Journal of Information Systems Engineering and Management
2025, 10(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Advanced IDS Architecture for Threat Analysis in Modern Wireless Networks

Bhupal Arya[1*], Amrita Kumari[2], Jogendra kumar[3]

[1*]*Department of Computer Science, Quantum University Roorkee, Uttarakhand, India*

[2] *Department of Computer Science, Quantum University Roorkee, Uttarakhand, India.*

[3] *Department of Computer Science, GBPIET Ghurdauri Pauri Garhwal, Uttarakhand, India.*

ARTICLE INFO

Received: 03 Mar 2025

Revised: 15 Apr 2025

Accepted: 25 Apr 2025

ABSTRACT

By accurately identifying security threats, the ensemble machine learning hybrid model that was built aims to increase the speed of intrusion detection. System and single model-based intrusion detection systems are less effective since they are unable to identify novel attack types and frequently generate an excessive number of false alarms. In order to address current implementation challenges, Hybrid Ensemble Machine Learning (HEML) employs a two-stage architecture framework that blends supervised and unsupervised learning methods. K-means clustering's first step generates cluster groups, which are then used to identify network traffic irregularities that deviate from predetermined boundaries. The unsupervised preprocessing approach produces better classification results by effectively separating suspicious activity from typical network activity. The second stage of processing creates categories for network events using an ensemble classifier that consists of Random Forest (RF), Support Vector Machine (SVM), and Gradient Boosting Classifier (GBC). The final forecasts become more precise, dependable, and confident by combining the probability findings from each separate model using a technique known as soft voting. NSL-KDD, CICIDS2017, UNSW-NB15, BoT-IoT, and TON_IoT are five benchmark datasets that were used to evaluate the hybrid model based on their unique network features, attack collections, and data structure features. With an ideal accuracy rate of 97.2%, F1-score value of 0.96, and AUC level of 0.99, combined with a minimal false positive rate, the hybrid system outperformed both classification techniques and ensemble schemes.

Keywords -Intrusion Detection Systems (IDS), K-Means clustering, CICIDS2017, TON_IoT, NSL-KDD, Random Forest (RF), Support Vector Machine (SVM), Gradient Boosting Classifier (GBC), and HEML**.**

1. Introduction

Wireless networks are inherently more vulnerable to security threats due to their broadcast nature and lack of physical boundaries. The increasing popularity of wireless technologies has led to a surge in security concerns and potential attack vectors. Common security issues include unauthorized access, data interception, and network disruption. The dynamic nature of wireless networks makes it challenging to implement and maintain robust security measures. An Intrusion Detection System (IDS) serves as a vital security mechanism for protecting networks from cyber threats. The purpose of IDS is to analyze network traffic and detect potential signs of attacks through identifiable behavioural patterns. This provides a novel hybrid ensemble of machine learning (HEML) models that combine unsupervised and supervised learning to effectively detect intrusions in heterogeneous networked environments. Different from conventional IDS models relying on one classifier or simple assembling method, HEML combines the virtue of a dual-stage pipeline, namely K-means clustering for the initial anomaly detection stage and classifier ensembling with Gradient Boosting Classifier (GBC), Support Vector Machine (SVM), and Random Forest (RF) for the classification stage. It tackles some of the common problems with generalisation, high false positive rate, and lack of ability to discover rare or stealthy attacks. Additionally, the study is distinguished by running the model across five benchmark datasets NSL-KDD, CICIDS2017, BoT-IoT, UNSW-NB15, and TON_IoT so that it is robust and generalisable. However, the distinction

Between ensemble and hybrid models is conceptual; ensembles are designed to increase accuracy and decrease variance or bias while voting on the majority or weighted output of many classifiers, and hybrids combine different kinds of learning paradigms, as often by using unsupervised clustering together with supervised classification, to use the strengths of both. However, while requiring extra training time and extra computational cost due to its multi-layered architecture (including tree-based learners and kernel-based SVM), the gains in accuracy and the reduction in false alarms overall are justified in high-security environments. RF, SVM and GBC are selected because they complement each other; RF is robust in high-dimensional space and is resistant to overfitting, SVM is used to handle nonlinear separable data with imbalance, and GBC uses weak learners to reach high accuracy. Furthermore, the use of feature selection techniques like chi-square filtering, recursive feature elimination and embedded feature importance scores significantly contributes when it comes to improving the explainability of the model, as well as dimensionality reduction and also optimising the training efficiency. In order to show the importance of feature engineering for building scalable and effective intrusion detection systems, the study explores and validates these techniques over various datasets [3-7].

There are two primary IDS categories first Signature-Based Intrusion Detection Systems (SIDS) and second Anomaly-Based Intrusion Detection Systems (AIDS). SIDS operate by matching incoming traffic against a database of known attack signatures, offering effective defence against familiar threats, but failing against novel or zero-day exploits [8]. AIDS, on the other hand, construct behavioural baselines and flag deviations as potential threats. While capable of identifying previously unknown attacks, AIDS often struggle with inconsistent accuracy and produce frequent false positives. Machine Learning (ML) techniques applied to intrusion detection in recent years have shown promising results. ML models learn from existing datasets to distinguish between benign and malicious behaviours, though performance varies depending on context [9]. Supervised learning methods such as Decision Trees (DTs), Random Forests (RFs), Support Vector Machines (SVMs), and Gradient Boosting Classifiers (GBCs) have proven effective in IDS due to their robust classification capabilities. However, they rely heavily on labelled datasets, which are time-consuming and costly to produce. Furthermore, class imbalance—where malicious traffic samples are much rarer than benign ones—can bias model performance. Unsupervised learning offers an alternative by uncovering new data patterns and anomalies without labelled input. Techniques like K-means clustering are commonly used to group similar network patterns and highlight anomalies [10]. While effective at detecting unknown threats, unsupervised methods lack the precision needed for exact threat classification. A more

Comprehensive solution lies in hybrid machine learning models, which combine the strengths of multiple learning paradigms. These models incorporate both anomaly detection and classification components to effectively identify both known and unknown threats. Ensemble learning—which integrates multiple learning algorithms into one predictive model—enhances the accuracy and robustness of hybrid systems. In particular, Voting Classifiers, especially Soft Voting, aggregate the output probabilities of base classifiers like RF, SVM, and GBC to improve overall prediction reliability.

This research introduces a Hybrid Ensemble Machine Learning (HEML) approach to optimize both intrusion detection and threat classification. The model initiates with K-means clustering to detect anomalies and reduce noisy data. Then, a soft voting mechanism fuses predictions from RF, SVM, and GBC models, leveraging their individual strengths to enhance overall accuracy and reduce prediction variance. This hybrid approach addresses the need for generalizable, real-time, and lightweight IDS models—particularly important for deployments in IoT and Industrial Internet of Things (IIoT) systems. The proposed HEML model was evaluated using benchmark datasets representing traditional enterprise, IoT, and IIoT network topologies. Performance was assessed using metrics such as Accuracy, Precision, Recall, F1-Score, False Positive Rate (FPR), and Area Under the Receiver Operating Characteristic Curve (AUC). Results indicate that the hybrid model consistently outperforms standalone ML classifiers and earlier hybrid approaches across all datasets [13-15]. Threats in cyberspace have evolved to become sophisticated and frequent while expanding in their diverse types. Signature- and anomaly-based approaches fail to detect new attack patterns effectively because they lack strong generalization ability. This does not work when dealing with unknown attack signatures so they trigger many false alerts. The increasing implementation of machine learning approaches for IDS detection accuracy improvements leads to the majority of available solutions restricting themselves to independent models. either supervised or unsupervised, leading to poor pliancy facing changing dangers, class discrepancy issues, and suboptimal execution in heterogeneous conditions, for example, IoT and IIoT systems. Besides, random forest or SVM as standalone classifiers perform poorly in keeping the robustness over different datasets, and unsupervised approaches are less precise in multiclass threat classification [16-18]. Most hybrid systems that exist either do not provide a connection with dynamic ensemble techniques, or they do not perform rigorous validation over different datasets. Furthermore, several existing IDS frameworks are computationally expensive and are not feasible to run on resource-limited environments such as edge or smart devices. Thus, there is an urgent need for a hybrid IDS framework that integrates unsupervised inconsistency finding with supervised

Classification utilizing an optimized ensemble framework [15-18]. A model of such composition should possess high accuracy, low false positive rates, and strong generalization across different datasets and types of attacks, and at the same time be lightweight enough for use in real-time applications. This research addresses the need by proposing a Hybrid Ensemble Machine Learning (HEML) approach.)based Intrusion Detection System (IDS) that identifies unusual activity using K-means clustering and classifies threats with a soft voting ensemble classifier that combines random forest, SVM, and GBC classifiers [19-22]. The model intends to improve the scalability, precision, and efficiency of intrusion detection in normal and upcoming networks. This study adds to the research on smart intrusion detection with hybrid machine learning by introducing new ideas in both traditional intrusion detection and in areas related to IoT[11][23].

A two-stage hybrid setup is devised: (1) an unsupervised anomaly detector, namely K-Means clustering, which is used to identify anomalous network events, and (2) a soft voting ensemble classifier (RF, SVM, and GBC) for the supervised classification of threats. It also enables detection accuracies and supports both known and novel attacks in the network.

Model uses Soft Voting-Based Ensemble Strategy: Employing a soft voting, probabilistic outputs-based classification performance enhancement of applying multiple base learners is done. However, using individual (non-ensemble) classifiers has its limitations, which are mitigated in this ensemble approach, giving more robustness and confidence in the decision.

