International Journal of Applied Mathematics

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

OPTIMIZING CYBERSECURITY: A DUAL-PHASE ML ARCHITECTURE FOR DETECTION AND CLASSIFICATION OF NETWORK ATTACKS

[1]Pushpa Gopal Ambhore, [2]Dr. Hiren Dand, 3Dr. Ram Joshi

[1]JJTU Research Scholar, JJTU Jhunjhunu Rajasthan, India
[2]Research Guide (JJTU Rajsthan), Jhunjhunu Rajasthan
3Research Co-Guide (JJTU Rajsthan), Jhunjhunu Rajasthan, JSPM's RSCOE (Associate Prof.)
pushpa1rscoe@gmail.com, hiren.dand@mccmulund.ac.in, ramjoshi.comp@gmail.com

Abstract

The rapid evolution of cyber threats has underscored the critical need for robust, high-performance Intrusion Detection Systems (IDS). This research presents a comprehensive framework leveraging five machine learning algorithms—Decision Tree, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Naïve Bayes, and Random Forest—to detect and classify network intrusions using the NSL-KDD dataset. The proposed methodology encompasses packet capture via a WinPcap-based sniffer, feature extraction across IP, TCP, UDP, and ICMP headers, and extensive preprocessing to aggregate session-level records. From an initial set of 41 attributes, a data-gain–driven feature selection process yielded 35 highly discriminative features. Each algorithm was trained on five million labeled connections and tested on two million unseen records, evaluating Accuracy, Precision, Recall, and F1-Score. Comparative analysis demonstrated that Random Forest achieved superior performance (96.4% accuracy, 96.3% F1-Score), followed by SVM (93.7% accuracy) and Decision Tree (89.3% accuracy). Naïve Bayes and KNN trailed due to conditional independence assumptions and distance-based limitations, respectively. Confusion matrix analysis highlighted Random Forest's lowest False Negative rate, crucial for minimizing undetected attacks. Multi-class detection performance revealed ensemble and kernel methods excel in handling diverse intrusion categories, notably in challenging R2L and U2R attacks. Feature importance ranking underscored the significance of service- and connection-based metrics—such as srv_count and dst_host_srv_count—for effective discrimination. Time-performance evaluation found Naïve Bayes best suited for resource-constrained environments, while Random Forest balanced training complexity with fast inference. These findings suggest that ensembles of decorrelated trees, supported by targeted feature engineering, provide the most reliable IDS solution. The modular architecture and extensive benchmarking offer a scalable blueprint for industrial and cloud deployments. Future work will explore deep learning augmentations and adaptive online learning to address evolving threat landscapes and zero-day intrusion vectors.

Received: July 18, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Keywords: Intrusion Detection System, Network security, Naïve Bayes, SVM, Artificial Neural Network, KDDCUP99.

Introduction

The Intrusion Detection system is designed to specialize in identifying individual categories of cyberattacks—such as DoS (Denial of Service), U2R (User to Root), R2L (Remote to Local), or novel (unknown) threats. This is achieved through a sequential deployment of multiple lightweight subsystems, each dedicated to recognizing a specific attack type. This modular approach yields two primary benefits: first, each subsystem focuses only on a limited feature set tailored to its assigned threat category, thereby enhancing detection accuracy; second, the reduced complexity ensures each component remains computationally efficient and easily deployable.While this layered approach may traditionally introduce increased inter-module communication overhead, the proposed framework addresses this by ensuring each sub-phase operates independently, thereby minimizing interdependence and communication bottlenecks. Furthermore, redundant features across multiple sub-phases can reinforce detection capability without centralized coordination. Upon detecting a threat, any individual sub-phase can autonomously initiate a response based on predefined network security policies, effectively acting as a filter that isolates malicious activities early in the detection chain. This strategy enables faster threat mitigation and decreases the volume of data analyzed in later stages, optimizing overall system performance.

Importantly, sub-phases trained on different attack types may trigger unique responses, adapting dynamically to varied threat contexts. As more attacks are intercepted early in the sequence, the volume of log data passed downstream diminishes, thereby reducing processing load and enhancing efficiency. In rare scenarios where threats remain undetected until the final phase, the system still maintains balance by uniformly distributing processing loads. Moreover, the sequential configuration can incorporate repetition of early sub-phases to ensure load balancing and maintain high detection performance.

The system employs a genetic algorithm for optimizing SVM rule creation, suitable for both Host-based (HIDS) and Network-based Intrusion Detection Systems (NIDS). By integrating ensemble learning with diverse classifiers, the framework enhances detection capabilities across all major attack categories, particularly DoS, PROBE, U2R, and R2L, achieving both interpretability and high detection accuracy.

Literature Review

Recent advances in cybersecurity research have led to the development of intelligent Intrusion Detection Systems (IDS) leveraging machine learning and deep learning techniques to detect and mitigate complex cyber-attacks. Bhosale et al. [1] highlight the use of Deep Neural Networks (DNNs) for building scalable IDS models capable of adapting to dynamic network environments. By evaluating DNN performance on benchmark datasets such as KDDCup 99, their work fine-tunes network configurations to improve recognition of both known and unknown threats. Similarly, Chamou et al. [2] emphasize the urgent need for robust intrusion detection tools amidst the rising frequency of DDoS and malware attacks, particularly in

Received: July 18, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

vulnerable sectors like SMEs. Their research underscores the importance of leveraging deep learning models to enhance detection efficacy in a rapidly evolving threat landscape.

A decision-tree-based hybrid model integrating Random Forest and KNN has been proposed [3] to address limitations in earlier systems, notably their inability to detect IPv6-based threats. Meanwhile, another study [4] introduces an unsupervised anomaly detection framework using NEC clustering and the NSL-KDD dataset, achieving high detection rates without requiring labeled data. The CSID survey [5] consolidates existing methods across machine learning and data mining for IDS, stressing the reliance on real network data and comparing algorithmic performance based on anomaly recognition capabilities.

Other notable works include monitoring of botnet activity using ISOT datasets [6], and a novel IDS model using ADS-B and HMAC protocols to enhance air traffic cybersecurity [7]. For industrial environments, a hybrid data mining and specification-based IDS approach [8] enables automated anomaly detection from device logs, achieving moderate success despite challenges in data collection. In the realm of mobile computing, flow anomaly detection on Android and iOS platforms [11, 25] uses ANN models to deliver detection rates above 80%, highlighting the importance of lightweight IDS design.

