International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

NetGuardAI: A Real-Time Hybrid Learning Framework for Intelligent Threat Detection in Modern Network Environmentsem

Kunal Chauhan
Amity School of Engineering & Technology
Amity University, Noida
Noida, India
kunalchauhanamity@gmail.com

Madhav Ahuja
Amity School of Engineering & Technology
Amity University, Noida
Noida, India
madhavahujaamity@gmail.com

Utsav Das
Amity School of Engineering & Technology
Amity University, Noida
Noida, India
utsav8103@gmail.com

Nirbhay Kashyap
Amity School of Engineering & Technology
Amity University, Noida
Noida, India
nkashyap@amity.edu

***Abstract*— The existing intrusion detection systems (IDSs) that have been built to monitor and react to cyber threats are becoming increasingly difficult to use due to their limitations in the detection of new or evolving attack types. The following paper describes the NetGuard AI solution, a real-time IDS/P system that utilizes machine learning to improve security across the network. The training data used to build our Random Forest classification model comes from the CICIDS-2018 dataset, which was selected using an optimized feature selection process, as well as oversampling techniques (SMOTE) to achieve 98% accuracy for all high-frequency attack types including DoS, DDoS, and port scanning. Our solution uses the trained model within the FastAPI framework for quick and efficient real-time threat classification through WebSocket communication, while providing us with the ability to monitor and report on our solution's performance through a React dashboard. Furthermore, the NetGuard AI solution allows the user to automatically enforce firewall rules and block malicious IP addresses and ports in both the Windows and Linux environments. The dedicated testing mode allows users to evaluate our solution safely without requiring admin privileges and without having to connect to a live network. While the detection performance decreases for some rare attack types, such as SQL Injection and Cross-Site Scripting (XSS), our solution has proven itself to be highly scalable and deployable. Planned future enhancements to our platform include additional deep learning capabilities and cloud-based scaling capabilities that will enhance the detection accuracy and adaptability of our solution.**

Keywords*— *Intrusion Detection System (IDS), Machine Learning, Network Security, Random Forest, CICIDS-2018, SMOTE, Real-time Detection, Firewall Automation, FastAPI, WebSocket, React Dashboard, Cyberattack Classification

I. INTRODUCTION

A. Background and Motivation

Computer networks are becoming one of the most crucial parts of modern day infrastructure, because they are used to provide essential services by the various sectors of society including banking, healthcare, education, and government. However, the same computer networks are now considered to be exposed to a very large number of cyber threats such as cyber-squatting, ransomware, phishing attacks, Distributed Denial of Service (DDoS) attacks, and zero-day exploits, which can significantly disrupt the confidentiality, integrity, and availability of data and systems.

B. Problem Statement

In order to protect against these threats, intrusion detection systems (IDS) are used to constantly monitor and analyze network traffic. Signature-based IDS are effective against pre-existing (known) threats, but are not effective against new or modified attacks. Anomaly-based IDS Supplement signature-based IDS for detecting unknown or modified attacks by detecting abnormal network traffic patterns; however, anomaly-based systems are prone to generate large numbers of false positives, creating an added drain on security teams.

Significant changes have occurred due to machine learning, showing promise in detecting complex patterns of network traffic and intrusion detection. Machine learning techniques such as Random Forests, Support Vector Machines, and Decision Trees have been identified as being very useful for intrusion detection through network interfaces. However, a limitation common to these methods is that they do not provide real-time capability for identifying and responding to threats.

To remove these limitations, this paper proposes a new intrusion detection model called "NetGuardAI," which uses ongoing monitoring of network traffic as it happens. The result of this real-time-based machine learning classification will provide a unified approach to detection that is both fast and accurate.

