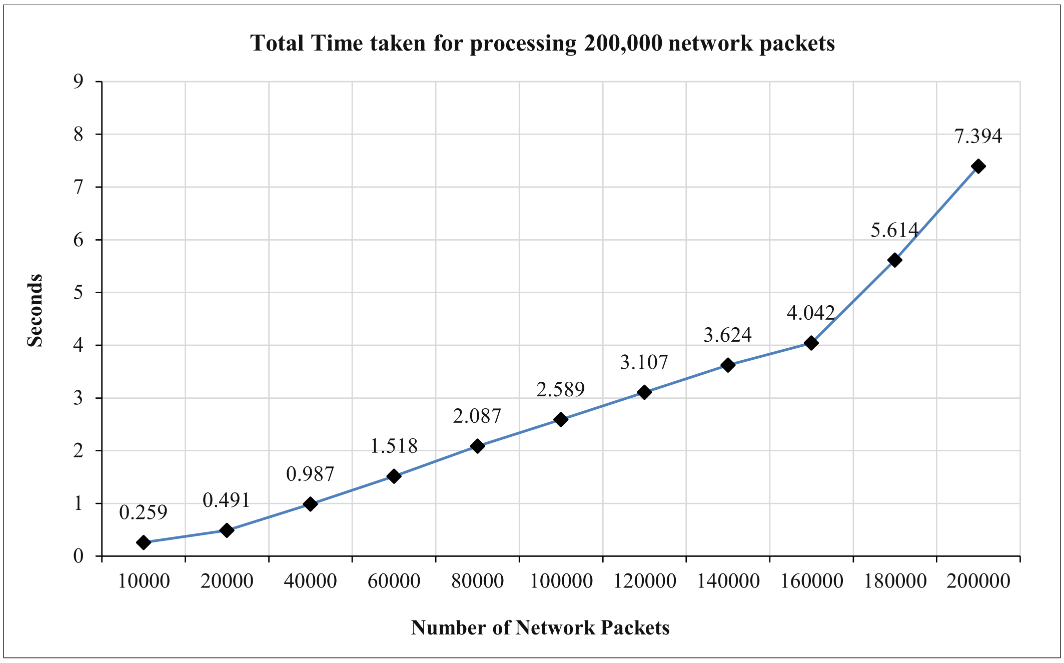


Figure 6. Time taken to process 200,000 network packets

In contrast to traditional IDS models, which sometimes show exponential increases in processing delays as traffic volume rises, the suggested DL-IDS exhibits noticeably higher efficiency. The two-layer method is largely responsible for this efficiency since it guarantees that only questionable packets are thoroughly examined at the CEM layer, which lowers the processing load overall.

4-5-2- Resource Utilization

To further evaluate the efficiency of the proposed DML-IDS architecture, the resource utilization for implementing DTAM, TPL, and CEM layers was measured. The CPU and memory usage were measured for each individual layer under a network workload of 200,000 packets. The DTAM layer and the TPL layer were implemented on the same machine, while the CEM models were implemented on a separate machine.

DTAM Layer: To measure the resource utilization of the DTAM layer, a dataset containing 200,000 input packets was fed into the model. The average CPU utilization for processing these network packets was 18.5%, with a peak utilization of 22%. Memory usage averaged 62.7%, and the total processing time was 2.88 seconds.

TP Layer: The packets that are classified as benign by the DTAM layer are reevaluated on the TPL layer. Out of 200,000 network packets, 98,128 packets were classified as benign network packets. To process these packets, the average CPU utilization is 20.1%, and the peak CPU utilization is 25.6%. Also, the average memory utilization is 66.9%, and the total time for processing the benign packets in the TPL layer is 1.12 seconds.