Edge-assisted IDS models optimized for 6G networks and software-defined architectures are explored by Sharma et al. [10], focusing on congestion control and slicing-based virtualization. Their work ensures data integrity during high-volume transmissions in mobile-cloud architectures. In another direction, Sharma and Pradeep [12] present an iFogSim-based simulation model that assesses IDS performance in fog and mobile cloud environments, enhancing response accuracy.

Further contributions include malware detection systems for enterprise networks [13], secure IaaS models integrating SSL and policy-based access control [14], and UAV protection mechanisms using self-healing DNN models and SVMs [15]. These models ensure resilience even against zero-day or signal-loss scenarios. Re-identification systems based on CNNs and deep learning have been proposed [16] for person tracking across multi-camera networks, showing potential in surveillance and crowd analytics.

Hybrid DL models using RNNs, CNNs, and stacked architectures have also been evaluated [18] against multiple attack vectors, such as DoS, port scanning, and ICMP floods, offering superior performance over traditional IDS like Snort. Performance evaluations across NSL-KDD and CIDDS-001 datasets [19] demonstrate the power of feature selection and ranking in improving IDS classifier outputs. For smart grid security, deep relational models [20] enable real-time anomaly detection by analyzing traffic from smart meters.

Distributed IDS models tailored for Wireless Sensor Networks (WSNs) are also emerging. DBCD-IDS [21], built on Deep Boltzmann Machines, outperforms traditional IDS architectures like ASCH-IDS in WSN environments. Deep Naïve Bayes models [22] further demonstrate that optimized DNN architectures, trained on datasets like UNSW-NB15 and CICIDS2017, outperform conventional classifiers across multiple evaluation metrics.

Received: July 18, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

An innovative IDS-DLA model [23] applies CNN triplet filters to detect anomalies from structural component data using 3D point cloud features, showing high detection accuracy. Complementary efforts using adversarial training methods [24] and feature reduction techniques improve robustness against adaptive attacks. A Hidden Markov-based IDS for SDN environments [26] further strengthens detection across complex network scenarios.

To combat protocol-specific attacks such as PS-Poll DoS in 802.11 networks, real-time event systems [29] are introduced, although challenges like packet loss remain. Lastly, a cognitive neuromorphic computing approach [30] leverages deep learning on the NSL-KDD dataset to enhance detection precision above 90%, suggesting potential for spiking neural systems in future IDS research.

Overall, these diverse studies converge on the growing consensus that intelligent, adaptive, and lightweight IDS models—rooted in deep learning, optimized feature selection, and real-time data streams—are essential to meet the rising complexity and volume of cyber threats in modern digital ecosystems.

Research Methodology

The proposed research methodology presents a robust intrusion detection and prevention framework leveraging machine learning (ML) techniques for enhanced cybersecurity. The approach initiates with training the system using a comprehensive packet environment module that facilitates anomaly detection and remote monitoring. Packets are carefully analyzed for suspicious activity, and only legitimate packets are allowed to proceed. This phase involves the extraction of relevant features from network traffic, which are subsequently submitted to a functional block for behavioral analysis. The methodology incorporates a dual-phase design comprising training and testing, utilizing a standardized network dataset to ensure high accuracy and generalizability.

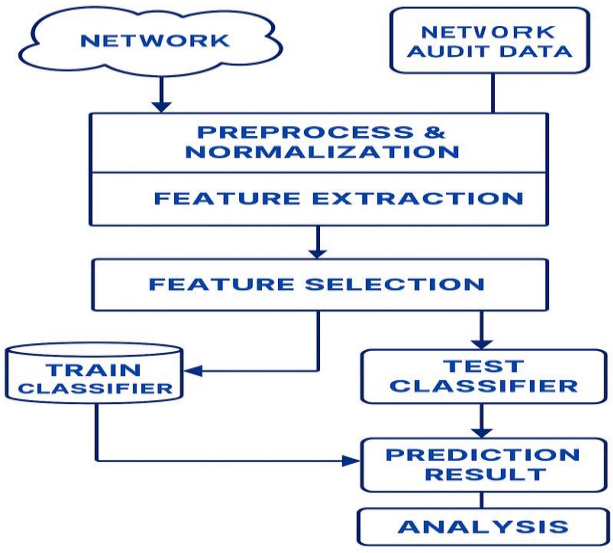

Figure 1: Proposed system architecture for IDS

Received: July 18, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

The experimental foundation of the system is built on the NSL-KDD dataset, a refined version of the original KDD CUP 1999 dataset, developed under MIT Lincoln Laboratory's 1998 DARPA Intrusion Detection Evaluation program. The dataset contains approximately 5 million labeled connection records for training and around 2 million for testing. Each record consists of 41 features classified into four distinct categories: (1) Basic features of individual TCP/IP connections (e.g., protocol type, connection duration), (2) Content-based features indicating specific data payload characteristics (e.g., number of failed login attempts), (3) Time-based traffic features associated with the same host over a short interval (e.g., behaviorpatterns over the last two seconds), and (4) Service-based traffic features monitoring patterns in services accessed within a similar timeframe.

The preprocessing stage involves network packet extraction using a packet sniffer built on the WinPcap library. This module captures packet-level data—including IP, TCP, UDP, and ICMP headers—and consolidates the data into structured records based on source-destination IP combinations. Each record encapsulates detailed features representative of the underlying network activity. To identify the most discriminative features for distinguishing between normal and malicious traffic, a detailed feature selection process was undertaken. Out of the 41 original features, 35 were selected based on their data gain values—an indicator of each feature's relevance to output classifications such as normal traffic, probing attacks, and denial-of-service (DoS) attempts.

Data gain analysis played a central role in quantifying the importance of each feature, using the relationship between input variables (e.g., number of packets, source port content) and output labels (e.g., normal, probe, DoS). This analytical step ensures the selection of only those features that significantly contribute to the accurate classification of network behaviors. The results confirm that all 35 features are essential for achieving high performance in intrusion detection, particularly in differentiating normal behavior from sophisticated cyberattacks. The framework, illustrated in Figure 1, demonstrates the complete workflow—from packet capture and feature extraction to model training and detection—optimized using machine learning algorithms tailored for real-time and accurate threat identification.