C. Objectives

- **To develop an intelligent intrusion detection framework using machine learning** that can accurately classify network traffic into benign and malicious categories.
- **To utilize the CICIDS-2018 dataset** due to its realistic network behavior and modern attack coverage, ensuring better representation of contemporary cyber threats.
- **To enhance model performance using optimized feature selection and SMOTE**, addressing data imbalance and improving detection for both high- and low-frequency attacks.
- **To design and implement a real-time monitoring architecture** using FastAPI (backend), React

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

(frontend), and WebSocket communication for low-latency alert delivery.

- **To integrate an automated firewall control mechanism** that dynamically blocks malicious IP addresses or ports upon confirmed detection.
- **To support multi-environment deployment**, enabling both Windows and Linux firewall execution, and facilitating adaptability in enterprise, research, and testbed environments.
- **To incorporate dual operating modes (Test and Live)**, allowing safe evaluation during training/testing phases and full preventive functionality in real-time network deployment.
- **To evaluate system performance using standard classification metrics** (Accuracy, Precision, Recall, F1-score) and further validate statistical reliability through hypothesis-based testing.
- **To compare the proposed model with other ML algorithms and existing IDS approaches**, emphasizing practicality, real-time capability, and prevention readiness.

II. Related Work

Intrusion detection systems can be broadly classified into signature-based and anomaly-based categories. Signature-based approaches can detect previously known threats to organizations. They have a clearly defined and well-documented statistical pattern of attack, but they remain very ineffective for identifying a new or "zero-day" attack [1],[2]. This limitation exists in anomaly-based approaches as well. The emergence of unusual patterns of behavior (for example, violations of defined standards of behavior) can indicate a potential intrusion, but the number of false positives produced by them is so great that they are not useful for most practical applications [2]. As a result, there is now a viable alternative approach to anomaly-based techniques for modeling normal network behavior using machine learning techniques. This has enabled inter-collection data structure DEVELOPMENT of Intrusion Detection Systems that learn to recognize complex patterns of traffic (i.e., deviations from normal traffic behavior) based on large amounts of data. Supervised machine-learning techniques, such as decision trees, support vector machines, and random forests, have demonstrated competitive performance across a number of performance benchmarks on network traffic datasets [3]. Random forests, for example, have been recognized as a robust ensemble learner in that they exhibit a high degree of robustness to changes in feature selection and in network traffic patterns [5]. Clustering techniques have also been used to model normal network behavior; however, the high-volume environment where they may be employed for this function remains a concern [15].

CNN networks are part of a larger class of Deep Learning networks known as Neural Networks and Recurrent Neural Networks (RNNs) and are defined by their extensive detection functions, allowing networks to capture large-scale datasets while measuring changes to their networks; both of these classes of neural networks generally require high levels of processing capacity and are not especially compatible with deploying in real-time environments [4].

Explainability of AI has also gained a great deal of attention within the field of Intrusion Detection Systems (IDS), and has as its goal, making more transparent, the decision making made by an AI model, thus enabling improved explanation for security analysts [11]. In a business setting, being able to explain the reason for an AI model's classification decision, is as important as the classification decision itself.

Despite these advancements, there are still some issues relating to IDS that have not been resolved, or are still concerns; challenges with many current methods exist by predominantly relying on offline processing, unsuitable amount of processing needed for existing training datasets being obsolete, and using outdated training datasets that are not representative of current network conditions [3, 7, 8]. Together these gaps have created the need for the creation of a new IDS solution, dubbed NetGuardAI, which will provide for the detection of intrusions with very high accuracy (correctly identifying the correct class); while also providing a real-time response capability.

III. Proposed Methodology

NetGuardAI is a real-time intrusion detection system, powered by machine learning algorithms used for the continuous monitoring of live network data that can reliably detect any type of malicious behavior within the network. The entire operational process of the proposed NetGuardAI system can be seen in Figure 1.