CEM Layer: The highAlert (hA) packets are further processed in this layer. As the CEM layer consists of multiple machine learning models, the resource utilization in the CEM layer is higher than the DTAM and TPL layers. The average CPU utilization is 42.7%, and the peak CPU utilization is 65.1%. The average memory utilization is 76% with a processing time of 4.12 seconds. Table 8 presents the CPU and memory utilization to process 200,000 network packets.

Table 8. Resource utilization to process 200,000 network packets of individual layer

Layer	Avg. CPU Utilization %	Peak CPU Utilization %	Avg. Memory Utilization %	Processing Time (s)
DTAM	18.5	22.0	62.7	2.88
TPL	20.1	25.6	66.9	1.12
CEM	42.7	65.1	76	4.12

Figure 7 illustrates the overall CPU and memory utilization of the proposed DML-IDS architecture to execute 200,000 network packets. Average CPU utilization remains relatively low, ranging from 18% to 25% across varying traffic loads, due to the multi-layer filtering that reduces unnecessary processing in later stages. Peak CPU utilization shows noticeable spikes at 150,000 packets (48.61%), corresponding to increased CEM activity when a higher number of High Alert packets are forwarded for parallel ensemble classification.

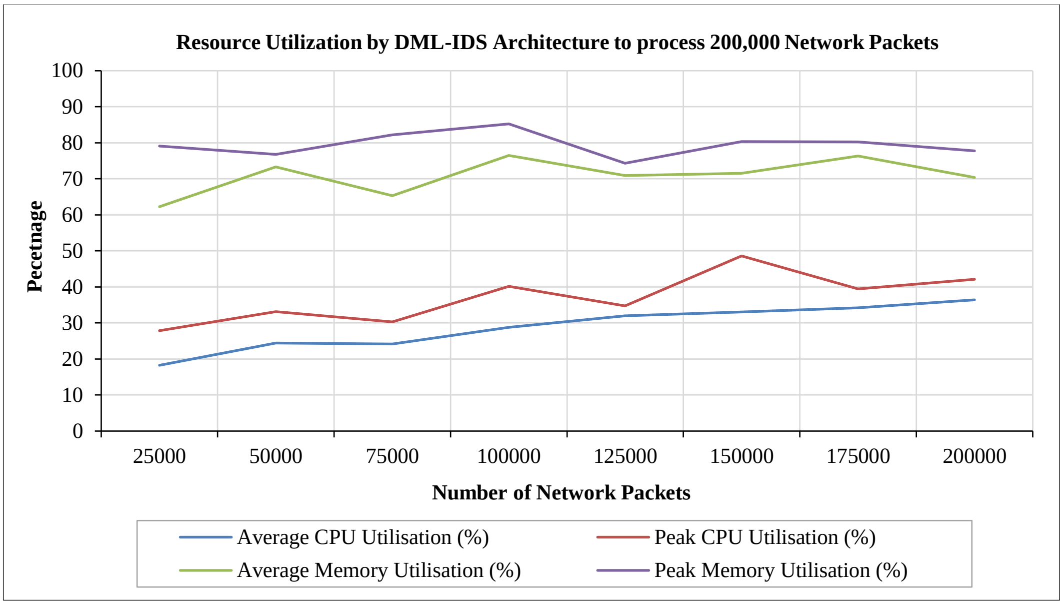


Figure 7. Resource Utilization to process 200,000 network packets

Average memory usage remains comparatively stable between 62% and 76%. Overall, the results confirm that while CPU demand fluctuates based on the volume of packets escalated to the CEM layer, memory usage stays steady across all layers, validating the scalability and efficiency of the DML-IDS in high-traffic environments.

4-5-3- Comparative Analysis with Existing Ensembled Machine Learning Based IDS

The computational overhead of the proposed DML-IDS scheme is compared with the conventional IDS models. Table 9 shows the comparative analysis of the computation overhead.