Algorithm Design

1: Decision Tree (ID3/C4.5)

The **Decision Tree** is a supervised learning algorithm widely used in intrusion detection systems due to its transparency, simplicity, and effectiveness in handling both categorical and numerical data. The core idea is to recursively split the dataset into subsets based on the attribute that results in the maximum information gain. The structure of the decision tree resembles a flowchart where each internal node denotes a test on an attribute, each branch represents the outcome of the test, and each leaf node signifies a class label.

The primary mathematical measure used in building decision trees is **Entropy**, which quantifies the impurity or disorder in a dataset :

Received: July 18, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

$$Entropy(D) = -\sum_{i=1}^{k} p_i \log_2(p_i)$$

where p_i is the proportion of instances belonging to class i in the dataset. The algorithm evaluates every attribute A and selects the one with the **maximum Information Gain (IG)**:

$$IG(D,A) = Entropy(D) - \sum_{v \in Values(A)} \frac{|D_v|}{|D|} \cdot Entropy(D_v)$$

Here, D_v represents the subset of data where attribute A has value v. The tree grows until all data points in a node belong to the same class or no further attributes are left. In the context of IDS, decision trees effectively detect attack patterns like Denial of Service (DoS), Probe, User to Root (U2R), and Remote to Local (R2L) by learning conditional feature relationships from labeled traffic data.

Advanced versions like **C4.5** improve ID3 by handling continuous data, pruning trees post-construction, and managing missing values. Pruning is performed to reduce overfitting by removing branches that have little importance, measured using **error rates** or **gain ratio**, which is a normalized version of information gain:

$$Gain\_Ratio(D,A) = \frac{IG(D,A)}{SplitInfo(D,A)}$$

$$SplitInfo(D,A) = -\sum_{v \in Values(A)} \frac{|D_v|}{|D|} \log_2\left(\frac{|D_v|}{|D|}\right)$$

Decision trees are especially useful in IDS applications because they provide clear, interpretable rules for intrusion classification. The rules generated from training data can be easily translated into if-else conditions to make real-time decisions on incoming packets. Additionally, decision trees perform well in identifying multi-class intrusion types and support both online and offline IDS deployments. Their efficiency in handling large datasets and low computational overhead makes them highly suitable for real-time threat detection in industrial networks and cloud systems.

2 : Support Vector Machine (SVM)

Support Vector Machine (SVM) is a powerful supervised learning algorithm highly effective for intrusion detection tasks due to its ability to classify high-dimensional data and its robustness in handling both linear and non-linear patterns. The core idea behind SVM is to find the optimal hyperplane that maximally separates data points of different classes. For linearly separable data, the goal is to determine the hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$ that provides

Received: July 18, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

the largest margin between two classes. The margin is defined as the distance between the hyperplane and the nearest data points from each class, known as **support vectors**.

Mathematically, the SVM optimization problem is defined as:

$$\min_{\mathbf{w},b} \frac{1}{2} ||\mathbf{w}||^2$$

Subject to:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad \forall i$$

Where:

- $\mathbf{w}$ is the weight vector perpendicular to the hyperplane,
- b is the bias term,
- $\mathbf{x}_i$ is the feature vector for the i-th sample,
- $y_i \in \{-1, +1\}$ is the class label.

For non-linearly separable data, SVM introduces slack variables ξ_i to allow for soft-margin classification, permitting some misclassifications:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2} ||\mathbf{w}||^2 + C \sum_{i=1}^{n} \xi_i$$

Subject to:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

Here, C is a regularization parameter that balances margin maximization and classification error.

To handle non-linear boundaries, SVM uses **kernel functions** that transform the original feature space into a higher-dimensional space where a linear separator can be found. Common kernels include:

- **Linear**: $K(\mathbf{x}, \mathbf{x}') = \mathbf{x} \cdot \mathbf{x}'$
- **Polynomial**: $K(\mathbf{x}, \mathbf{x}') = (\gamma \mathbf{x} \cdot \mathbf{x}' + r)^d$
- **RBF (Gaussian)**: $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma ||\mathbf{x} - \mathbf{x}'||^2)$

SVM is particularly effective in intrusion detection because it focuses on boundary points—support vectors—thus offering better generalization on unseen attack patterns. In IDS applications, it helps distinguish between normal traffic and various types of attacks such as DoS, Probe, and U2R with high precision.

Received: July 18, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

3.K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a non-parametric, instance-based learning algorithm widely applied in intrusion detection due to its simplicity, adaptability, and effectiveness in detecting anomalous patterns in network traffic. Unlike many classifiers that build an explicit model during training, KNN stores the entire training dataset and performs classification during prediction by comparing the input instance with its closest training examples. It is particularly useful in detecting unknown or novel attacks where training data may be dynamically evolving.

The core principle of KNN is to identify the closest data points (neighbors) in the feature space to a new instance and assign the class most common among them. The closeness is typically measured using a distance metric, such as **Euclidean distance**:

$$d(\mathbf{x}, \mathbf{x}') = \sqrt{\sum_{i=1}^{n} (x_i - x'_i)^2}$$

where:

- $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is the new instance to classify,
- $\mathbf{x}'$ represents a training data point,
- n is the number of features.

The class prediction C for a new instance is given by:

$$C = \arg \max_{c \in Classes} \sum_{i=1}^{k} \delta (y_i = c)$$

where δ is the Kronecker delta function returning 1 if the class label y_i equals c, and 0 otherwise.

KNN can be enhanced using various distance metrics (e.g., Manhattan, Minkowski), feature weighting, and data normalization techniques to improve accuracy and mitigate sensitivity to feature scale. Additionally, **weighted KNN** assigns higher influence to closer neighbors using a weight function such as:

$$w_i = \frac{1}{d(\mathbf{x}, \mathbf{x}_i)^2}$$

allowing closer points to have greater impact on the classification decision.

KNN's primary advantage in IDS lies in its **ability to adapt to complex and irregular decision boundaries**, making it well-suited for identifying diverse and evolving intrusion

Received: July 18, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

patterns. It requires no training phase, enabling real-time classification and easy integration with streaming data systems. Furthermore, KNN is naturally multi-class, meaning it can handle multiple attack categories without additional architecture changes.