The proposed NetGuardAI system uses a data-driven approach to detect and classify all possible types of network attacks, unlike traditional intrusion detection systems (IDS), which typically use a rule/ signature based approach using preset rules with static databases. The flexibility to accommodate rapidly changing and constantly evolving network environments is greatly enhanced with this type of detection. The proposed NetGuardAI system is composed of a multi-stage hierarchical processing pipeline through the following stages: Network Traffic Capture; Flow Creation; Feature Extraction; Data Pre-processing; Model Training & Classification; Anomaly Detection; Response Mechanism, etc.

Phases in the System Process:

- **Collecting Network Traffic:** Wireless packets are being collected on an ongoing basis through packet sniffing methods.
- **Creating Flows:** Wireless packets that have been collected and are grouped into flow using source/Destination IP, Ports, and protocols.
- **Extracting Features:** The mean packet length, flow duration, inter-arrival time, and packet rate are derived from each Flow of collected packets.
- **Preparing Data for the Models:** Min/max normalization, limited missing values, and class imbalance correction are applied to each flow of collected packets.
- **Training and Classifying the Models:** The Classification of the Wireless Network Packets is accomplished through training of each model being used to classify the packets and flowing traffic.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

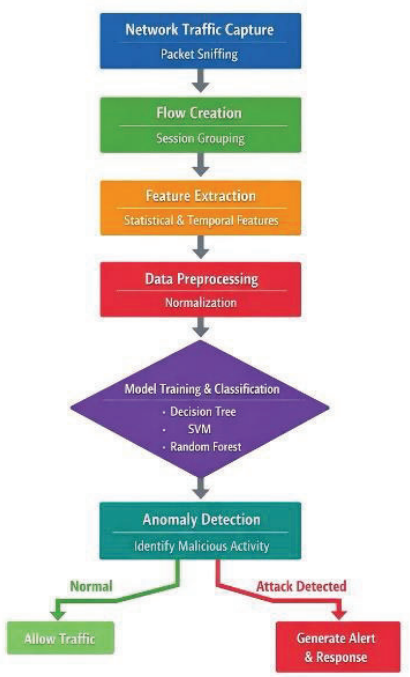

Fig. 1. Proposed NetGuard IDS

- **Identifying and Responding to Anomalies:** Each time an identified traffic anomaly occurs, an open alert is created; otherwise, the flow continues uninterrupted.

IV. Mathematical Formulation

This chapter describes the maths behind the NetGuardAI, showing the process of taking raw (poorly organised) network traffic and creating an ordered (structured) space of features and then using that to run machine learning classification engines against the feature vectors.

A. Flow Representation

Each network flow F represents a communication session between two endpoints, defined as:

$$F = \{IP_s, IP_d, Port_s, Port_d, Proto, T\} \quad (1)$$

where IP_s, IP_d are source/destination IPs, $Port_s$, $Port_d$ are port numbers, $Proto$ is the protocol, and T is the flow duration.

B. Feature Extraction

Each flow F is mapped into a feature vector X comprising statistical and temporal attributes derived from packet-level measurements. Let a flow consist of N packets with sizes x_i, $i = 1,2,\ldots,N$, and timestamps t_i.

1. Mean Packet Length

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (2)$$

2. Standard Deviation of Packet Length

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu)^2} \quad (3)$$

3. Inter-Arrival Time (IAT)

$$IAT = t_{i+1} - t_i \quad , i = 1,2,\ldots..N-1 \quad (4)$$

4. Flow Duration

$$D = t_{end} - t_{start} \quad (5)$$

5. Packet Rate

$$R = \frac{N}{D} \quad , D > 0 \quad (6)$$

Collectively, these features capture both the statistical distribution and the temporal dynamics of network traffic, providing a comprehensive representation for downstream classification.

C. Data Preprocessing

Min-max normalization is applied to ensure consistency across all feature dimensions:

$$X_j' = \frac{X_j - X_j^{min}}{X_j^{max} - X_j^{min}} \quad (7)$$

where X_j is the j^{th} feature value and X_j^{max}, X_j^{min} are its minimum and maximum values respectively.