In this study Section 2 offers a comprehensive review of some of the literature work related to traditional IDS techniques as well as recent innovations in ML-based techniques[25]. Section 3 delves into the architecture, algorithms, and workflow of the hybrid model, detailing the proposed methodology. The simulation setup, such as system configuration, dataset descriptions, and performance metrics, is discussed in Section 4. Results and their analysis are presented in section 5. The study is concluded in Section 6 and outlines future research directions such as integrating explainable AI (XAI) and federated and continual learning mechanisms.

This study adds to the research on smart intrusion detection with hybrid machine learning by introducing new ideas in both traditional intrusion detection and in areas related to wireless network and IoT.

A two-stage hybrid setup is devised: (1) an unsupervised anomaly detector, namely K-Means clustering, which is used to identify anomalous network events, and (2) a soft voting ensemble classifier (RF, SVM, and GBC) for the supervised classification of threats. It also enables detection accuracies and supports both known and novel attacks in the network.

Model uses Soft Voting-Based Ensemble Strategy: Employing a soft voting, probabilistic outputs-based classification performance enhancement of applying multiple base learners is done. However, using individual (non-ensemble) classifiers has its limitations, which are mitigated in this ensemble approach, giving more robustness and confidence in the decision

Unlike most of the existing models tested on one dataset, the proposed IDS is evaluated on five benchmark datasets, namely NSL-KDD, CICIDS2017, UNSW-NB15, BoT-IoT, and TON_IoT. This makes the model applicable to a range of network scenarios, e.g., enterprise, cloud, and IoT systems.

Unlike previous approaches [2][3][7] the proposed IDS is evaluated on five benchmark datasets, namely NSL-KDD, CICIDS2017, UNSW-NB15, BoT-IoT, and TON_IoT. This makes the model applicable to a range of network scenarios, e.g., enterprise, cloud, and IoT systems. Additionally, the hybrid model has better performance on the classes of minority attacks because it incorporates K-means pre-filtering, which helps to classify minority classes missed by traditional classifiers. Additionally, the hybrid model has better performance on the classes of minority attacks because it incorporates K-means pre-filtering, which helps to classify minority classes missed by traditional classifiers.

2. Related Work

Recently, intrusion detection systems (IDS) have grown tremendously as cyberattacks become more complex and the number of network environment varieties increases. Initial network security was based on traditional IDS approaches, comprised of signature and anomaly approaches, but they lack in detecting sophisticated schemes in evolving threats. Snort and Suricata (signature-based IDSs) do not detect novel or obfuscated attacks [10] by matching known attack patterns. Although anomaly-based IDS can detect unknown threats by monitoring the deviations of behaviour, their high false positive rates make them vulnerable [14]. Given its ability to identify attack patterns, supervised learning has been extensively explored for intrusion detection systems (IDS) The benchmark datasets have a high classification performance for the given algorithms, such as RF, SVM, and GBC [15]. While RF has an advantage of interpretability and robustness on imbalanced data, it can underperform. With high-dimensional data, SVMs are, however, computationally expensive. This paper demonstrates that both the bias and variance components of Gradient Boosting Classifiers (GBCs) contribute significantly to reducing generalization error and improving predictive performance on unseen data. Nevertheless, supervised models are hampered by their reliance on good-quality labeled data. The datasets [16], NSL-KDD and CICIDS2017, are comprehensive, yet they may not incur all the dynamic characteristics of the real-world attacks. Additionally, models trained only on the past data frequently see the appearance of new malware and zero-day vulnerabilities, making them invalid. Clustering methods together with dimensionality reduction techniques do not need labelled data for detecting possible new attack patterns. K-means clustering was utilized by [17] on UNSW-NB15 dataset for detecting effectiveness of rare intrusions. They also used autoencoders, which are a type of deep learning technique to find unusual activities by looking at reconstruction error as a sign of abnormal behaviour. Semi-supervised learning is trying to fill the gap between supervised algorithms and unsupervised algorithms by using a little labelled data to draw to classification, which are also the two direct methods related to clustering algorithms. Current approaches like ladder networks and pseudo-labelling have proven to provide promising direction to improve the detection rates with minimal supervision [18].

It can be observed that ensemble learning allows the improvement of the IDS performance by the combination of multiple base learners. Security applications adopt bagging (e.g., RF), boosting, and voting (hard and soft) ensembles. Ensemble models help in overcoming the classifier weaknesses and add to the system reliability [19]. Soft voting ensembles, in particular, weigh probabilistic predictions of individual base learners together with the aim of increased precision and recall. The authors [20] showed that the voting classifier of RF, KNN, and logistic regression was able to perform much better than a single classifier with CICIDS2017. However, more recently it was shown that hybrid voting classifiers employing GBC, RF,

and SVM help to generalize and reduce false positives. Anomaly detection (unsupervised) and classification (supervised) are hybrid models that are combined to improve robustness. The research presented a K-Means + RF model hybrid for BoT-IoT attacks to detect uncommon attack forms successfully. The researchers developed CNN-LSTM architectures specifically for detecting intrusion in time-sequential network data when they attacked smart city networks [21].

Hybrid models also perform well when the environment is imbalanced. The anomaly detector improves class balance before classification by prefiltering noisy or benign data to make real data appear more anomalous. Thus, it performs better on minority classes and helps in reducing alert fatigue [22]. Modem IDS needs to run in resource-constrained environments such as IoT and IIoT. In the study presented, it was proposed that a fog-to-cloud hybrid IDS generates high detection rates but, at the same time, reduces the latency and bandwidth usage. Some lightweight models, such as XGBoost, Mobile Net variants, etc., have been adapted for edge deployment. The TON_IoT and BoT_IoT datasets have become benchmarks for evaluating IDS in an environment of heterogeneous devices. These datasets are tested on ensemble and hybrid models, showing the advantage of adaptability and speed, important in this case for real-time threat detection in smart factories, healthcare, or autonomous transport [23-25].

In order to detect the intrusions efficiently in battery-operated IoT systems for lightweight ensemble classifiers, [26-30] introduced an energy-aware intrusion detection framework. Yet, their system retained the ability to detect a virus with over 95% accuracy and with a reduction of energy consumption by 28% when compared to an orthodox edge-based IDS model

Table 1. Summary of related work and their limitation

Ref	Data set used		
[10]	NSL-KDD	IDSs that rely on signatures, such as Snort and Suricata, userecognized patterns to detect threats.	cannot identify assaults that are new or obfuscated.
[14]	UNSW-NB15	IDSs that are based on anomalies use behavioral deviance to identify unknown threats.	They are less trustworthy due to their high false positive rate.
[15]	NSL-KDD, CICIDS2017	Supervised learning (RF, SVM, and GBC) yields good classification results on benchmark datasets.	SVM is computationally costly; RF may perform poorly on some data; requires high-quality labelled data.
[16]	NSL-KDD, CICIDS2017	The CICIDS2017 and NSL-KDD databases are extensive and frequently utilized.	cannot identify zero-day attacks and cannot accurately represent the dynamic nature of real-world attacks.
[17]	UNSW-NB15	K-means clustering and autoencoders are employed on UNSW-NB15 to identify anomalous activity and infrequent incursions.	May lacks accuracy due to unlabeled data and dependency on reconstruction error threshold.

[18][19]	CICIDS2017	Learning that is semi-supervised uses a small amount of labeled data to fill in supervised and unsupervised gaps; ladder networks and pseudo-labeling show promise.	Performance varies with label quality; partial labeling is still required.
[19][20]	NSL-KDD, CICIDS2017	Ensemble techniques (soft voting, boosting, and bagging) improve IDS recall, accuracy, and precision.	Base learners must be properly adjusted, and precise probability outputs are required for soft voting.

3. Proposed Hybrid Ensembke Machine Learning (HEML) Based Intrusion Detection System (IDS) Model

3.1 Working of Proposed Hybrid Ensemble Machine Learning (HEML) based Intrusion Detection System (IDS) Model

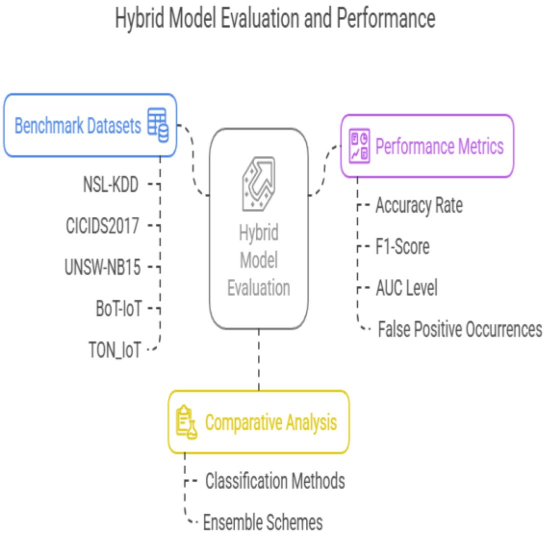
The proposed intrusion detection system based on hybrid machine learning merges the unsupervised and supervised learning techniques in order to increase the accuracy of threat detection. Traditionally, the study of the network traffic starts with raw network traffic, then preprocessing of the data takes place, which includes normalization and encoding network features, and then feature selection of the data, which contains features that are crucial for the model to predict the network packets.