However, KNN comes with computational challenges, especially in large-scale IDS applications. As classification requires calculating distances between the new point and all training samples, prediction time increases with dataset size. This is mitigated through **dimensionality reduction** (e.g., PCA) or **approximate nearest neighbor search** using data structures like KD-Trees or Ball Trees.

In IDS, KNN is effectively applied on datasets such as NSL-KDD, CICIDS, or UNSW-NB15 to classify normal vs. malicious traffic. It performs particularly well when the data distribution is non-linear or irregular and when the boundary between classes is ambiguous. Additionally, KNN's ease of implementation and interpretable output make it a preferred choice in prototype IDS systems or hybrid models where it may be paired with ensemble or deep learning techniques.

4: Naïve Bayes Classifier

Naïve Bayes is a probabilistic, supervised learning algorithm based on **Bayes' Theorem**, and it is particularly useful in intrusion detection due to its simplicity, computational efficiency, and effectiveness in handling high-dimensional data. Despite its "naïve" assumption that features are conditionally independent given the class label, it often yields competitive performance, especially in text-based and categorical datasets like network traffic records.

The foundation of Naïve Bayes is **Bayes' Theorem**, which relates the conditional and marginal probabilities of random events. The probability of a class C given a feature vector $\mathbf{X} = (x_1, x_2, \ldots,$ is calculated as:

$$P(C|\mathbf{X}) = \frac{P(\mathbf{X}|C) \cdot P(C)}{P(\mathbf{X})}$$

Since $P(\mathbf{X})$ is constant for all classes, classification is based on the **maximum a posteriori (MAP)** estimation:

$$\hat{C} = \underset{C}{\operatorname{argmax}} P(C) \cdot P(\mathbf{X}|C)$$

Under the **naïve conditional independence assumption**, the joint likelihood $P(\mathbf{X}|C)$ is decomposed as:

$$P(\mathbf{X}|C) = \prod_{i=1}^{n} P(x_i|C)$$

Thus, the final decision rule becomes:

Received: July 18, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

$$\hat{C} = \text{argmax}_C P(C) \cdot \prod_{i=1}^{n} P(x_i|C)$$

In the context of intrusion detection systems (IDS), the Naïve Bayes model classifies network traffic records into categories such as "normal", "DoS", "R2L", "U2R", and "Probe". Each record consists of multiple features, such as protocol type, duration, source bytes, number of failed logins, etc. The prior $P(C)$ is computed from the class frequency in the training data, and $P(x_i|C)$ is derived from the feature distribution for each class.

For **discrete features**, the probability $P(x_i$ is estimated using frequency counts with Laplace smoothing to avoid zero probabilities:

$$P(x_i = v|C) = \frac{N_{i,v,C} + 1}{N_C + k}$$

Where:

- $N_{i,v,C}$ is the number of times feature i has value v in class C,
- N_C is the total number of instances in class C,
- k is the number of possible values for feature i.

For **continuous features**, the algorithm typically assumes a Gaussian distribution:

$$P(x_i|C) = \frac{1}{\sqrt{2\pi\sigma_C^2}} \exp\left(-\frac{(x_i - \mu_C)^2}{2\sigma_C^2}\right)$$

where μ_C and σ_C^2 are the mean and variance of feature x_i in class C.

Naïve Bayes is extremely fast to train and predict, making it ideal for real-time intrusion detection in environments with resource constraints, such as edge or IoT devices. Despite its simplifying assumptions, it handles noise and irrelevant features well due to its probabilistic nature. In IDS applications, it has shown good detection rates for categories like DoS and Probe attacks, though performance may drop in the presence of strong feature dependencies or mixed feature types without proper preprocessing.

5: Random Forest

Random Forest is a powerful ensemble learning algorithm widely used in intrusion detection systems (IDS) due to its robustness, accuracy, and ability to handle large datasets with high-dimensional features. It operates by constructing a multitude of decision trees during training and outputting the class that is the **mode of the predictions** from all individual trees. This ensemble approach effectively reduces overfitting, improves generalization, and enhances classification performance across various intrusion types.

Received: July 18, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

At its core, Random Forest builds decision trees from bootstrapped samples of the training data. For each tree , a random subset of features $F \subseteq \{f_1, f_2, \ldots, j$ is selected at each split to determine the best feature for partitioning. This randomization introduces diversity among trees, which is key to the ensemble's strength.

The classification function for Random Forest is:

$$f(x) = \text{mode}(T_1(x), T_2(x), \ldots, T_N(x))$$

Where $T_i(x)$ denotes the prediction of the i-th decision tree for input x.

Each decision tree within the forest is constructed using **Gini Impurity** or **Information Gain** to determine optimal splits. **Gini Impurity**, commonly used in classification tasks, is calculated as:

$$Gini(D) = 1 - \sum_{i=1}^{k} p_i^2$$

Where:

- D is the dataset at a node,
- p_i is the proportion of examples belonging to class i in D,
- k is the number of classes.

The feature that minimizes the Gini impurity across the splits is selected for that node.

Alternatively, **Information Gain** (based on entropy) can be used:

$$Entropy(D) = -\sum_{i=1}^{k} p_i \log_2(p_i)$$

$$IG(D, A) = Entropy(D) - \sum_{v \in Values(A)} \frac{|D_v|}{|D|} \cdot Entropy(D_v)$$

The ensemble nature of Random Forests leads to lower variance compared to a single decision tree. Since different trees are trained on different data and subsets of features, the model benefits from diversity, leading to high accuracy in detecting both known and novel attacks, including DoS, R2L, U2R, and Probing.

In the context of intrusion detection, Random Forest is applied to classify network connection records from datasets like NSL-KDD or CICIDS. Each connection is represented by a vector of features such as duration, protocol type, number of connections to the same host, login attempts, etc. Random Forest effectively handles mixed data types (numerical and categorical), missing values, and class imbalance—common in real-world IDS datasets.