Table 9. Comparison of Computational Efficiency of existing ensembled technique with proposed architecture

Metric	Existing Ensembled Technique	Proposed Distributed and Multilayer based DML-IDS architecture
Processing Time (200K packets)	11.9 Sec	7.34 Sec
CPU Utilizations	40%-55%	20%-35%
Peak CPU Utilization	85%	75%
Memory Utilization (Avg)	75%-85%	70%-80%

The main reason for this improvement in the computational efficiency is because of the multi-layer approach, the DTAM and TPL layer forwards only the high-risk packets to the computationally demanding CEM layer after filtering out innocuous traffic.

5- Conclusion

The proposed DML-IDS: Distributed Multi-Layer Intrusion Detection System framework enhances cyber threat detection while minimizing computational overhead. The proposed framework integrates three function layers, such as the Distributed Threat Analysis Module (DTAM), the Threat Prioritization Layer (TPL), and Confirmatory Ensemble Models (CEM), to filter, prioritize, and verify the malicious network packets. Instead of running the IDS on a single machine, the proposed DML-IDS divides the IDS work into multiple layers and deploys it in a distributed network, which significantly improves the scalability and reduces the computational overhead on a single machine. The proposed DML-IDS model is trained with CICIDS-2017 data for detecting various types of cyber-attacks such as DDoS, Patator, and Web Attacks. The accuracy of the DTAM base classifier model is 98.5%, while the specialized CEMS models designed to detect DDoS, Patator, and Web attacks achieved 99.01%, 98.87%, and 98.91%, respectively.

Also, the computational efficiency of the proposed model is analyzed by evaluating the packet processing time and resource utilized. The multi-layer filtering strategy reduces unnecessary processing in later stages, allowing the system to maintain lower average CPU usage (18–25%) and stable memory consumption (62–76%). Compared to conventional IDS, the proposed DML-IDS reduced processing time for 200,000 packets by 28.32% and exhibited improved efficiency in both CPU and memory usage. The proposed DML-IDS model has achieved high detection accuracy and computational efficiency. The future work will focus on integrating federated learning and collaborative model training across multiple healthcare datasets to further improve the detection accuracy against evolving cyber threats. Also, explainable AI methods such as SHAP or LIME will be incorporated to enhance the interpretability of model decisions, enabling security analysts to better understand the detection outcomes.

6- Declarations

6-1- Author Contributions

Conceptualization, M.S.Y. and V.P.; methodology, M.S.Y.; software, M.S.Y.; validation, M.S.Y., V.P., and J.M.; formal analysis, M.S.Y.; investigation, M.S.Y.; resources, M.S.Y.; data curation, M.S.Y.; writing—original draft preparation, M.S.Y.; writing—review and editing, M.S.Y.; visualization, M.S.Y.; supervision, M.S.Y.; project administration, M.S.Y.; funding acquisition, V.P. All authors have read and agreed to the published version of the manuscript.

6-2- Data Availability Statement

The data presented in this study are available in the article.

6-3- Funding and Acknowledgements

The authors would like to thank Modern College of Business and Science, Muscat, Oman; for supporting this work by providing research grant.

6-4- Institutional Review Board Statement

Not applicable.

6-5- Informed Consent Statement

Not applicable.

6-6- Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

7- References

[1] Alder, S. (2025). Healthcare data breach statistics. The HIPAA Journal, Dallas, United States. Available online: https://www.hipaajournal.com/healthcare-data-breach-statistics/ (accessed on November 2025).

[2] Veriti. (2024). The State of Healthcare Cybersecurity 2025: A Veriti Research Report. Veriti, Tel Aviv, Israel. Available online: https://veriti.ai/wp-content/uploads/2024/12/The-State-of-Healthcare-Cybersecurity-2025-_-A-Veriti-Research-Report.pdf (accessed on November 2025).

[3] NSFOCUS. (2024). Over 300,000! GorillaBot: The new king of DDoS attacks. NSFOCUS, Beijing, China. Available online: https://nsfocusglobal.com/over-300000-gorillabot-the-new-king-of-ddos-attacks/ (accessed on November 2025).

