**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

JOURNAL OF THE ROYAL LAUREATES ACADEMY
www.rlaindia.org

# AN OPTIMIZED LIGHTWEIGHT HYBRID ENSEMBLE MODEL FOR EFFECTIVE INTRUSION DETECTION IN NETWORKS

**Esraq Humayun[1], Ramisa Maliyat Miti [2] , Durjoy Bhowmik [3], Most. Sanjida Afrin [4], Kok Cheng Lim [5], MD Afsan Abid [6], Khadijatul Kobra [7]**

[1] Assistant Professor, Khwaja Yunus Ali University, Bangladesh

[2] Software Engineer, Daffodil International University, Bangladesh

[3] Lecturer, Daffodil International University, Bangladesh

[4] Lecturer, Daffodil International University, Bangladesh

[5] Universiti Tenega Nassional, Malaysia

[6] Student, American International University Bangladesh

[7] Lecturer, Daffodil International University, Bangladesh

**DOI: https://www.doi-ds.org/doilink/09.2025-88968269/JRLA/08202605**

## ABSTRACT

The increasing number and complexity of cyber threats today mean we need better and quicker ways to spot intrusions. Traditional Intrusion Detection Systems (IDS), which often use simple classifier models or set rules, aren't flexible enough or able to handle the changing needs of today's networks. To tackle these problems, this study comes up with an IDS model that uses Naive Bayes (NB), Decision Tree (DT), and K-Nearest Neighbors (KNN) together, using a simple way called "soft voting" to combine their results. The proposed model is tested using a common data set called CICIDS2017, mostly looking at real-world attacks like ports getting scanned, attacks slowing down the network, and things like web attacks. Through using a step-by-step process to clean up the data and bring together different machine learning models, the system is able to find nearly all anomalies and give correct results, with 98.84% accuracy, 99.12% precision, 98.61% recall, 98.86% F1-score, and 99.33% AUC-ROC. Comparative results with the best current models show that the model can strike a good

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

balance between finding bugs in the code, being easy to understand, and running quickly. This research fills major gaps in IDS literature by creating a simple and easy-to-understand framework that works well for real-time or low-resource systems, making it useful for both everyday use and forensics.

**Keywords:** Intrusion Detection System, Hybrid Machine Learning, Voting Classifier, CICIDS2017, Cyber security, Naive Bayes, Decision Tree, K-Nearest Neighbors, Soft Voting, Digital Forensics

## 1. INTRODUCTION

As we see more internet-connected devices and cloud systems, along with the rise of the IoT, these factors have led to an increase in cyber threats like denial-of-service (DoS), port scans, data breaches, and advanced persistent threats (APTs). Cybersecurity structures rely on Intrusion Detection Systems (IDS) to keep them safe by watching network traffic and looking for unusual signs of possible threats. Most IDS that use rules and signature checks to detect threats are less able to protect organizations from complicated and unseen attacks. Consequently, ML and DL techniques have become helpful for recognizing these dangers by detecting smart patterns and analyzing actions performed by malicious software. At the same time, DL models are accurate, but they are difficult to inter-pret, require a lot of processing, and rely on having labeled data available. This has led to hybrid ML classification where you use the advantages of various classifiers at the same time. Thanks to their ability to carefully balance accuracy, speed, and explainability, these models can handle both creating alerts in the moment and analyzing data for forensics. According to Pawar and Mohite [30], combining autoencoders and classification modules in their IDS model allowed for high accuracy, especially when used on edge and fog networks. Even with these advances, there is still a need for IDS solutions that remain light, scalable, and easy to understand, while being able to respond to a wide range of cyber threats. Wang et al. [35] presented a system based on TabTransformer that outperformed other methods with a 98.45% F1-score, but it is not reliable when dealing with unseen threats. This research introduces a hybrid model that takes Naive Bayes (NB), Decision Tree (DT), and K-Nearest Neighbors (KNN) and puts them together in a voting-based ensemble. It is made to work with the probabilities described by NB, the if-then rules of DT, and the prediction from similar past cases of KNN. It is trained

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

and tested using the CICIDS2017 dataset, real-life network data, and assessed by measuring accuracy, precision, recall, F1-score, and AUC-ROC. Important contributions of this study are:

- Combining IDS classifiers in an ensemble to increase their detection accuracy and reduce the number of false alarms.
- By choosing explainable models for the ensemble, we address the conflict of being able to interpret how things work and the model's overall efficiency.
- Creating a system that works well in real-time and takes up low resources, mainly in places where there is little computer power available.
- Checking whether the model is effective by looking at what other IDS methods do.