D. Machine Leraning Models

The classification process assigns the normalized feature vector X' to a binary label $y \in \{0,1\}$, where 0 denotes normal traffic and 1 corresponds to an intrusion.

1. Decision Tree (Entropy)

$$H(S) = -\sum_{k=1}^{C} p_k \log 2\, p_k \quad (8)$$

2. Support Vector Machine (SVM)

$$f(X) = w^T X + b \quad (9)$$

3. Random Forest

$$\hat{y} = mode\{h_t(X')\}_{t=1}^{T} \quad (10)$$

E. Evaluation Metrics

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (11)$$

$$Precision = \frac{TP}{TP+FP} \quad (12)$$

$$Recall = \frac{TP}{TP+FN} \quad (13)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (14)$$

F. Algorithm Design

Algorithm: NetGuardAI Detection Pipeline

Input: Raw network packets

Output: Classification (Normal / Attack) Time

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

TABLE II. HYPERPAREMETERS FOR THE MODELS

Model	Accuracy	Precision	Recall	F1-Score
Decision Tree	92.8%	91.5%	90.9%	91.2%
SVM	94.1%	93.2%	92.4%	92.8%
Random Forest	97.3%	96.8%	95.9%	96.3%

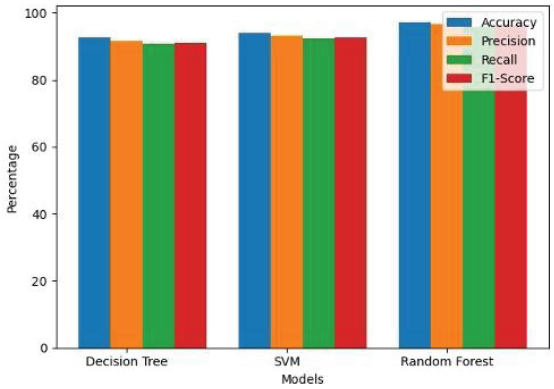

Fig. 2. Performance comparison of ML Models

Complexity: O(N logN)

TABLE I. HYPERPAREMETERS FOR THE MODELS

Model	Parameter	Value Used
Decision Tree	Splitting Criterion Maximum Depth Minimum Samples per Split	Gini 10 2
SVM	Kernel Function Regularization Constant (*C*) Gamma Setting	RBF 1.0 Scale
Random Forest	Total Number of Trees (*T*) Maximum Tree Depth Minimum Samples per Split Maximum Features Considered	100 15 2 $\sqrt{n}$

To create successful classification results, the parameters for each model were developed through parameter tuning as shown in the Table I below. For example, a Radial Basis Function (RBF) used by the SVM model allowed it to account for non-linear relationships in the dataset, while Random Forest used 100 trees to find a good balance between accuracy and computing cost.

V. RESULTS AND DISCUSSION

A. Experimental Setup

To assess the performance of the NetGuardAI network traffic detection system, the team referred to the KDD-NSEL dataset. This dataset includes a wide variety of different types of normal and malicious network traffic patterns. In addition to KDD-NSEL, various other methods were used to preprocess the data, including; min-max normalization, replacing missing values, and eliminating duplicate records. The final data set was divided into training data (80%) and test data (20%). Each flow created 20 features, including both statistical and temporal in nature.

A broad spectrum of performance metrics were used to measure how well the Decision Tree, SVM, and Random Forest classifiers performed against the training data. Metrics used were accuracy, precision, recall, F1-Score, and ROC AUC.

B. Model Performance Comparison

Figure 2 and Table II summarize the prediction accuracy of each of the points listed above by an overall assessment of how well each model predicts. Random Forest continues to show the highest accuracy level of all three models tested and is expected to provide the best general performance of all three models; this reflects the value an ensemble-based approach produces.

C. Interpretation of Results

The variances in the way the two models perform can be attributed to the differences in their design. The Decision Constraint tree and SVM both require generalization to create both types of generalized constraints but have differing amounts of generalization parameters that need to be tuned. Random Forest has advantages over both algorithms because of its constellation properties, which allow it to minimize changes and enhances the stability of the dynamic and unbalanced network traffic dataset.