[4] Bhati, B. S., & Rai, C. S. (2020). Analysis of Support Vector Machine-based Intrusion Detection Techniques. Arabian Journal for Science and Engineering, 45(4), 2371–2383. doi:10.1007/s13369-019-03970-z.

[5] Azam, Z., Islam, M. M., & Huda, M. N. (2023). Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis Through Decision Tree. IEEE Access, 11, 80348–80391. doi:10.1109/ACCESS.2023.3296444.

[6] Ahmed, U., Jiangbin, Z., Almogren, A., Khan, S., Sadiq, M. T., Altameem, A., & Rehman, A. U. (2024). Explainable AI-based innovative hybrid ensemble model for intrusion detection. Journal of Cloud Computing, 13(1), 150. doi:10.1186/s13677-024-00712-x.

[7] Alsolami, T., Alsharif, B., & Ilyas, M. (2024). Enhancing Cybersecurity in Healthcare: Evaluating Ensemble Learning Models for Intrusion Detection in the Internet of Medical Things. Sensors, 24(18), 5937. doi:10.3390/s24185937.

[8] doost, P. A., Moghadam, S. S., Khezri, E., Basem, A., & Trik, M. (2025). A new intrusion detection method using ensemble classification and feature selection. Scientific Reports, 15(1), 13642. doi:10.1038/s41598-025-98604-w.

[9] Fares, I. A., & Abd Elaziz, M. (2025). Explainable TabNet Transformer-based on Google Vizier Optimizer for Anomaly Intrusion Detection System. Knowledge-Based Systems, 316. doi:10.1016/j.knosys.2025.113351.

[10] Torre, D., Chennamaneni, A., Jo, J. Y., Vyas, G., & Sabrsula, B. (2025). Toward Enhancing Privacy Preservation of a Federated Learning CNN Intrusion Detection System in IoT: Method and Empirical Study. ACM Transactions on Software Engineering and Methodology, 34(2), 1–48. doi:10.1145/3695998.

[11] Nassreddine, G., Nassereddine, M., & Al-Khatib, O. (2025). Ensemble Learning for Network Intrusion Detection Based on Correlation and Embedded Feature Selection Techniques. Computers, 14(3), 82. doi:10.3390/computers14030082.

[12] Xu, Z., Wu, Y., Wang, S., Gao, J., Qiu, T., Wang, Z., ... & Zhao, X. (2025). Deep Learning-based Intrusion Detection Systems: A Survey. arXiv Preprint, arXiv:2504.07839. doi:10.48550/arXiv.2504.07839.

[13] Zhang, Y., Muniyandi, R. C., & Qamar, F. (2025). A Review of Deep Learning Applications in Intrusion Detection Systems: Overcoming Challenges in Spatiotemporal Feature Extraction and Data Imbalance. Applied Sciences (Switzerland), 15(3), 1552. doi:10.3390/app15031552.

[14] Mamatha, P., Balaji, S., & Anuraghav, S. S. (2025). Development of Hybrid Intrusion Detection System Leveraging Ensemble Stacked Feature Selectors and Learning Classifiers to Mitigate the DoS Attacks. International Journal of Computational Intelligence Systems, 18(1), 20. doi:10.1007/s44196-025-00750-6.

[15] Ataa, M. S., Sanad, E. E., & El-khoribi, R. A. (2024). Intrusion detection in software defined network using deep learning approaches. Scientific Reports, 14(1), 29159. doi:10.1038/s41598-024-79001-1.

[16] Amouri, A., Al Rahhal, M. M., Bazi, Y., Butun, I., & Mahgoub, I. (2024). Enhancing Intrusion Detection in IoT Environments: An Advanced Ensemble Approach Using Kolmogorov-Arnold Networks. 2024 International Symposium on Networks, Computers and Communications (ISNCC), 1–6. doi:10.1109/isncc62547.2024.10758956.