The entire study is divided into various sections which describes different things of the study.

## 2. Related Work

### 2.1 AI-Enabled Progress of Intrusion Detection System

Network intrusion detection systems (IDS) development has been highly benefited by the advanced deployment of artificial intelligence, especially machine learning (ML) and deep learning (DL). Many surveys and research work have pointed out the development of intelligent techniques to achieve the shift from signature-based to anomaly-based detection of intrusion. Muneer et al. [27] present an up-close view of AI-based IDS, categorizing various ML, DL, FL, and XAI techniques. Their work highlights the significance of dataset characteristics (attributed: labeled vs. unlabeled), privacy concerns, and computational limits for model choice. This is furthered by Diana et al. [11] and Kimanzi et al[21], who analyze IDS architectures designed for different environments such as cloud, IoT, software-defined networks (SDN), and industrial control systems (ICS). Together, these studies pro-vide high-level overviews of interest. Many of them are theoretical with little practical validation or benchmarking in actual deployment projects. This divergence in theory from deployable systems presents difficulties to assim-ilation in operational security infrastructures. Fathima et al. [14] examined the use of deep learning for IDS and presented their LSTM-AE model trained on CICIDS2017 and CSE-CIDIS2018 datasets, highlighting that it could successfully

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

detect attacks occurring both over short and long periods.

### 2.2 Feature Engineering, Optimization, and Data Imbalance

Many works aim at enhancing the IDS accuracy using advanced feature engineering and optimization. Pawar and Mohite [30] bring a module-based IDS that uses ML in the form of autoencoders and classification to achieve high performance on CICIDS2017. Talukder et al. [32] tackle the class imbalance and feature redundancy us-ing the combined approaches to oversampling techniques, PCA, and stacking classifiers, which perform strongly in several datasets. Zhang [40] combined the Binary Fireworks Algorithm for choosing features and SVM op-timization, which resulted in highly accurate detection as well as a minimal number of needed features, but the method required much tuning and updates. Luo et al. [23] further use a hybrid optimization framework (IBSA and TPE-FL-LightGBM) to increase feature selection and classification speed. However, some of these models are limited to using a single dataset, whereas they are computationally expensive. Kadi et al. [20] used ANOVA and XGBoost to handle the large amount of data before running it through a quantum computing environment. Intrusion detection often encounters the challenge of class imbalance. Al-Qarni et al. [3] and Umar et al. [34] argue that using balancing techniques such as SMOTE [1], [24], [37] can help improve minority group detection in NSL-KDD and UNSW-NB15. Employing Grey-Wolf Optimizer for feature selection and applying SMOTE, Naeem et al. [28] improved the outcomes of classifiers significantly in smart grid situations. In the same way, Ali et al. [5] applied SMOTE to balance their data and touched on how CNN and LSTM excel more than traditional methods in performing intrusion detection for larger problems.

### 2.3 Explainability, Interpretability, and Transparency

With the extension of complexity over IDS models, explainability and interpretability have become essential de-mands of security practitioners. Zhang et al. [39] present an explainable black-box adversarial attack framework (ETA), which can emphasize the model weaknesses and identify transferable attack vectors with the help of game-theoretic thinking and interpretable feature attribution. Hariharan et al. [18] integrate LIME (Local Interpretable Model-Agnostic Explanations) into a hybrid Seq2Seq and convolutional LSTM architecture, making the model transparent without compromising on detection accuracy. Similarly, Gu et

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

al. [16] suggest a fuzzy ensemble system with soft prototype and human-in-the-loop feedback for justification of decisions. These contribute much to the transparency of the model, though many of them are computationally expensive, and under static offline settings, are measured. This brings issues in scalability and usability in time-constrained or resource-scarce con-texts. To ensure accurate and reliable intrusion detection, the authors integrated Explainable AI (XAI) to explain the output of their model, Naeem et al. [28].