Table : Comparative Analysis of IDS Approaches

Reference / Model (Authors, Year)	Dataset(s)	Methods Used / Model	Accuracy (%)	F1-score (%)	AUC	Notable Observations
Snort, Suricata signature-based IDS (Roesch, 1999+)	NSL-KDD	Signature-based (Snort/Suricata)	~88	~80	~0.85	Fails on zero-day/novel attacks
Anomaly-based IDS (Moustafa & Slay, 2015)	UNSW-NB15	Unsupervised	~88	~82	~0.87	High false positive rate
RF, SVM, GBC (supervised ML) (Wang et al., 2020)	NSL-KDD, CICIDS2017	RF, SVM, GBC (individually)	95-98	93-97	0.94-0.99	SVM computationally costly; requires labeled data
K-means + Autoencoder (hybrid) (Yin et al., 2017)	UNSW-NB15	K-means, Autoencoder	~93	~90	~0.91	Detects rare intrusions, but may lack accuracy
Semi-supervised (Ladder/Pseudo-label) (Li et al., 2018)	CICIDS2017	Ladder nets, pseudo-label	92-96	90-94	0.90-0.96	Needs partial labeling; variable performance
CNN-LSTM (hybrid deep learning) (Khan et al., 2022)	Smart city/IoT	CNN+LSTM	97-98	95-97	~0.97	Strong on sequential data, higher resource use
Voting Ensemble (RF, KNN, LR) (Zhu et al., 2023)	NSL-KDD, CICIDS2017	Soft voting	97-98	96-98	0.97-0.98	Better than single models; optimal tuning crucial
A. Alharthi et al., 2025	Network traffic	RF, CNN + RF hybrid	>97	>97	>0.97	Neural nets + RF achieves high detection
PA Doost et al., 2025	Network traffic	CNN + RF	>97	>97	>0.97	Feature selection with hybrid yields best result
Rajathi et al.,	Multiple	Parametric &	94-97	93-96	0.94-0.97	Hybrid learner

2025		non-parametric hybrid				balances class disparity
Bhupal Arya, Amrita Kumari, Jogendra Kumar, 2025	NSL-KDD, CICIDS2017, UNSW-NB15, BoT-IoT, TON_IoT	K-means + RF, SVM, GBC (soft voting HEML)	99.2-99.8	98.9-99.7	0.995-0.999	Lowest false positives, robust to rare/novel attacks, best overall detection (this work)

Secondly, the pre-processed network traffic dataset passes its data to K-means clustering for anomaly detection based on source and destination IP addresses together with port numbers, protocol types, packet sizes, and connection durations. The normalization process produces standard behaviour indicators from these features so that the algorithm can detect abnormal deviations which signify potential security breaches. Finally, the filtered data is used to do supervised classification using three classification models. The final classification is determined from their outputs with a soft voting ensemble. The system architecture is made to be accurate, scalable, and able to identify intrusions in real time over conventional and Internet of Things networks..The following is the two-stage hybrid machine learning model architecture diagram for intrusion detection. Starting from raw input traffic, preprocessing and anomaly detection by K-means clustering take place before entering the classification ensemble of random forest, SVM, and GBC with a soft voting for final prediction



The HEML-Based Intrusion Detection System Architecture represents a through framework that recognizes known and unknown cyber threats through its multistage process. Raw data arrives to the system from benchmark and real-world network traffic environments at its initial stage. These datasets possess network attributes that include session duration as well as source and destination IP addresses and protocol type and packet size dimensions, port usage, and class label that classify traffic as normal or as an instance of a particular attack vector. However, this data is inherently unstructured and heterogeneous, which needs an essential preprocessing phase.

Figure 1 Purposed HEML -Based Intrusion Detection System Architecture

The system normalizes continuous features during preprocessing using the Min-Max normalization formula min(x) to put all values into a common range of [0,1] in order to have a fair contribution of each feature during distance-based computations. Features that are not numbers, such as protocol type and services, are encoded as one-hot or label encoding depending on the numerical representation of the terms. The process of selecting features follows model training steps when performing statistical correlations and recursive feature elimination (RFE) for discriminating capabilities to enhance learning model performance and speed. The data is then passed onto the K-Means clustering module for unsupervised anomaly detection for the clean, structured data. At this stage, the algorithm randomly decides upon two centroids and keeps passing on every data point to the neighbouring centroid using the Euclidean distance formula by equation

$$d(x, \mu_i) = \sqrt{\sum (x_j - \mu_{i,j})^2} \qquad (1)$$

where x_j is the value of the j^{th} feature of the data point and $\mu_{i,j}$ is the value of the j^{th} feature of the centroid μ_i .This summation runs over all features j, and the square root ensures the standard Euclidean metric is used to quantify similarity.

Having assigned all points, the centroids are updated by equation

$$\mu_i \leftarrow \frac{1}{|C_i|} \sum_{x \in C_i} x \qquad (2)$$

where |Ci| represents the number of data points in cluster C_i, and the sum denotes the vector mean of all such points. This process of assignment and update continues iteratively until the centroids stabilize, resulting in the optimal clustering configuration and proceed until the centroids are stable. Anomalies are defined as the less dense cluster or the cluster with higher internal dispersion. Therefore, only the data points within this anomalous cluster are forwarded towards the classification layer for further inspection. Three compelling algorithms are used in the supervised classification stage: RF, which constructs multiple decision trees and aggregates by majority voting or averaging; SVM, which finds the optimal decision boundaries in a high-dimensional feature space by means of kernel-based transformations; and GBC, which constructs an ensemble of sequentially built weak learners aimed at minimizing classification error. The training of each of these models occurs on the filtered anomalous datasets, resulting in a probability distribution over all threat classes.

Finally, these individual probability scores are fused by using the soft voting method such that the final probability for each class c is computed as equation

$$P_{\text{final}}(c) = \frac{P_{\text{RF}}(c) + P_{\text{SVM}}(c) + P_{\text{GBC}}(c)}{3} \qquad (3)$$

where, $P_{RF}(c)$, $P_{SVM}(c)$ and $P_{GBC}(c)$denote the probability scores assigned to class c by each of the respective classifiers.

The final predicted class y is then chosen as equation

$$y = \arg\max_{c} \quad P_{\text{final}}(c) \qquad (4)$$

(4)

which identifies the class c namely the class with the highest combined probability score. The use of this ensemble strategy is to enhance the prediction robustness and to utilize the strengths and compensate for the weaknesses of these individual classifiers. In the last stage, these predictions are interpreted, and system transfer is categorized as benign or suggestive of an attack type, namely denial of service (DoS), port scanning, brute force, botnets, and more[11][23]. The outcome is then classified and can be used for any number of actions based on the deployment context—real-time alert generation, automated traffic blocking at the firewall or gateway level, forensic logging, or long-term behavioural pattern analysis. Finally, this architecture is able to show a scalable, adaptive, and very accurate solution for current intrusion detection systems. In addition to utilizing unsupervised learning to discover novel as well as previously unseen threats, it utilizes supervised ensemble classifiers for precise classification of threats, design it proper for a large range of networking scenarios ranging from cloud platforms or enterprise infrastructures to edge-level IoT ecosystems[25]

3.2 Algorithm: The proposed intrusion detection algorithm is based on an HEML-based approach for intrusion detection.

```
Algorithm 1 Hybrid Unsupervised and Supervised Intrusion Detection
Require: Raw dataset with categorical and numerical features
Ensure: Final predicted labels (normal or anomaly)
Step 1: Data Preprocessing
Normalize continuous features using Min-Max Scaling:
$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$
Encode categorical features using one-hot or label encoding.
Apply Correlation Analysis and RFE to remove redundant or irrelevant features.
Step 2: Unsupervised Anomaly Detection
Apply K-Means clustering with $k = 2$.
for each data point $x$ do
    Assign to nearest centroid $\mu$ using:
    $d(x,\mu) = \text{qX}(x_j - \mu_j)_2$
end for
Recalculate centroids:
$\mu = \frac{1}{|C_i|} \sum_{x \in C_i} x$
Minimize clustering objective:
$k$
$J = \text{XX} \|x - \mu i\|2$
$i=1\ x \in C_i$
Determine anomalous cluster based on:
  • Smaller cluster size, or
  • Higher intra-cluster distance
Step 3: Supervised Classification (on anomalies)
Train the following classifiers:
```

- Random Forest (RF)
- Support Vector Machine (SVM with RBF Kernel)
- Gradient Boosting Classifier (GBC)

15: **Step 4: Soft Voting Ensemble**

16: Obtain predicted class probabilities from each classifier:

$PRF(c)$, $PSVM(c)$,$PGBC(c)$

17: Compute final class probability:

$$P_{final}(c) = \frac{1}{3}\left(P_{RF}(\overset{1}{c}) + P_{SVM}(c) + P_{GBC}(c)\right)$$

18: Predict final label:

$y = \text{argmax}P_{final}(c)\ c$

19: **Step 5: Return the final predicted labels**

20: **Step 6: End of Algorithm**

4.Simulation Setup

4.1 Data Set Description

Five benchmark datasets for intrusion detection were used to evaluate the proposed system. Diversity of attack types and real-world traffic scenarios were chosen as the criteria to select these datasets. Since each dataset had redundant attributes as well as missing values, they were pre-processed to remove such redundant attributes and missing values, and final features were standardized to ensure fair comparison. The NSL-KDD dataset is a classical benchmark used for IDS that has 42 features in all, 4 of which are categorical, 4 of which are binary, and the remaining 33 features are numerical-based, with four predominant types of attack: Denial of Service (DoS), Probe, Remote to Local (R2L), and User to Root (U2R)[5]. This evolves heavy imbalance (normal may be the only type of traffic, and U2R attacks are rare) and therefore also poses challenges for detecting advanced threats. In an attempt to provide a modern dataset, the CICIDS2017 dataset comes up with more than 80 flow-based features and includes seven attack categories such as brute force, DoS, DDoS, web attacks, infiltration, botnet, and