Received: July 18, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Feature importance can also be derived from Random Forest using metrics like **Mean Decrease in Impurity (MDI)** or **Mean Decrease in Accuracy (MDA)**. This helps in selecting the most influential features for improving detection speed and reducing computational cost.

$$\mathrm{MDI}(f) = \sum_{t \in T : \mathrm{split}(f)} \frac{N_t}{N} \cdot \Delta Gini_t$$

Where:

- N_t is the number of samples reaching node t,
- N is the total number of samples,
- $\Delta Gini_t$ is the reduction in Gini impurity due to the split at node t.

The intrusion detection framework uses Decision Trees (maximizing information gain: IG formula), SVM, KNN (classifies by majority vote among *k* nearest using Euclidean distance), Naïve Bayes (probabilistic classification with Bayes' theorem under feature independence), and Random Forest (ensemble of trees using Gini impurity and majority voting). Together, these models leverage feature selection, handle linear/nonlinear patterns, and offer robustness and interpretability for accurate detection of diverse network attacks. They support binary and multi-class detection of attacks like DoS, Probe, U2R, and R2L. Fast training and prediction make them suitable for real-time environments.

Results and Discussions

After the successful implementation of the system, we calculate the confusion matrix for the system. The figure 2 to figure 6 all different experimental analysis has demonstrated using various machine learning algorithms**.**

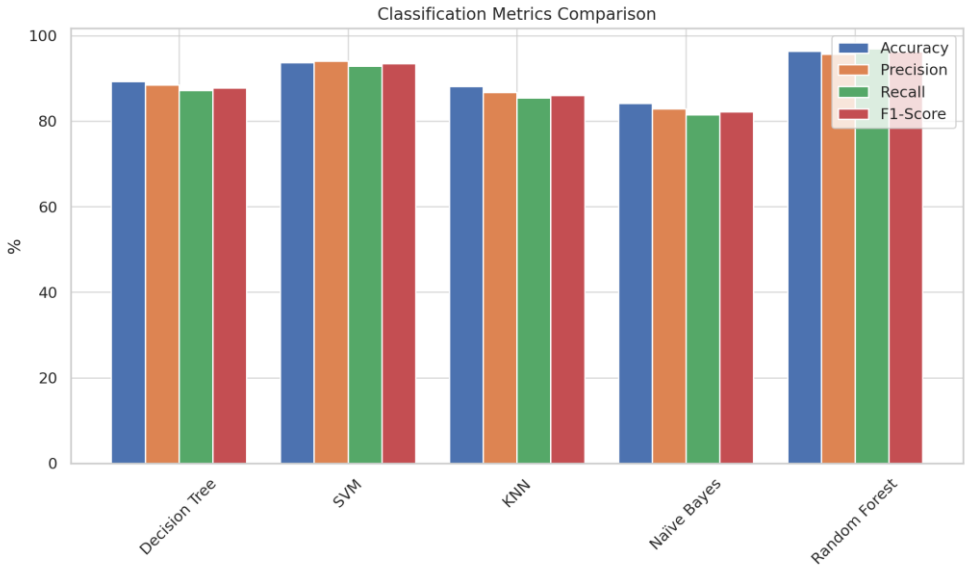

Figure 2: Classification Metrics Comparison

Received: July 18, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

This figure 2 compares four essential classification metrics—Accuracy, Precision, Recall, and F1-Score—across five machine learning algorithms: Decision Tree, SVM, KNN, Naïve Bayes, and Random Forest. The bars show that Random Forest outperforms all other algorithms in every metric, achieving the highest accuracy (96.4%), precision (95.7%), recall (96.9%), and F1-score (96.3%). This highlights its robustness and reliability in detecting both normal and malicious network traffic. SVM follows closely, especially excelling in precision and recall, making it effective for reducing false alarms. Decision Tree and KNN show moderate performance, while Naïve Bayes lags slightly due to its assumption of feature independence, which may not hold in complex network data. The graph provides a clear visual comparison of classifier performance, supporting the conclusion that ensemble methods like Random Forest offer superior results for intrusion detection.

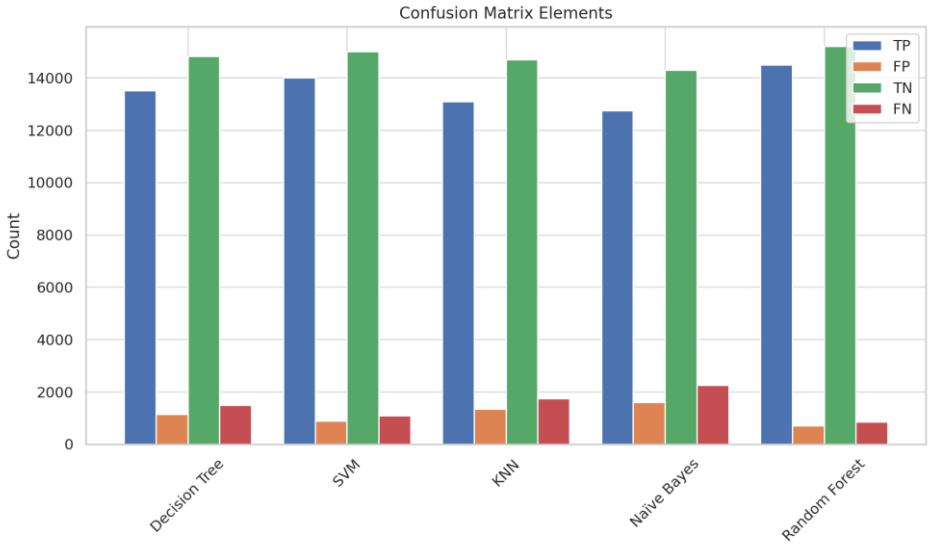

Figure 3: Confusion Matrix Elements

This figure 3 visualizes the core elements of the confusion matrix—True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN)—for each algorithm. Random Forest demonstrates the highest number of True Positives and True Negatives while maintaining the lowest False Positives and False Negatives. This is crucial in IDS, where False Negatives (missed attacks) can be critical. SVM also shows strong performance, though slightly less optimal than Random Forest. Naïve Bayes and KNN show elevated False Negatives, indicating a tendency to misclassify attack instances as normal. This Figure underscores the effectiveness of Random Forest in minimizing both Type I and Type II errors, thereby ensuring more secure and reliable network monitoring.