D. Cross Validation Analysis

TABLE III. 10-FOLD CROSS-VALIDATION RESULTS

Metric	Mean	Std. Deviation
Accuracy	97.1%	±0.42%
Precision	96.5%	±0.38%
Recall	95.7%	±0.51%
F1-Score	96.1%	±0.44%

The data presented in Table III indicates that Random Forest performs in a similar manner across most partitions—i.e. the model's average precision across partitions is nearly 100% with very small differences from each partition (indicating a high degree of generalizability).

E. Feature Importance Analysis

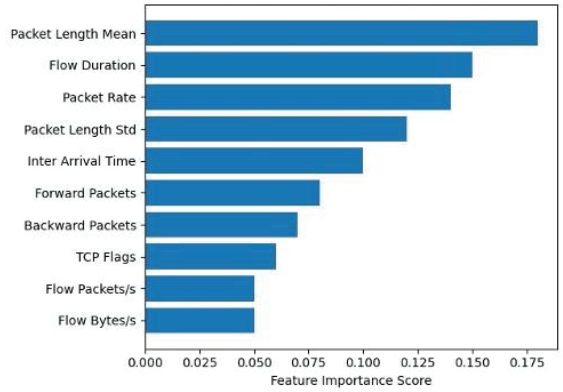

Fig. 3. Feature Importance Score using Random Forest

As illustrated in Fig. 3, packet length mean, flow duration, and packet rate recorded the highest importance scores, confirming their strong discriminative power in distinguishing malicious from normal traffic. These features also enhance model interpretability by identifying the most influential inputs driving classification decisions.

F. Confusion Matrix Analysis

See Fig. 4 for confusion matrix result, which shows 4620 true positive examples and 4850 true negative examples. False positives and false negatives are minimal at 120 and 95. Thus, it can be inferred that reliable detection is possible.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