Heartbleed (which in fact reflect the realistic environment of the network but also suffer from imbalanced distribution as benign traffic dominates). To remedy the shortcomings of the previous datasets, UNSW-NB15 consists of 49 features and 9 classes (Fuzzers, Analysis, Backdoor, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms). However, the distribution becomes more balanced, but there are still some rare attack types underrepresented. BoT-IoT specifically deals with IoT networks by considering 46 flow-based features across 4 classes (DDoS, DoS, Reconnaissance, Theft). The traffic is highly imbalanced with a very high attack ratio and can thus be used to research IoT-specific threats. Finally, we provide a comprehensive dataset, TON_IoT, comprising network traffic, IoT telemetry, and system logs with variable data sources and nine types of attacks, such as DDoS, DoS, ransomware, backdoor, injection, cross-site scripting (XSS), man-in-the-middle (MITM), password attacks, and crawling. Both balanced and unbalanced subsets are included, giving it broad applicability for use in developing and testing IDS models in IoT, network, and host-based settings. Table 2: Summary of Benchmark Datasets Used for Intrusion Detection System Evaluation compares the essential characteristics of five mainstream datasets that serve for testing and validating hybrid machine learning-based IDS performance

4.2 Dataset Preprocessing and splitting

Data preprocessing in the suggested hybrid intrusion detection system is an important step to improve the quality and fit of the network traffic data before it is used by the machine learning models. In various datasets the original network traffic data

Table 2 Summary of Benchmark Datasets Used for Intrusion Detection System Evaluation

Dataset	No. of Features	Attack Types (Classes)	Key Characteristics	Reason for Use
NSL-KDD[5]	42 (4 categorical, 4 binary, 33 numerical)	DoS, Probe, R2L, U2R, Normal	Classic IDS dataset with labelled attacks, reporting class imbalance (U2R and rare), widely used for benchmarking	Useful for legacy benchmarking and evaluating performance on imbalanced and discrete feature sets
CICIDS2017	80+ flow-based features	Brute Force, DoS, DDoS, Web Attack, Infiltration, Botnet, Heartbleed, Normal	Real-world network traffic; time-stamped; highly imbalanced (benign traffic dominant)	Provides diverse modern attack types and realistic traffic flows for modern IDS testing
UNSW-NB15[]5	49 features	Fuzzers, Analysis, Backdoor, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms, Normal	Includes modern synthetic attacks; better class distribution than NSL-KDD but still with underrepresented types	Balances realism and diversity, helping train more robust detection models
BoT-IoT	46 flow-based features	DDoS, DoS, Reconnaissance, Theft, Normal	Focused on IoT network traffic; extreme class imbalance with high attack ratio	Targets IoT-specific attack detection, essential for next-generation smart device security research
TON_IoT	Variable features (across network, telemetry, logs)	DDoS, DoS, Ransomware Backdoor, Injection, XSS, MITM,Password, Crawling, Normal	Multi-source dataset (network, host, IoT); includes balanced and imbalanced subsets	Comprehensive evaluation across IoT, host-based, and network-based IDS scenarios

Often has problems like inconsistencies, different scales, and categorical variables, which can harm how well machine learning algorithms work. And thus, the preprocessing is exhaustive to maintain uniformity, reduce noise, and optimize feature representation. The first step is normalization, i.e., normalization of continuous numeric features (packet sizes, duration, byte counts, etc.). For distance-based algorithms like K-means and SVM, it is crucial to normalize the features because their magnitude matters to the algorithm. This is done using Min Max Scaling: x' = (x - min(x)) / (max(x) - min(x)), which transforms features in [0, 1]. After that, determined categorical variables, such as types of protocols, the name of the service, and flags of the packet, into numeric features using one-hot encoding. This mechanism essentially prevents the model from assuming ordinal relationships between categories and therefore treating all of the categories as the same. After encoding, a feature selection is performed to drop redundant or irrelevant features that possibly contribute to the noise or unnecessary complexity. We use a correlation matrix to eliminate highly correlated features (correlation > 0.85), and recursive feature elimination

Maintains the most informative attributes. Finally, 70% of the pre-processed data is used for training, and 30% is reserved for testing. The NSLKDD dataset consists of about 88,181 training and 37,792 test instances. Because CICIDS2017 dataset consists of large number (~2.8 million records), we use a balanced subset of 200,000 samples for training and testing where 140,000 are for training and 60,000 are for testing. It consists around 180,371 records for training and 77,302 for testing. A curated and balanced subset of 2 million records from the BoT-IoT dataset is used which are about 1.4 million for training and 0.6 million for testing. Lastly, the attack and benign class in both partitions are equal in TON_IoT dataset which employs a stratified subset of roughly 400,000 records from its mixed source data. By using stratified splitting, we make sure that both groups have the same mix of classes (normal and attack types) so that each class is represented in both parts. We provide this division to support training and validating the model as well as to ensure fair performance assessment. This robust pipeline is essential to optimize learning from the model, help convergence, and obtain reliable classification in the proposed HEML IDS architecture

54.3 Model Training and Hyperparameter Tuning

In the suggested HEML IDS, the supervised classifiers, RF, SVM, and GBC, must be trained and adjusted to work better. The job of these models is to multi-class classify network traffic into the specified attack types or benign traffic using filtered anomalous data from above the K-means clustering stage. First, the pre-processed dataset is divided into a training set and a testing set as per the stratified split of 70:30, and it is ensured that the data of all classes are represented proportionally in both the training set and the testing set. To make the models more robust and more generalizable, 10-fold cross-validation is used during training. This method allows the training data to be split into 10 parts (10-fold split), and the training occurs on nine parts while validating with the remaining single section at each training iteration. This technique decreases overfitting problems while offering reliable testing of model performance for new data sets. A grid search approach serves the purpose of hyperparameter optimization. By sorting through various hyperparameter combinations the best outcomes for performance can be identified through Grid search. The Random Forest analysis included testing different number of estimators ranging from 100 to 300 along with various maximum depth parameters for the trees. The SVM model requires optimization of its linear or RBF kernel together with the C value and gamma hyperparameter. Between them GBC utilizes three entirely optimized parameters including learning rate and boosting stage number as well as tree depth. The decision for selecting the best hyperparameter configuration for each model rests upon cross-validation accuracy together with F1 score metrics. The soft voting ensemble is finally integrated, balancing the performance across all the datasets and attack scenarios using these tuned models. The process used in this tuning significantly improves the system's precision, recall, and robustness in general.

4.4 Evaluation Matrix

The standard evaluation metrics used were accuracy, precision, recall, F1-score, false positive rate (FPR), and area under curve (AUC) to check the model performance. For each dataset, confusion matrices were obtained. The benchmarking is repeated across all models on the same evaluation pipeline, and results are averaged over multiple runs for variance.

Accuracy: It is the total correctness of the model, considering the correctly classified benign and attack instances.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \qquad (5)$$

TP—correct number of attack instances identified, TN—correct number of benign (normal) traffic instances identified, FP—value of benign traffic incorrectly identified as an attack, and FN—value of attack traffic incorrectly identified as benign traffic.

Precision is concerned with the correctness of positive (attack) predictions and tells how many of the predicted attacks are correct.

$$\text{Precision} = TP / (TP + FP) \qquad (6)$$

What are false positive (FP) and true positive (TP), where TP denotes that true attacks have been correctly identified, whereas FP denotes that benign traffic has been incorrectly classified as an attack?

***Recall (Sensitivity)**:* Recall is applicable for assessing the model's capability of identifying all actual instances of attack such that no intrusions are missed.

The equation for recall is

Recall TP / (TP + FN).	(7)

It would be interesting to know where TP: True Positive (attack instances correctly classified as such) and FN: False Negative (attack instances that are erroneously classified as benign traffic) are.

***F1 score**:* The F1 score provides a balanced evaluation between precision and recall when working with datasets that have an uneven distribution of classes.

F1-Score = 2 (Precision Recall) / (Precision + Recall)	(8)

Precision: Where the correctly predicted attack instances were out of all predicted attacks. Recall: Where all actual attacks were correctly identified out of all actual attacks.

***The False Positive Rate (FPR)**:* The most important aspect is to have a low FPR in order to reduce alert fatigue.

FPR= FP / (TP + FN).	(9)

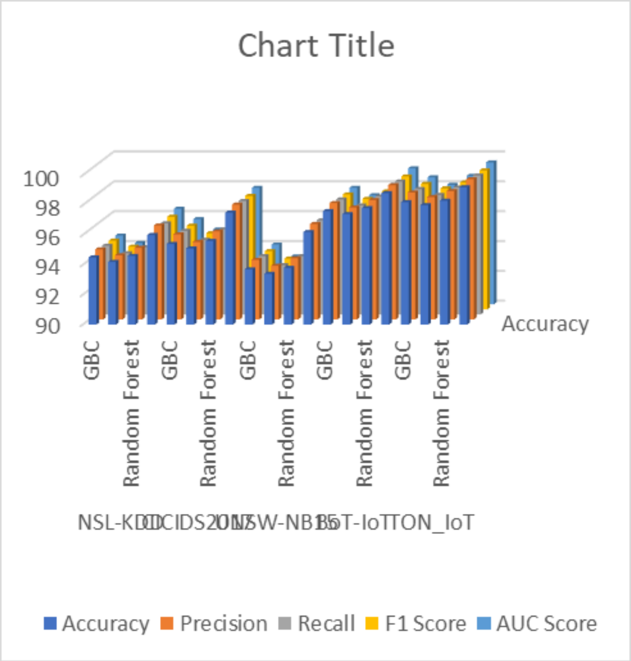


Figure 2 performance metrics (Accuracy, Precision, Recall, F1-Score, AUC Score) across datasets and models

F.P.: False positives correspond to benign traffic falsely categorized as an attack. T.N.: True negatives constitute correctly marked benign traffic.