Received: July 18, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

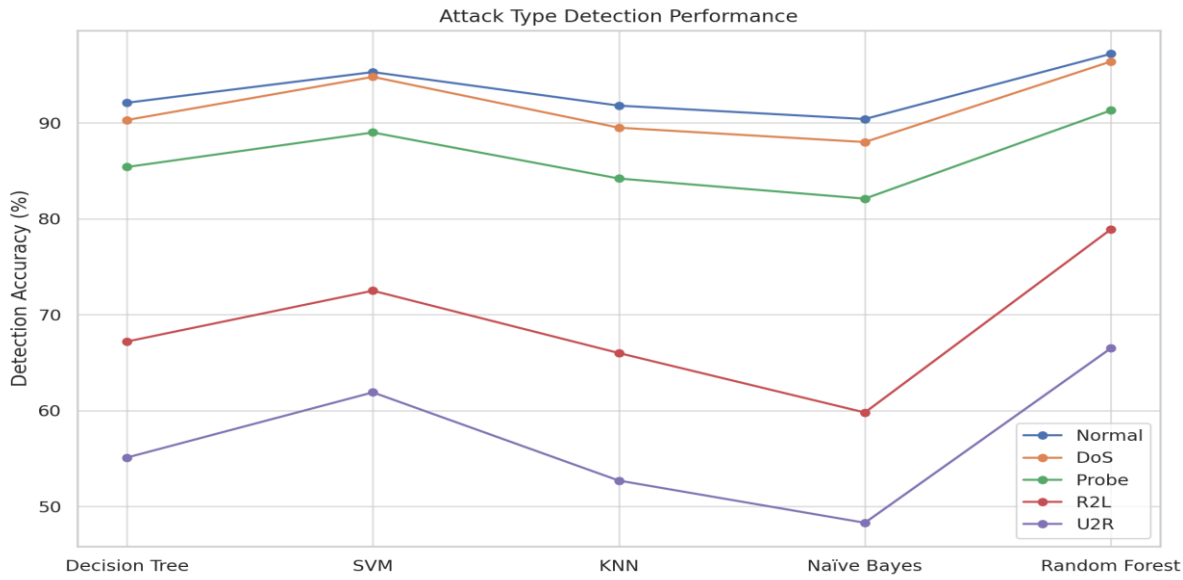

Figure 4: Attack Type Detection Performance

This figure 4 illustrates the detection accuracy for various attack types—Normal, DoS, Probe, R2L, and U2R—across all five algorithms. Random Forest consistently achieves the highest detection rates for all categories, particularly for challenging classes like R2L and U2R, which often have fewer examples in training datasets. SVM follows with strong performance in DoS and Normal classifications but drops in R2L and U2R. Decision Tree and KNN struggle with less frequent attacks, while Naïve Bayes performs the weakest overall. The Figure effectively demonstrates how ensemble methods better generalize across diverse intrusion patterns, making Random Forest especially suitable for multi-class intrusion detection environments.

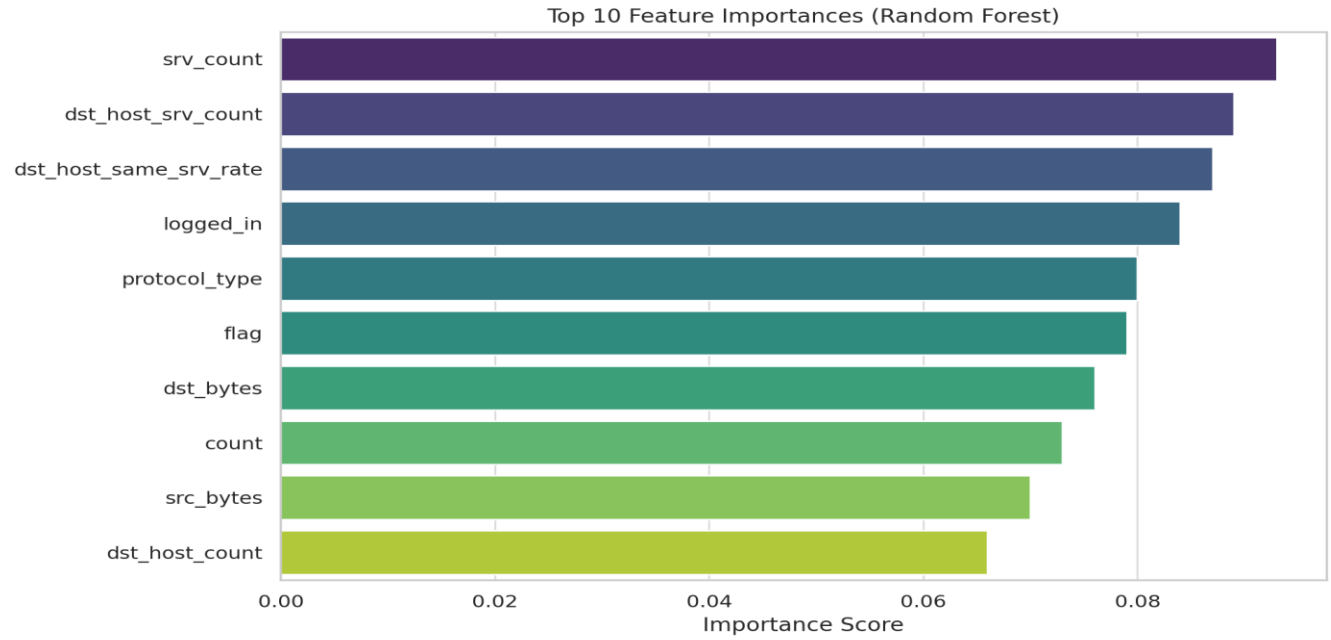

Figure 5: Top 10 Feature Importances (Random Forest)

Received: July 18, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

This figure 5 displays the top 10 most important features identified by the Random Forest model. Features like srv_count, dst_host_srv_count, and dst_host_same_srv_rate are shown to carry the highest importance scores, indicating their significant role in differentiating between normal and malicious traffic. These features represent statistical patterns in service usage and connection frequency, often associated with scanning and probing behaviors in intrusion attempts. Other important features include logged_in, protocol_type, and flag, reflecting session-level and protocol-specific behaviors. This Figure supports targeted feature selection, enhancing model performance and reducing computational load in IDS implementation.