[17] Bibers, I., Arreche, O., & Abdallah, M. (2024). A comprehensive comparative study of individual ML models and ensemble strategies for network intrusion detection systems. arXiv Preprint, arXiv:2410.15597. doi:10.48550/arXiv.2410.15597.

[18] Liu, J., Simsek, M., Kantarci, B., Bagheri, M., & Djukic, P. (2022). Collaborative Feature Maps of Networks and Hosts for AI-driven Intrusion Detection. 2022 IEEE Global Communications Conference (GLOBECOM 2022), 2662–2667. doi:10.1109/globecom48099.2022.10000985.

[19] Lansky, J., Ali, S., Mohammadi, M., Majeed, M. K., Karim, S. H. T., Rashidi, S., Hosseinzadeh, M., & Rahmani, A. M. (2021). Deep Learning-Based Intrusion Detection Systems: A Systematic Review. IEEE Access, 9, 101574–101599. doi:10.1109/access.2021.3097247.

[20] Gao, X., Shan, C., Hu, C., Niu, Z., & Liu, Z. (2019). An Adaptive Ensemble Machine Learning Model for Intrusion Detection. IEEE Access, 7, 82512–82521. doi:10.1109/ACCESS.2019.2923640.

[21] Bringer, M. L., Chelmecki, C. A., & Fujinoki, H. (2012). A Survey: Recent Advances and Future Trends in Honeypot Research. International Journal of Computer Network and Information Security, 4(10), 63–75. doi:10.5815/ijcnis.2012.10.07.

[22] Titarmare, N., Hargule, N., & Gupta, A. (2019). An Overview of Honeypot Systems. International Journal of Computer Sciences and Engineering, 7(2), 394–397. doi:10.26438/ijcse/v7i2.394397.

[23] Verma, A. S., & Dubey, A. (2020). A Review on Honeypot Deployment. LJP London Journal of Research in Computer Science and Technology, 20(1), 1-10.

[24] Sharafaldin, I., Habibi Lashkari, A., & Ghorbani, A. A. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. Proceedings of the 4th International Conference on Information Systems Security and Privacy, 108–116. doi:10.5220/0006639801080116.

[25] Abbas, A., Khan, M. A., Latif, S., Ajaz, M., Shah, A. A., & Ahmad, J. (2022). A New Ensemble-Based Intrusion Detection System for Internet of Things. Arabian Journal for Science and Engineering, 47(2), 1805–1819. doi:10.1007/s13369-021-06086-5.

[26] Zhou, Y., Cheng, G., Jiang, S., & Dai, M. (2020). Building an efficient intrusion detection system based on feature selection and ensemble classifier. Computer Networks, 174, 107247. doi:10.1016/j.comnet.2020.107247.

[27] Das, S., Saha, S., Priyoti, A. T., Roy, E. K., Sheldon, F. T., Haque, A., & Shiva, S. (2022). Network Intrusion Detection and Comparative Analysis Using Ensemble Machine Learning and Feature Selection. IEEE Transactions on Network and Service Management, 19(4), 4821–4833. doi:10.1109/tnsm.2021.3138457.

[28] Thockchom, N., Singh, M. M., & Nandi, U. (2023). A novel ensemble learning-based model for network intrusion detection. Complex and Intelligent Systems, 9(5), 5693–5714. doi:10.1007/s40747-023-01013-7.

[29] Mhawi, D. N., Aldallal, A., & Hassan, S. (2022). Advanced Feature-Selection-Based Hybrid Ensemble Learning Algorithms for Network Intrusion Detection Systems. Symmetry, 14(7), 1461. doi:10.3390/sym14071461.

[30] Maseer, Z. K., Yusof, R., Bahaman, N., Mostafa, S. A., & Foozy, C. F. M. (2021). Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset. IEEE Access, 9, 22351–22370. doi:10.1109/access.2021.3056614.