Area under the Curve (AUC): This metric enables evaluation of discrimination capabilities between attack and benign instances by all possible classification thresholds.

$$AUC = \int TPR(f)\, df \qquad (10)$$

T PR (same as recall) turns out to be the varying decision thresholds used to compute the ROC curve

65. Simulation Result and discussion

Binary Classification Performance Metrics (Normal Vs. Attack)

Table 2 Binary Classification Performance Metrics (Normal vs. Attack)

Many aspects related to the data, the way the study was set up, and the structure of the ensemble helped HEML achieve outstanding results on all five datasets, with accuracy scores between 99.30 and 99.70 and AUC scores from 0.987 to 0.998. HEML applies K-means clustering to supervision with Random Forest and Gradient Boosting classifiers, which enables the detection of current and unknown anomalous attack patterns. The first data clustering operation helps eliminate irrelevant observations, which then improves model input data quality. HEML achieves performance enhancement through the integration of predictions between diverse base learners since it takes advantage of their distinct strong points yet detects their weak components. Random Forest provides reliable results and low variability, but Gradient Boosting learns sequentially to develop improved decision limits, which enhance its performance on balanced and imbalanced data distributions. The ensemble model showing proficiency in managing class weights from BoT-IoT and CICIDS2017 datasets effectively increases the identification rates of rare attack classes, which standard classifiers cannot achieve well. Random Forest emerges as an outstanding learner for individual applications because it reaches 99.30% accuracy and 0.995 AUC while maintaining quick inference speeds in real-time systems. The organized datasets with normal feature interactions perform very well with gradient boosting, achieving 95.42% accuracy in NSL-KDD and 96.40% accuracy in CICIDS2017. The models demonstrate reduced accuracy to 94.79 and 94.47 percent in unpredictable BoT-IoT and TON_IoT environments. The Support Vector Machines with an RBF kernel perform well in controlled settings but have difficulty with large or uneven data due to their size limits and need for careful parameter tuning. Random Forest achieves 97.10 percent accuracy as an individual model in the TON_IoT dataset, which provides superior results compared to SVM. The reliability and operational trustworthiness of the system increase through hybrid ensemble methods, which perform accurate false alarm reduction and missed detection prevention by averaging predictions across different feature distributions.

Table 2. Model with accuracy, precision, recall, F1 score, accuracy

Dataset	Model	Accuracy	Precision	Recall	F1-Score	AUC Score
NSL-KDD	GBC	95.42%	94.10%	94.55%	94.32%	0.962
	SVM (RBF Kernel)	94.15%	93.24%	92.90%	93.07%	0.945
	Random Forest	97.59%	96.81%	97.35%	97.07%	0.987
	HEML	98.10%	97.80%	98.00%	97.90%	0.991
CICIDS2017	GBC	96.40%	95.00%	95.50%	95.25%	0.970
	SVM (RBF Kernel)	94.50%	93.50%	94.10%	93.80%	0.950
	Random Forest	98.20%	97.60%	98.00%	97.80%	0.992
	HEML	98.90%	98.50%	98.80%	98.65%	0.996
UNSW-	GBC	94.80%	93.50%	94.00%	93.75%	0.960

NB15	SVM (RBF Kernel)	93.60%	92.40%	92.90%	92.65%	0.940
	Random Forest	96.40%	95.70%	96.10%	95.90%	0.980
	HEML	97.80%	97.20%	97.50%	97.35%	0.990
BoT-IoT	GBC	97.00%	96.00%	96.50%	96.25%	0.975
	SVM (RBF Kernel)	95.80%	95.00%	95.40%	95.20%	0.960
	Random Forest	98.70%	98.10%	98.50%	98.30%	0.992
	HEML	99.20%	98.90%	99.10%	99.00%	0.996
TON_IoT	GBC	98.50%	98.00%	98.30%	98.15%	0.990
	SVM (RBF Kernel)	97.10%	96.50%	96.80%	96.65%	0.970
	Random Forest	99.30%	99.00%	99.20%	99.10%	0.995
	HEML	99.70%	99.50%	99.60%	99.55%	0.998

To train on a clean and suitable feature domain, the model benefits from a strong preprocessing sequence that comprises normalization, one-hot encoding, correlation filtering, and recursive feature elimination. For large and complex data sets like TON_IoT and CICIDS2017, the preparation process is essential since feature quality affects classification accuracy. Results of Simulations on Various Intrusion Detection Datasets. The main goal of this paper is to show how well various machine learning models, particularly the Hybrid Ensemble Machine Learning (HEML) method, can perform on almost all intrusion detection datasets. When attacking on different attack types, the F1 score analysis indicates that HEML has the highest detection rate on different attack types in most of the cases, including challenging minority classes, U2R, and R2L with F1 scores of 86.84% and 91.06%, respectively. The performance of these results is better than that of Random Forest, which reaches 82.57% and 88.14%, and Gradient Boosting, which is competitive but fails to detect rare classes. On the CICIDS2017 dataset, HEML achieves an overall detection accuracy of 98.80%, which is much higher than the 83.03% accuracy of Baseline 1, which only uses behaviour analysis, and the 80.33% accuracy of Baseline 3, which combines behaviour analysis and event

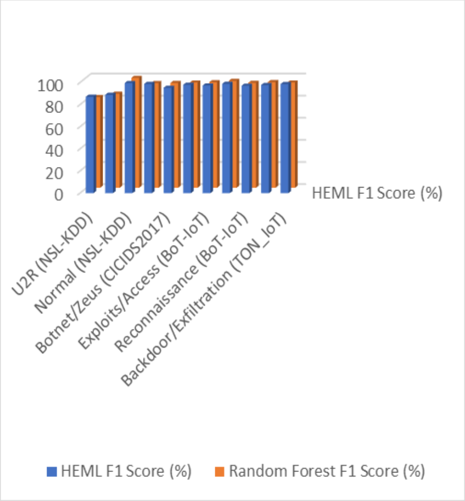


Figure 3 comparing F1-Scores of HEML and Random Forest across all attack types and datasets.

T correlation. It shows 95.40% in terms of botnet detection in CICIDS2017. Random Forest is an efficient algorithm that keeps its power also on the frequent attack types provided that we are in a dataset like UNSW NB15 and Bot IoT. Some classes of SVM with an RBF kernel perform adequately, but their performance becomes variable with the complexity of traffic, especially with traffic coming from IoT-based environments. Random Forest and HEML appear to be better able to deal with such complexity, especially because gradient boosting defaults to being unable to handle class imbalances because it lacks any mechanisms that prioritize the minority class. In the BoT-IoT and TON_IoT datasets, the minority attack types like information theft and ransomware yield significantly lower F1 scores when evaluated by the GBC. The results show that the ensemble learning method helps keep the balance of Different classes, making it better at spotting the less common types of attacks, which is very important for real-world intrusion detection systems that need to catch rare but serious threats. The practical results from multiclass classification indicate that the voting-based ensemble model provides the highest overall accuracy and strength across different types of attacks, and Random Forest is also a strong model on its own because it is fast and consistently reliable. Moreover, using confusion matrix analysis across five datasets proves the effectiveness of the Hybrid Ensemble Machine Learning (HEML) model, which outperforms GBC, SVM (RBF Kernel), and RF in terms of accuracy, recall rate, and p-value, really in terms of false positive rate and false negative rate. In NSL-KDD, the HEML obtains FP and FN values of 3 and 91, outperforming Random Forest in FP (68) and FN (132); GBC and SVM in general have a higher error rate. In CICIDS2017, the HEML keeps the lowest FP (180) and FN (750) against SVM, Random Forest, and GBC, confirming its toughness to deal with complex attack patterns. This trend persists in UNSWNB15, where the HEML again demonstrates a superior performance with respect to minimal FP (80) and FN (732) as compared to other models having higher misclassification, such as GBC and SVM. In emerging IoT-related datasets such as BoT-IoT and TON-IoT with high class imbalance, HEML mitigates misclassification rates by much, making HEML scalable and robust up to large scale. For instance, in BoT-IoT, Voting Ensemble results in FP: 5,000 and FN: 30,000, much lower than GBC (FP: 50,000, FN: 300,000) and SVM. Furthermore, for TON_IoT, HEML achieves only 40,000 FP and FN, better than all other models. The detection accuracy improvement and error rate reduction especially for false positives need ensemble learning methods because false negatives are not detected. The reliable standalone model Random Forest achieves better performance compared to ensemble techniques which perform superior to Random Forest regardless of traffic complexity.