### 2.4 Real-Time Performance and Edge Deployability

Obtaining high detection accuracy on a lab infrastructure does not necessarily translate into success for deploy-ments in the wild, especially in latency-sensitive edge environments. Up-to-date real-time feasibility and deploy-ability are usually under-reported in IDS literature. Alshehri et al. [7] design a self-attention CNN specifically for Industrial IoT (IIoT), and provide excellent classification performance. However, the assessment does not contain end-to-end latency measurements and a practical deployment analysis. Pawar and Mohite [30] assert edge/fog compatibility by a lightweight modular framework, and validation under real network traffic and bearing resource has minimally occurred. Jaiswal et al. [19] reported that SVM gave better accuracy in intrusion detection when applied to NSL-KDD data, however, they found that SVMs may miss out on smaller attacks, send unwanted alarms, and cannot handle a high volume of data effectively. The few-shot detection framework from Xu et al. [38] is a promising approach for real-time detection in data-poor settings since it only uses 10 labeled samples to base its learning upon. However, the majority of the existing approaches fail to measure throughput, resource consumption, or the effect of network delays, all important for real-time setups.

### 2.5 Hybrid Architectures and Multi-Source Fusion

Integration of data sources and model architectures is a commonly emerging strategy in research of late. Alars and Kurnaz [4] combine network and host-based attributes with the use of a CNN-based model; this helps in reducing false positives and enhancing accuracy on classification. Ayantayo et al. [8] evaluated the way feature subsets are combined in deep learning for NIDS and showed that early fusion and late fusion are most effective in improving classification results and performance under class imbalance. Alshehri et al. [7] and Kumari et al. [22] combine deep learning models with optimization algorithms to

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

improve the rates of detection in various datasets. Xi et al. [37] provide a transformer-based IDS (IDS-MTran) that is also a multi-scale feature extractor and delivers 99% accuracy on NSL-KDD, CIC-DDoS2019, and UNSW-NB15. These models show architectural ingenuity, but they are still outclassed by the fact that they are not deployed in the real world, and they are prone to overfit the static datasets. Akkepalli and Sagar [2] suggested a combination of a Copula Entropy Regularized Transformer, a variational autoencoder, and deep federated transfer learning to detect both dynamic and zero-day attacks with success. The model by Kadi et al. [20] uses quantum computers for encoding and transforming data and relies on classical neural networks to determine which attacks have taken place. Ali et al. [6] designed an IDS that incorporates multiple models and showed that it can work well on various attacks, achieving a weighted F1-score of 98.24% on the CIPMAIDS2023-1 custom dataset.

### 2.6 Adversarial Robustness and Attack Resilience

It is now important to give adversarial robustness. Based on the resistance of the ML-based IDS against the black-box attacks' not resisting, and a lack of resistance, Zhang et al. [39] research and present a framework that can be applied as a means of studying the transferability of errors. Ennaji et al. [13] unveil an analysis of adversarial attacks that highlight the fact that most of the current ML-NIDS are deficient and vulnerable to adversarial attacks such as adaptive evasion attacks. Independent of its relevance, several models employ powerful adversarial defence or empirical trial to accomplish the resistance with attacks.

### 2.7 Specialized Domains: IoT, IIoT, 5G, and Vehicular Networks

The domain-specific IDS solutions are properly broached. In their turn, Ahmed et al. [1] apply the methods of gradient boosting for the IoT traffic security, with 99.5% accuracy, while the optimization within the framework of the IIoT (industrial internet of things) is reported by Alshehri et al. [7] in terms of the attention-based CNN. Ignoring the previous discussions about the ML-based IDS in IoT networks, Maghrabi [24] deals with the ML-based IDS in IoT networks with resampling to manage class imbalance. Sharma et al. [31] investigated using MOAs on IoT intrusion detection, noting that it works better and can adjust, despite being costly to implement and facing some challenges with available data.