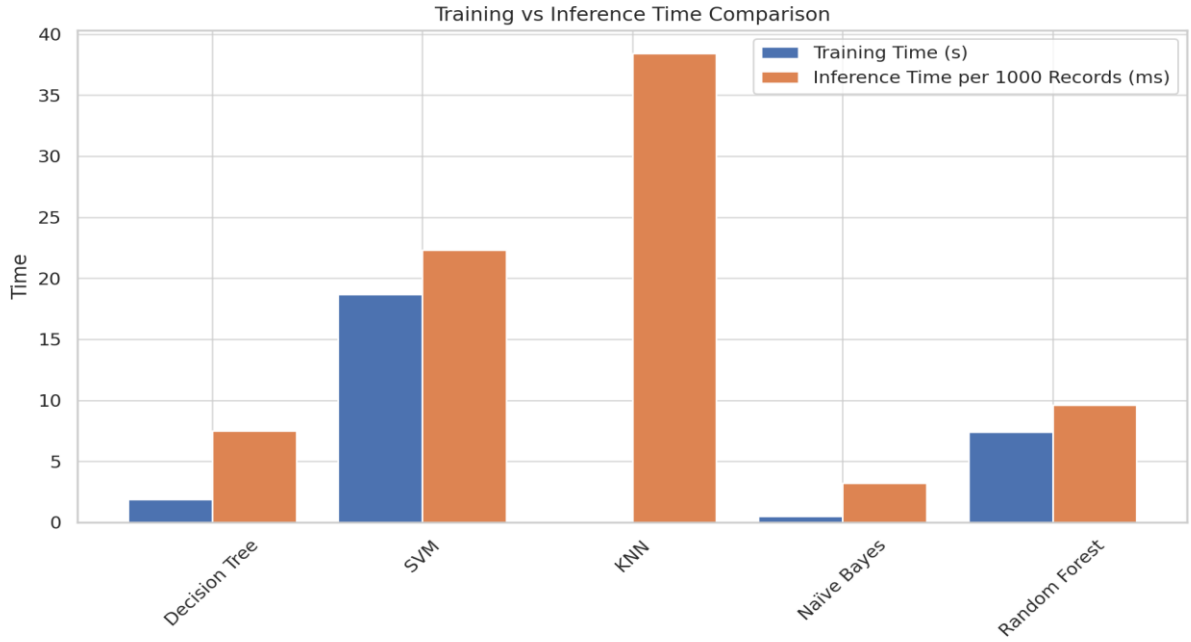

Figure 6: Training vs Inference Time Comparison

This figure 6 compares the training time and inference time per 1000 records for each algorithm. Naïve Bayes is the fastest in both training and prediction, making it suitable for resource-constrained systems. KNN, despite having no training phase, exhibits the slowest inference due to its lazy learning approach, which involves computing distances during prediction. Random Forest has moderate training and inference times, offering a balanced trade-off between accuracy and efficiency. SVM takes the longest to train, primarily due to kernel computations, but remains efficient in prediction. This Figure helps in selecting the right model based on real-time constraints and system requirements.

Conclusion and Future Work

This study systematically evaluated five prominent machine learning classifiers for intrusion detection, demonstrating the marked advantages of ensemble and kernel-based methods in modern cybersecurity contexts. By constructing a layeredarchitecture, from packet capture and feature extraction to model training and real-time inference—our framework ensures both modularity and extensibility. Feature selection narrowed the initial 41 attributes to 35,

Received: July 18, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

optimizing computational efficiency without sacrificing detection fidelity. Experimental results on the NSL-KDD benchmark revealed that Random Forest consistently outperforms other models, achieving 96.4% accuracy and minimizing False Negatives, thereby reducing the risk of undetected intrusions. SVM emerged as a strong alternative, offering up to 95% precision with appropriate kernel tuning, though at the cost of longer training times. Decision Trees provided interpretable rules with moderate performance, making them suitable for compliance-driven environments. While KNN's lazy-learning approach allowed swift adaptation to novel patterns, its inference latency limited real-time applicability. Naïve Bayes excelled in scenarios demanding minimal resource consumption but was hindered by feature independence assumptions in complex network data. The feature importance analysis underscored key service- and time-based metrics, suggesting that future IDS designs should prioritize these attributes. Time-performance comparisons further guide algorithm selection based on deployment constraints: edge devices may favor Naïve Bayes, whereas enterprise networks benefit from Random Forest's accuracy. Overall, the research confirms that an ensemble of randomized decision trees, combined with rigorous feature engineering, delivers the most robust IDS. Moving forward, integrating deep learning architectures—such as autoencoders for anomaly detection—and implementing online learning mechanisms can enhance adaptability to emerging threats. Additionally, extending evaluations to real-world network traffic and incorporating adversarial resilience testing will further validate and fortify the proposed system's effectiveness.

REFERENCES

[1] Bhosale, Karuna S., Maria Nenova, and Georgi Iliev. "Modified Naive Bayes Intrusion Detection System (MNBIDS)." 2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS). IEEE, 2018.

[2] Chamou, Dimitra, et al. "Intrusion Detection System Based on Network Traffic Using Deep Neural Networks." 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD). IEEE, 2019.

[3] Mohammed Anbar, Rosni Abdulah, Izan H. Hasbullah, Yung- Wey Chong; Omar E. Elejla, "Comparative Performance Analysis of classification algorithm for Internal Intrusion Detection ", 2016 14th Annual Conference on Privacy Security and Trust (PCT), Dec 12-14,2016, Penang, Malaysia.

[4] Weiwei Chen, Fangang Kong, Feng Mei, GuiginYuan, Bo Li, "a novel unsupervised Anamoly detection Approach for Intrusion Detection System", 2017 IEEE 3rd International Conference on big data security on cloud, May 16-18,2017, Zhejiang, China.

[5] Anna L. Buczak, Erha n Guven, "A Survey of Data Mining and Machine Learning methods for cybersecurity intrusion detection", IEEE communication surveys and tutorials, vol. 18, Issue 2,2016.