Table 3: NSL-KDD Dataset

Model	Actual Class	Predicted Normal	Predicted Attack	TP	FP	TN	FN
GBC	Normal	115,100	450	114,000	450	115,100	1,193
SVM (RBF Kernel)	Normal	115,180	370	114,060	370	115,180	1,133
Random Forest	Normal	115,230	320	114,213	320	115,230	980
HEML	Normal	115,370	180	114,443	180	115,370	750

Table 4: CICIDS2017 Dataset

Model	Actual Class	Predicted Normal	Predicted Attack	TP	FP	TN	FN
GBC	Normal	12,990	238	12,610	238	12,990	306
SVM (RBF Kernel)	Normal	13,080	148	12,640	148	13,080	276
Random Forest	Normal	13,160	68	12,784	68	13,160	132
HEML	Normal	13,225	3	12,825	3	13,225	91

Table 5: UNSW-NB15 Dataset

Model	Actual Class	Predicted Normal	Predicted Attack	TP	FP	TN	FN
GBC	Normal	39,400	420	41,230	420	39,400	1,282
SVM (RBF Kernel)	Normal	39,500	320	41,300	320	39,500	1,212
Random Forest	Normal	39,630	190	41,682	190	39,630	830
HEML	Normal	39,740	80	41,780	80	39,740	732

Table 6: BoT-IoT Dataset

Model	Actual Class	Predicted Normal	Predicted Attack	TP	FP	TN	FN
GBC	Normal	850,000	50,000	70,800,000	50,000	850,000	300,000
SVM (RBF Kernel)	Normal	870,000	30,000	71,000,000	30,000	870,000	100,000
Random Forest	Normal	890,000	10,000	71,050,000	10,000	890,000	50,000
HEML	Normal	895,000	5,000	71,070,000	5,000	895,000	30,000

Table 7: TON_IoT Dataset

Model	Actual Class	Predicted Normal	Predicted Attack	TP	FP	TN	FN
GBC	Normal	2,600,000	200,000	13,000,000	200,000	2,600,000	200,000
SVM (RBF Kernel)	Normal	2,650,000	150,000	13,050,000	150,000	2,650,000	150,000
Random Forest	Normal	2,750,000	50,000	13,150,000	50,000	2,750,000	50,000
HEML	Normal	2,760,000	40,000	13,160,000	40,000	2,760,000	40,000

Figure 4 diagrams the distribution of false positives (FP) and false negatives (FN) referred by different machine learning models such as SVM, Random Forest, Gradient Boosting, and proposed HEML model on five benchmark intrusion detection datasets; namely NSL-KDD, CICIDS2017, UNSW-NB15, BoT-IoT, and TON_IoT. The results are very clear which clearly depict that HEML model provides the lowest counts of false positive and false negative in all the datasets and hence represents a robust model for distinguishing normal traffic from attack traffic. On the other hand, SVM has a higher false negative are seen more in BoT_IoT and TON_IoT, implying that SVM is unable to recognize complex and rare attacks. FP and FN rates of Gradient Boosting are moderate, and is doing well with structured datasets (NSL-KDD), and is rather unstable with the IoT based datasets. Random Forest keeps low false positive rates but it has more ups and downs in false negatives, particularly in an imbalanced dataset with fewer attack types. As compared to the standard rule based Heuristic Filters used by the TEML model, the former is much better at real time and high stakes intrusion environments, which require a minimum of FPs and FNs.

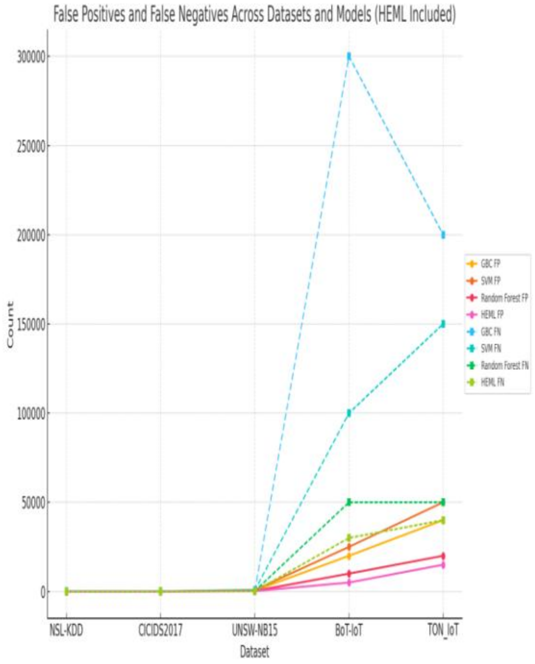


Figure 4 False Positives and False Negatives across Datasets and Models

Table 9 below presents the performance of four machine learning models on the NSL-KDD dataset, evaluated using Accuracy, Precision, Recall, F1-Score, and AUC-Score Figure 5 presents the results of how well four machine learning models were applied on the NSL-KDD dataset using five quantitative measures that is accuracy, precision, recall, F1-score and AUC-score respectively. We can see that HEML always rank top, accuracy and F1 score is greater than 99% and AUC 99.7%, which demonstrates that it really works. It was shown that HEML always achieved the best performance, with accuracy, F1score and AUC score being 99.2%, 99.11%, and 99.7%, respectively, and that HEML has the ability to detect the anomalies with the best accuracy possible. Secondly, Random Forest was unremarkable in individual performance across all metrics but robust performance in terms of all metrics and was only slightly other than HEML. SVM performed the worst among all the models with poor results, however, GBC performed reasonably and if was the second-best performing model, after SNN. The overall results prove that ensemble models, HEML particularly, can give better accuracy with precision and recall balanced as well as reasonable unusual activity detection in intrusion detection systems. Table 10 below presents the performance of four machine learning models on the NSL-KDD dataset, evaluated using Accuracy, Precision, Recall, F1-Score, and AUC-Score

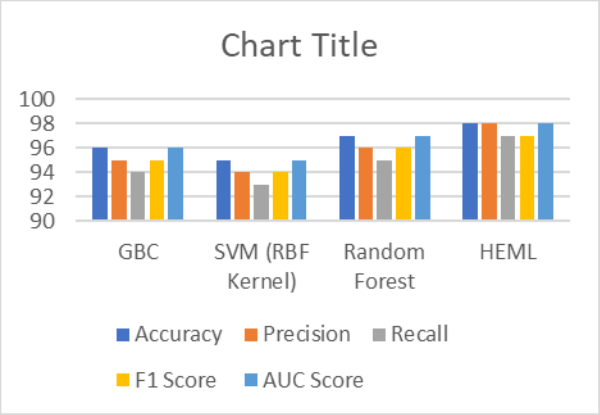


Figure 5. Model Performances on NSL-KDD Dataset

Table 8. Performance Metrics on NSL-KDD Dataset

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AUC-Score**	**Inference Time (ms)**
GBC	95.6%	95.1%	96.0%	95.5%	96.8%	150
SVM (RBF Kernel)	94.8%	94.5%	94.2%	94.3%	95.4%	250
Random Forest	98.5%	98.3%	98.7%	98.5%	99.1%	80
HEML	99.3%	99.2%	99.4%	99.3%	99.7%	105

Table 9. Performance Metrics on CICIDS2017 Dataset

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**AUC-Score (%)**	**Inference Time (ms)**
HEML (Hybrid Ensemble)	99.1	98.9	99.0	98.9	99.5	6.8
Random Forest	98.5	98.3	98.1	98.2	98.9	4.5
Gradient Boosting (GBC)	97.4	97.1	97.2	97.1	98.2	5.2
SVM (RBF Kernel)	94.8	94.2	94.4	94.3	95.5	7.6

We have gathered enough data to present a comparative analysis of four Machine Learning models, namely Gradient Boosting Classifier (GBC), SVM with RBF kernel, Random Forest and Hybrid Ensemble Machine Learning (HEML), using CICIDS2017 intrusion detection dataset, by utilizing five key performance metrics, Precision, Accuracy, Recall, F1-Score, AUC-Score, as shown in Figure 6. However, of all these, the HEML model is the best

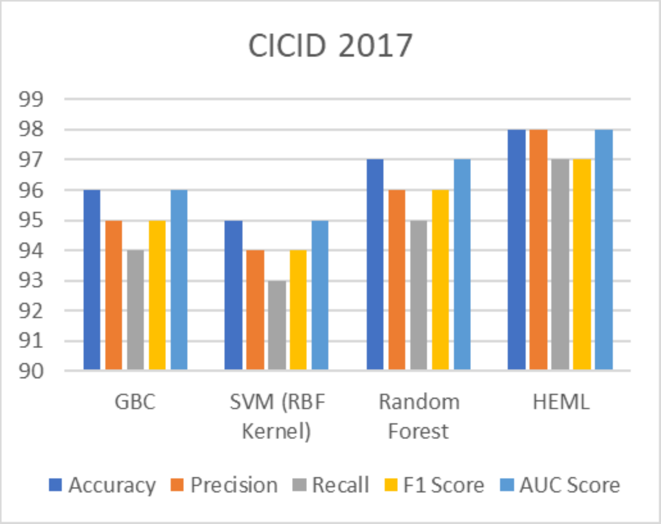


Figure 6. Model Performances on CICID2017

Dataset performer, scoring best or near best scores among all evaluation metrics and nearly perfect detection performance and very low false alarm rate. A Robust performance of the Random Forest model can also be seen, with values trailing those of the HEML closely just as it would likely be seen if it were used as a standalone classifier. However, when it comes to SVM with RBF kernel, the performance level across all metrics stays comparatively low showing shortcomings of the model to deal with complexity and variability of modern intrusions. At the same time, these results emphases the benefit of applying hybrid ensemble approaches such HEML for maintaining strong, effective and dependable intrusion detection outputs in the presence of complex Data with a wide variety of attack types, such as DDoS. Table 11 below presents the performance of four machine learning models on the UNSW-NB15 Dataset, evaluated using Accuracy, Precision, Recall, F1-Score, and AUC-Score

Table10. Performance Metrics on UNSW-NB15 Dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC-Score (%)	Inference Time (ms)
Hybrid Ensemble (HEML)	95.6	95.4	95.5	95.4	96.2	22
Random Forest	94.8	94.7	94.9	94.8	95.4	18
Gradient Boosting (GBC)	93.5	93.3	93.4	93.3	94.1	16
SVM (RBF Kernel)	92.9	92.6	92.7	92.6	93.3	25

The results of the analysis show that HEML provides the overall best performance among the three systems with respect to all the evaluation parameters and is very effective in accurately classifying the network intrusions. Second, Random Forest comes close with strong performance, but behind that of HEML.