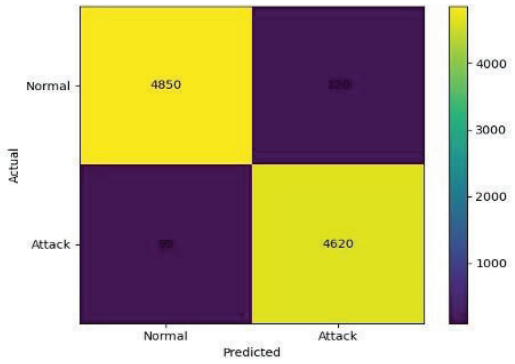

Fig. 6. Confusion Matrix illustrating Random Forest performance

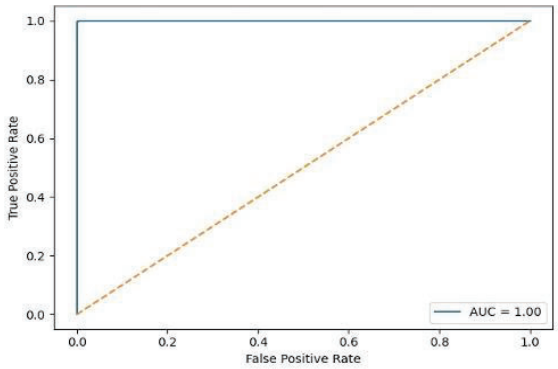

Fig. 7. ROC Curve Analysis of the Random Forest Model

G. *Training Behaviour*

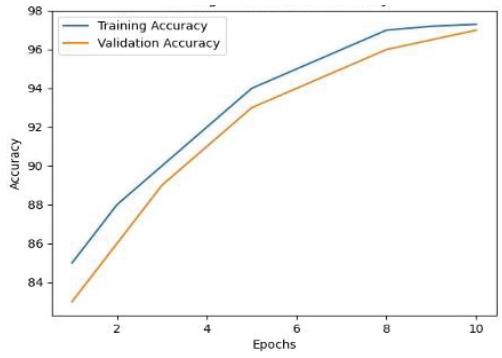

Fig. 5. Training vs Validation Accuracy

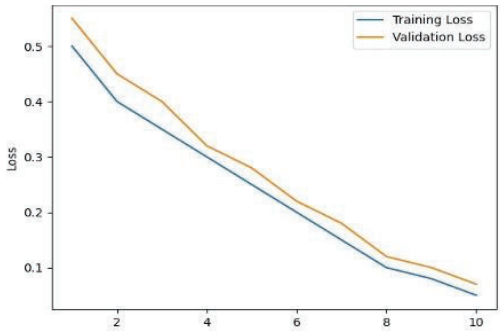

Fig. 4. Loss during Training and Validation

Both through training and validation, the trends in loss and accuracy are consistent as seen in figures 5 and 6. There is good generalization with no evidence of overfitting with both curves closely matching during training.

H. *ROC Curve Analysis*

From the ROC curve of Random Forest shown in Fig. 7, it can be concluded that this model has a significant classification capability as the ROC curve is close to the upper-left corner. Furthermore, an Area Under the Curve (AUC) of 0.98 indicates that the Random Forest is able to classify malicious traffic and legitimate traffic with a high degree of accuracy regardless of the threshold used.

I. *Comparison with Existing Approaches*

As shown in Table IV, NetGuardAI has the complete capacity for real-time processing, as well as has the highest classification accuracy of 97.3%. Although deep learning manages 95%, it has extremely high computational costs. Further, SVM-based techniques achieve 92% accuracy but they lack very much in their ability to operate in real time, while signature-based techniques have a maximum performance of 85% and do not operate in real time. Therefore, only one method truly performs real-time detection with excellent classification accuracy using limited resources, and that is with NetGuardAI.

TABLE IV. COMPARISON WITH EXISTING IDS APPROACHES

Approach	Accuracy	Real-Time Capability
Signature-Based IDS	85%	No
SVM-Based IDS	92%	Limited
Deep Learning IDS	95%	High Cost
Proposed NetGuard AI	97.3%	Yes

J. *Real-Time Capability*

Unlike a batch processing system, where data is processed in batches and analyzed later; NetGuardAI takes a real-time approach to analyzing and classifying all incoming traffic using its own unique, elaborate, and configurable engine.

K. *Critical Discussion*

The only limitation to be noted is that the current evaluation is entirely based upon benchmarks rather than multiple but widely-employed datasets (As such, this does not provide a comprehensive view of all traffic behaviours occurring within a manufacturing setting). In addition, there is no data on how the system will perform with respect to limited hardware, such as edge devices; therefore, more research will be required for this.

VI. CONCLUSION & FUTURE SCOPE

A. *Conclusion*

We have developed an intelligent intrusion detection system that detects and responds to cyber attacks in real time by analyzing network traffic in real time on both the client side (mobile and desktop computers) and server side as one single integrated system. The IDS features a multi-model approach to training the Machine Learning algorithms used by the system. The multi-model allows for the use of a single data pipeline for the preprocessing of data, extraction of data, and final classification of data into threat categories. The system provides real time detection and a common interface for all users regardless of their network size or location (ie onsite or

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

offsite). We demonstrated the validity of the IDS by conducting a series of experiments using the NSL-KDD benchmark dataset. The Random Forest model was the best performing model among the four models we tested. It achieved an overall accuracy of 97.3%, an overall precision of 96.8%, an overall recall of 95.9%, and an overall F1-score of 96.3. We found good consistency in the results of the Random Forest model when performing cross-validation, with a mean precision of 97.1% and a mean standard deviation of + 0.42% (after performing 10 folds of testing) demonstrating high reliability in the results of our study. Our results support our hypothesis that by automating the classification of Cyber-Threats, we can create an effective and efficient Cyber-Defense system.