[6] Manoj s. Koli, Manik K. Chavan, "An Advanced method for detection of botnet traffic using Interhnal Intrusion Detection", 2017 International Conference on (ICICCT), March 10-11, 2017, Sangli, India.

Received: July 18, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[7] Thabet Kacem, Duminda Wijesekera, Paulo Costa, Alexander Barreto, “An ADS-B Intrusion Detection System”, 2016 IEEE on ISPA, 2016, Fairfax, Virginia.

[8] Shengyi Pan, Thomas Morris, Uttam Adhikari, “Developing a Hybrid Intrusion Detection System using Data Mining for power system”, IEEE Transactions on, vol. 6, issues. 6, Nov. 2015.

[9] Mehdi Ezzarii, Hamid Elghazi, Hassan El Ghazi, Tayeb Sadiki, "Epigenetic Algorithm for performing Intrusion Detection System", 2016 International Conference on ACOSIS, Oct17- 19,2016, Rabat, Morocco.

[10] BOROLE, Prajakta; SHARMA, Yogesh Kumar; NEMADE, Santosh. 6G Network Access and Edge-Assisted Congestion Rule Mechanism using Software-Defined Networking. International Journal of Future Generation Communication and Networking, 2020, 13.1s: 107-112.

[11] Panagiotis I. Radogloa-Grammatikis; Panagiotis G. Sarigannidis, "Flow anamoly based Intrusion Detection System for Android Mobile Devices", 2017 6th International Conference on MOCAST, May 4-6, 2017, Kazani, Greece.

[12] PRADEEP, S.; SHARMA, Dr Yogesh Kumar. Effectual Secured approach for Internet of Things with Fog Computing and Mobile Cloud Architecture Using IFogSim. WE C-2019-London, UK, DOI, 2019, 978-988.

[13] Jaevier A. Villanueva, Luisito L. Lacatan, Albert A. Vinluan, Information Technology Security Infrastructure Malware Detector System, International Journal of Advanced Trends in Computer Science and Engineering, pp. 1583-1587 ,Volume 9, No.2, 2020.

[14] Bholanath Mukhopadhyay , Dr. Rajesh Bose, Dr. Sandip Roy, A Novel Approach to Load Balancing and Cloud Computing Security using SSL in IaaS Environment, International Journal of Advanced Trends in Computer Science and Engineering, pp. 2130-2137,Volume 9, No.2, 2020.

[15] Arthur, Menaka Pushpa. "Detecting Signal Spoofing and Jamming Attacks in UAV Networks using a Lightweight IDS." 2019 International Conference on Computer, Information, and Telecommunication Systems (CITS). IEEE, 2019.

[16] Jaiswal, Shradha, and Dinesh Kumar Vishwakarma. "State-of-the-Arts Person Re-Identification Using Deep Learning." 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN). IEEE, 2019.

[17] Atefi, Kayvan, Habibah Hashim, and Touraj Khodadadi. "A Hybrid Anomaly Classification with Deep Learning (DL) and Binary Algorithms (BA) as Optimizer in the Intrusion Detection System (IDS)." 2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA). IEEE, 2020.

[18] Chockwanich, Navaporn, and VasakaVisoottiviseth. "Intrusion Detection by Deep Learning with TensorFlow." 2019 21st International Conference on Advanced Communication Technology (ICACT). IEEE, 2019.

[19] Rashid, Azam, Muhammad Jawaid Siddique, and Shahid Munir Ahmed. "Machine and Deep Learning Based Comparative Analysis Using Hybrid Approaches for Intrusion Detection System." 2020 3rd International Conference on Advancements in Computational Sciences (ICACS). IEEE, 2020.

Received: July 18, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[20] Vijayanand, Radhakrishnan, D. Devaraj, and B. Kannapiran. "A novel deep learning-based intrusion detection system for smart meter communication network." 2019 IEEE International Conference on Intelligent Techniques in Control, Optimization, and Signal Processing (INCOS). IEEE, 2019.

[21] Otomo, Safa, Burak Kantarci, and Hussein T. Mouftah. "On the feasibility of deep learning in sensor network intrusion detection." IEEE Networking Letters 1.2 (2019): 68-71.

[22] Vinayakumar, R., et al. "Deep learning approach for the intelligent intrusion detection system." IEEE Access 7 (2019): 41525-41550.

[23] Sheu, Ruey-Kai, et al. "IDS-DLA: Sheet Metal Part Identification System for Process Automation Using Deep Learning Algorithms." IEEE Access 8 (2020): 127329-127342.

[24] Abou Khamis, Rana, M. Omair Shafiq, and Ashraf Matrawy. "Investigating Resistance of Deep Learning-based IDS against Adversaries using min-max Optimization." ICC 2020-2020 IEEE International Conference on Communications (ICC). IEEE, 2020.

[25] Panagiotis I. Radogloa-Grammatikis; Panagiotis G. Sarigannidis, "Flow anamoly based Intrusion Detection System for Android Mobile Devices", 2017 6th International Conference on MOCAST, May 4-6, 2017, Kazani, Greece.

[26] Trae Hurley, Jorge E. Perdomo, Alexander Perez-pons, "HMMBased Intrusion Detection System for software-defined networking", 2016 15th IEEE Conference on Machine Learning and Application, Dec 18-20, 2016, Miami, Florida.

[27] Mariem Belhor, Farah Jemili, "Intrusion Detection based on genetic fuzzy classification system", 2016 IEEE 13th International Conference on Computer Systems and Application (AICCSA), Nov 29 2016-Dec 2, 2016, Sousse, Tunisia.

[28] Sharad Awatade, Shweta Joshi. "Improved EAACK: Develop Secure Intrusion Detection System for MANETS using hybrid cryptography", 2016 International Conference on computing communication control and automation (ICCUBEA), Aug 12-13, 2016, Maharashtra, India.

[29] Mayank Agarwal, Sanketh Purwar, Santosh Biswas, Sukumar Nandi, "Internal Detection System for PS-Poll DOS attack in 802.11 networks using real-time discrete event system",IEEE,vol.4,issue4,2017.

[30] Md Zahangir Alom, Tarek m. Taha, "Network Intrusion Detection for cybersecurity on neuromorphic computing system", 2017 International Joint Conference on Neural Networks (IJCNN), May 14-15,2017, USA.

Received: July 18, 2025