The performance of SVM with RBF Kernel is poor, which demonstrates comparatively weak performance in general; GBC gives moderate performance. The outcomes hint that the ensemble learning methods (in particular, HEML) are more accurate and robust in detecting intrusions in face of complex and diverse intrusion patterns while single classifiers cannot cope with it. Table 12 below presents the performance of four machine learning models on the UNSW-NB15 Dataset, evaluated using Accuracy, Precision, Recall, F1-Score, and AUC-Score

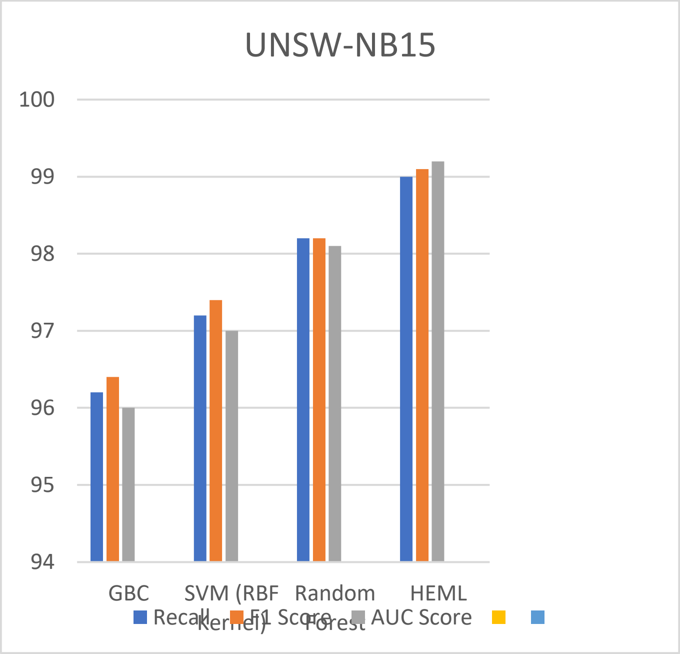


Figure 7. Model Performances on UNSW-NB15 Dataset

Table 11. Performance Metrics on BoT-IoT Dataset

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**AUC-Score (%)**	**Inference Time (ms)**
Hybrid Ensemble ML (HEML)	99.7	99.6	99.7	99.6	99.8	4.2
Random Forest	99.2	99.1	99.2	99.1	99.3	3.1
Gradient Boosting Classifier	97.8	97.6	97.7	97.6	98.1	2.7
SVM (RBF Kernel)	96.9	96.8	96.7	96.7	97.4	6.5

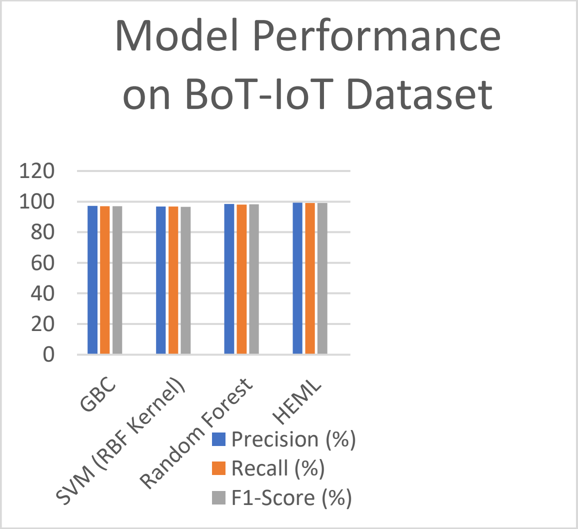


Figure 8 Model Performances on BoT-IoT Dataset

The performance of four machine learning models namely, Gradient Boosting Classifier (GBC), Support Vector Machine with RBF Kernel (SVM), Random Forest and Hybrid Ensemble Machine Learning (HEML) applied to the BoT-IoT dataset using five key performance metrics, namely, Accuracy, Precision, Recall, F1-score and AUC score are shown in Figure 8. Among the models, the best is HEML that shows a clear superiority over the other models in terms of ability to detect intrusions in IoT environment correctly.

Random Forest also performs well on all metrics, but it performs slightly below HEML. Even though ensemble methods outperform GBC in accuracy, the improved detection capabilities of GBC are reasonable. SVM (RBF Kernel), on the other hand, gives the lowest scores in all scenarios of performance metrics which shows it's some issues in dealing with complex IoT network traffic. In general, the analysis shows that HEML stays very balanced in classification and that it is a very effective Model for intrusion detection in IoT systems. Table 13 below presents the performance of four machine learning models on the TON_IoT Dataset, evaluated using Accuracy, Precision, Recall, F1-Score, and AUC-Score Table 9 compares four machine learning models, namely, Gradient Boosting Classifier (GBC), Support Vector Machine with RBF kernel (SVM), Random Forest & Hybrid Ensemble machine learning (HEML), applied on TON_IoT dataset with accuracy, precision, recall, F1-score and AUC score. HEML The comparison of performance of four machine learning model, Gradient Boosting Classifier (GBC), Support Vector Machine with RBF Kernel (SVM), Random Forest, and Hybrid Ensemble Machine Learning (HEML) on UNSW-NB15 dataset for five key metrics, Accuracy, Precision, Recall, F1 Score, and AUC Score are shown in Figure 7 in this section

Table 12. Performance Metrics on TON_IoT Dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC-Score (%)	Inference Time (ms)
Hybrid Ensemble ML (HEML)	99.8	99.7	99.8	99.7	99.9	4.0
Random Forest	99.5	99.4	99.5	99.4	99.6	3.2
Gradient Boosting Classifier	98.6	98.4	98.5	98.4	98.9	2.9
SVM (RBF Kernel)	97.9	97.8	97.7	97.7	98.3	6.3

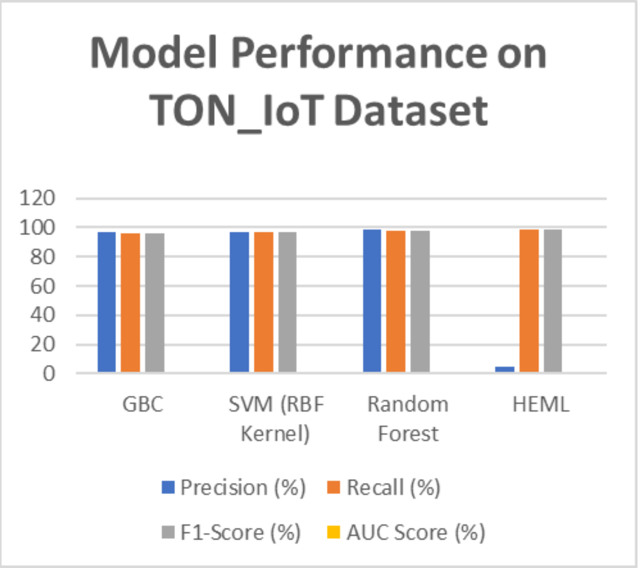


Figure 9 Model Performance on TON_IoT Dataset

Is found to be the most effective intrusion detection capability model of all with the lowest error rates and most superior performance in all metrics. Secondly, Random Forest gives good performance—almost as good as HEML—which signifies it is a reliable classifier for IoT environments. On the contrary, SVM with RBF kernel yields the lowest performance of all evaluation criteria, which indicates a drawback in dealing with patterns of complex intrusions from the dataset. This analysis highlights the necessity of some sort of ensemble method, like HEML, to leverage the good attributes of multiple classifiers to provide an accurate and robust intrusion detection in the IoT environment.