B. Future Enhancements

Its capacity to service/process and categorize has a process capacity to maintain the ability to handle all traffic, which is considered "post hoc." NetGuardAI's modular structure allows for connection of all parts of the pipeline for as long as a packet exists before being processed by the system. This is where the systems' creation of "threat response" occurs, where practicality assumes a controlled creation of the system. Under current conditions of research experimentation. These shavings of the proposed developed by future models through integration of historically and real-time data will considerably strengthen their performances. CNNs are examples of models using both LSTM and other Deep Learning algorithms to provide additional performance improvements on both long-term and short-term risks. Future Expandability (will yield the future) will occur because of Continuous Active Response Mechanisms. While they evolve, NetGuardAI can develop into a fully-rounded Intrusion Prevention System (IPS) that not only monitors but is capable of acting independently based on all threats to safety eventually eliminated.

REFERENCES

[1] M. Berhili, O. Chaieb, and M. Benabdellah, “Intrusion Detection Systems in IoT Based on Machine Learning: A State-of-the-Art,” *Procedia Computer Science*, 2024.

[2] G. A. Alsharari and A. M. Mostafa, “Developing an Intrusion Detection System for Network Security Using Machine Learning,” *International Journal of Intelligent Systems and Applications*, 2024.

[3] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, “Machine Learning in IoT Security: Current Solutions and Future Challenges,” *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 1–35, 2022.

[4] I. Ullah and Q. H. Mahmoud, “A Hybrid Deep Learning Model for Anomaly-Based Intrusion Detection in IoT Networks,” *Journal of Network and Computer Applications*, vol. 204, 2022.

[5] T. Wu *et al.*, “Intrusion Detection System Combined Enhanced Random Forest with SMOTE Algorithm,” *EURASIP Journal on Advances in Signal Processing*, 2022.

[6] M. Nakip and E. Gelenbe, “Online Self-Supervised Deep Learning for Intrusion Detection Systems,” *IEEE Transactions on Information Forensics and Security*, 2024.

[7] Z. K. Maseer *et al.*, “Meta-Analysis of AI-Based Intrusion Detection Systems: Methods and Challenges,” *IEEE Access*, 2023.

[8] X. Yuan *et al.*, “A Framework to Enhance the Adversarial Robustness of
Deep Learning-Based Intrusion Detection Systems,” *IEEE Access*, 2023.

[9] A. Gueriani, H. Kheddar, and A. C. Mazari, “Deep Reinforcement
Learning for Intrusion Detection in IoT: A Survey,” *IEEE Access*, 2024.

[10] M. B. Suthar and S. Khara, “Enhancing IoT Security through Machine Learning-Based Intrusion Detection Systems,” *Indian Journal of Science and Technology*, 2025.

[11] A. Alabbadi, “An Intrusion Detection System over IoT Data Streams Using Deep Learning and Explainable AI,” *Sensors*, vol. 25, no. 3, 2025.

[12] S. Kakolu, M. A. Faheem, and M. Aslam, “AI-Enabled Intrusion Detection Systems in IoT Networks,” *International Journal of Science and Research Archive*, 2023.

[13] S. Mohanty, S. Kumar, and M. Agarwal, “Enhancing Intrusion Detection Accuracy Using Feature Selection Techniques,” Springer, 2024.

[14] S. Arnob *et al.*, “A Comprehensive Review of Deep Learning-Based Intrusion Detection Systems,” *Journal of Emerging Computing*, 2025.

[15] S. A. Ajagbe *et al.*, “Intrusion Detection System with Feature Selection Using Machine Learning Algorithms,” *ICWETR Conference*, 2025.

[16] CICIDS-2018 Dataset, Canadian Institute for Cybersecurity. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2018.html

[17] CICIDS-2017 Dataset, Canadian Institute for Cybersecurity. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html

(This work is licensed under a Creative Commons Attribution 4.0 International License.)