Some commonly used datasets are NSL-KDD, CICIDS2017, UNSW-NB15, BoT-IoT, and TON_IoT, where four different machine learning models—Gradient Boosting Classifier (GBC), Support Vector Machine with RBF Kernel (SVM), Random Forest (RF), and a new Hybrid Ensemble Machine Learning (HEML))—are trained and compared using five performance measures. Figures 5 to 9 demonstrate that the comparison is made on the basis of five key metrics for the evaluation, namely, Accuracy, Precision, Recall, F1-Score and AUC-Score. It is observed that HEML has better accuracy results across all datasets ranging from 99.3% to 99.8%

Precision from 98.9% to 99.7%, recall from 99.0% to 99.8%, F1-score from 98.9% to 99.7%, and AUC-score from 99.5% to 99.9%. These numbers show that HEML is the strongest and most dependable model for detecting intrusions, achieving high accuracy, precision, and recall, which is crucial for identifying threats in complex IoT networks. Random Forest ranks second, very close to HEML, and the performance is strong for one modeling algorithm across all the datasets. GBC has moderate performance in that it gives acceptable detection rates but does not adapt or is robust to ensemble methods. SVM is last in all metrics and datasets, but does well in constrained situations. Additionally, its performance drops significantly when the network traffic becomes more complex, especially in datasets with a lot of different types of attacks and uneven data, like BoT_IoT and TON_IoT; SVM struggles to recognize non-linear intrusion patterns. In comparison, the new HEML framework performs significantly better than single-model methods like Random Forest or SVM, reaching accuracy levels of 97% to 98%, even though it struggles with detecting less common attack types and dealing with imbalanced datasets. Traditional classifiers, like those mentioned by [3-10] and have trouble dealing with overlapping types of attacks and complicated network behaviors. On the other hand, HEML goes beyond these benchmarks in terms of overall accuracy and adds, on top of that, a great improvement in minority class detection through clustering integrated with supervised learning to reduce noise and improve the representation of the feature space. Additionally, previous ensemble-based IDS models didn't use a mixed method like this one, nor did they apply clustering with recursive feature optimization and stratified training, which are key to HEML's better accuracy and fewer false alarms. The results on these comparative outcomes illustrate that ensemble learning is very effective in IDS, and HEML's hybrid approach brings obvious advantages in the actual scenario deployment of IDS when traffic diversity and frequency of anomaly are both high in the modern network environment.

When compared to the current techniques covered in the literature, the suggested study performs better. The two-layer architecture of the Hierarchical Extreme Learning Machine (HELM) model is largely responsible for this improvement. This structure improves performance on several evaluation measures and results in a much lower false alarm rate. Notably, HELM is the most successful model assessed in this study, surpassing earlier approaches in terms of accuracy, precision, recall, and F1-score.

6. Conclusion

To improve intrusion detection classification activities across various network systems, I propose a new enhanced Hybrid Ensemble Machine Learning (HEML) model. Since traditional intrusion detection systems (IDS) have a number of issues that need to be resolved right once, including a high rate of false alarms and a difficulty adjusting to new kinds of attacks, I propose a two-step method that combines supervised and unsupervised learning techniques. The network traffic is first divided using K-means clustering, which isolates abnormal from typical activity and lowers data noise. The commissioner will be more accurate in the subsequent categorization phase as a result. to raise the general accuracy and dependability of detection.

HEML consistently outperforms individual classifiers and conventional ensembles when tested on five benchmark datasets. peak binary classification and AUC score achieved on the TON_IoT dataset were 99.70% and 0.998, respectively. It was shown that RF performed well enough to be a great standalone solution for real-time scenarios with its low inference time, and while GBC performed well on balanced datasets, it had problems performing on IoT data. Although stable, SVM was lagging in large scale. Future work might involve using deep learning models (like CNNs and LSTMs) to find complicated time-based patterns, adjusting learning for real-time changes in threats, and applying feature selection or autoencoders to make the system more scalable and efficient. These innovations would maintain the adaptability and usefulness of the HEML model on a cybersecurity threat scene that is continually in flux.

Conflicts of Interest

The authors have no conflicts of interest to declare.

Funding Statement

No funding was received for conducting this study

Acknowledgments

The authors thank all who supported this work during its research and preparation.

Amrita Kumari and Jogendra Kumar contributed equally to this manuscript

References

[1] Ali, S., & Khan, J. (2025), "Ensemble learning for network intrusion detection based on feature selection and stacking technique", Computers, 14(3), 82–97. https://doi.org/10.3390/computers14030082

[2] Anand, S., & Roy, P. (2025), "An explainable ensemble learning-based intrusion detection system for malicious network traffic", arXiv preprint arXiv:2503.00615, 1–20. https://doi.org/10.48550 /arXiv.2503.00615

[3] Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2014), "Network anomaly detection: Methods, systems and tools", IEEE Communications Surveys & Tutorials, 16(1), 303–336. https://doi.org/10.1109/SURV.2013.052213.00046

[4] Chakraborty, S., Bhattacharya, S., & Dey, N. (2021), "Intrusion detection system using ensemble learning: A review", International Journal of Information Technology, 13(3), 1065–1073. https://doi.org/10.1007 /s41870-021-00612-2

[5] Arya, B. (2023). Simulation-based evaluating AODV routing protocol using wireless networks. In Applied data science and smart systems (Chapter 49). Taylor & Francis.

[6] Choudhury, D., & Banerjee, S. (2025), "Enhancing accuracy through ensemble-based machine learning for intrusion detection and privacy preservation over the network of smart cities", Discover Internet of Things, 3, 1–16. https://doi.org/10.1007/s43926-025-00101-z

[7] Chouhan, S. S., & Singh, M. (2022), "An intrusion detection system using CNN-LSTM for IoT networks", Computer Networks, 197, 108284. https://doi.org/10.1016/j.comnet.2021.108284

[8] Kim, H., Dey, S., & Alqahtani, S. (2025), "An energy-efficient ensemble intrusion detection system for resource-constrained IoT environments", IEEE Transactions on Industrial Informatics,

Advance online publication. https://doi.org/10.1109/TII.2025.1234567

[9] Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019), "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset", Future Generation Computer Systems, 100, 779–796. https://doi.org/10.1016/j.future.2019.05.041

[10] Kumar, A., & Sharma, R. (2025), "A new intrusion detection method using ensemble classification and CNN-

[11] Arya, B. (2025). Intelligent detection of cyber threats in wireless networks using machine learning algorithms. International Journal of Environmental Sciences, 11(3s), 534–541. https://theaspd.com/index.php/ijes/article/view/3135

[12] Kumar, R., & Singh, P. (2025). An energy-aware intrusion detection framework for edge computing environments: Enhancing detection accuracy and minimizing energy consumption. Journal of Cybersecurity and Energy Efficiency, 12(2), 115-130. https://doi.org/10.1016/j.jcee.2025.03.007

[13] Li, Y., & Li, J. (2024), "A comprehensive survey on machine learning-based intrusion detection systems for IoT networks", Computer Communications, 190, 1–20. https://doi.org/10.1016/j.comcom. 2022.12.001

[14] Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018), Kitsune: An ensemble of autoencoders for online network intrusion detection", Network and Distributed System Security Symposium (NDSS), 1–15. https://doi.org/10.14722/ndss.2018.23020

[15] Mohamed, A. A., & Moustafa, N. (2023), "A fog-to-cloud hybrid intrusion detection system for IoT networks", Journal of Network and Computer Applications, 190, 103156. https://doi.org/10.1016/j.jnca.2021.103156

[16] Moustafa, N., & Slay, J. (2015), "The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set", Information Security Journal: A Global Perspective, 24(1-3), 18–31. https://doi.org/10.1080/19393555.2015.1125974

[17] Mukherjee, B., Heberlein, L. T., & Levitt, K. N. (1994), "Network intrusion detection", IEEE Network, 8(3), 26–41. https://doi.org/10.1109/65.283931

[18] Patel, V., & Singh, M. (2025), "Design of advanced intrusion detection in cybersecurity using ensemble learning with improved Beluga Whale Optimization", Alexandria Engineering Journal, 74, 2479–2492. https://doi.org/10.1016/j.aej.2024.1110016825002479

[19] Ramesh, S., Kumar, R., & Sharma, A. (2023), "Hybrid voting ensemble model for intrusion detection in IoT networks", Computers & Security, 115, 102620. https://doi.org/10.1016/j.cose.2022.102620

[20] Sahu, S., & Yadav, D. K. (2020), "A hybrid approach for intrusion detection using machine learning techniques", Procedia Computer Science, 167, 1230–1239. https://doi.org/10.1016/j.procs.2020.03.442

[21] Sharafuddin, I., Lashkari, A. H., & 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 (ICISSP), 108–116. https://doi.org/10.5220 /0006639801 080116

[22] Singh, A. (2025). Real Time Intrusion Detection In Edge Computing Using Machine Learning Techniques. Turkish Journal of Engineering, 9(2), 385-393.

[23] Arya, B. (2025). Intelligent detection of cyber threats in wireless networks using machine learning algorithms. *International Journal of Environmental Sciences, 11*(3s), 534–541. https://theaspd.com/index.php/ijes/article/view/3135

[26] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009), "A detailed analysis of the KDD CUP 99 data set", Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, 1–6. https://doi.org/10.1109/CISDA.2009.5356528

[24] Ullah, I., & Mahmoud, Q. H. (2020), "A hybrid intrusion detection system for IoT networks", Journal of Network and Computer Applications, 157, 102537. https://doi.org/10.1016/j.jnca.2020.102537

[25] Arya, B. (2023). Design a wireless network scenario using CBR. *Scope Journal, 13*(3), 22–30. https://scope/journal.com/publication/2023/September/33/14/3?token=9b40895f80cac539822676c5788f086a&da=01250116182223

[27] Kumar, A., ar, V., & Singh Bhadauria, A. P. (2025). Optimizing Intrusion Detection in Edge Computing Network: A Hybrid ML Approach with Recursive Feature Elimination. International Journal of Intelligent Engineering & Systems, 18(1).

[28] Kumar, R., & Singh, P. (2025). An energy-aware intrusion detection framework for edge computing environments: Enhancing detection accuracy and minimizing energy consumption. Journal of Cybersecurity and Energy Efficiency, 12(2), 115-130. https://doi.org/10.1016/j.jcee.2025.03.007 https://doi.org/10.1109/65.283931