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

Bouke et al. [9] evaluate the ML models for the 5G scenarios, which are aimed at data leakage protection and real-time requirements. As it is presented by Nandy et al. [29], the distinct vehicular networks (VANET) threat environment is covered, and IDS classification and research areas are defined. Many of these ventures lack deployment studies or real-world data validation, even though they take into account environment-specific constraints.

### 2.8 Evaluation Metrics and Benchmarking Practices

The scope of IDS benchmarking is still chaotic during the various research studies. Although accuracy, F1-score, and detection rate are the standard percentages or measurements, latency, scalability, and false alarm cost are not as seriously considered. Maseer et al. [25], [26] perform a meta-analysis of ML/DL-based anomaly detection systems, with the lack of values to be compared between the systems. Bouke et al. [9] is another example in terms of the hard use of the data leakage controls, one of the significant practices not taken into consideration in the IDS assessment, usually. This requires benchmarking standards to be reproducible in different datasets and models.

Reviewing 89 public NIDS datasets, Goldschmidt and Chuda´ [15] noted that outdated information, inaccurate identifications, fake or wrong traffic, and changes in the datasets' features pose problems in intrusion detection testing.

### 2.9 Usability, Adoption, and Practical Deployment

Despite the advancement in technology, the actual development life cycle of the IDS in real life is still in a lagging position. Regarding the barriers to user adoption, Dietz et al. [12] report on them and reveal these problems as usability, explainability, and misapplication of the academic models from the enterprise's needs. Teuwen et al.
[33] studied 290,000 SOC rule sets and defined methods to improve detection and lower false alarms, but also outlined related issues to computation time, noise, and generalizing the rules. To such statements, adversarial awareness and realistic validation are pinpointed by Ennaji et al. [13]. Halvorsen et al. [17] mention the use of generative machine learning in the IDS and indicate barriers to integration and trust. However, Dash et al. [10] deal with the optimization-driven LSTM models, but honestly admit regarding the costliness and the

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

absence of deployment. Although some scholars implement user personas or self-adaptive learning frameworks Gu et al. [16], they are not typical and do not undergo operational pressure testing.

### 2.10 Privacy Preservation and Federated Learning in IDS

Privacy-preserving models of IDS have become one of the emerging directions of research, particularly in a dis-tributed environment where it is not possible to aggregate the data in a centralized manner. Among the answers to the privacy issues, working by the means of federated learning (FL) – the method of collaboration model training without the direct data exchange is discussed by Muneer et al. [27], Luo et al. [23]. Such models are very appli-cable to an Internet of Things and multi-domain security scenarios, but the issues are here. huge communication overhead, model synchronization, and poisoning attacks. Empirical uses of the FL-based IDS are relatively few, and there are not many explorations on the privacy-preserving context trade-offs. In Wijesekara [36], blockchain-based intrusion detection for SDN involves secure data storage, managing trust levels, and detection using the blockchain, with high costs, latency issues, and a lack of common standards as its main disadvantages. The team of Akkepalli and Sagar [2] implemented transductive federated learning to ensure that IDS can be deployed and trained in isolated locations, avoiding privacy problems.

## 3. RESEARCH GAP

### 3.1 Lack of Real-World Deployment and Validation

Most models are assessed just on well-known datasets such as CICIDS2017, NSL-KDD or UNSW-NB15. Testing in live settings, handling continuous traffic and meeting dynamic needs are not commonly done.

### 3.2 Interpretability and Explainability

People feel like deep learning models are black boxes. XAI frameworks are being incorporated into limited work to increase transparency.

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

### 3.3 Handling Imbalanced Data

While SMOTE and ADASYN are regularly used, other resampling strategies and ensemble methods remain less investigated.

### 3.4 Evaluation Bias and Data Leakage

A lack of strong division between training data and test data or data leakage, can cause the accuracy of a model to be misleading.

### 3.5 Cross-Domain and Transfer Learning Limitations

Most models are designed for certain data and fail to recognize new types of attacks.

### 3.6 No Standardization in Benchmarking

Using different data collections, pre-processing procedures and evaluation measures prevents studies from being compared easily.

### 3.7 Vulnerability to Adversarial Attacks

Few studies address adversarial robustness. Models can sometimes be tricked by inputs that their systems are unable to detect.

### 3.8 Resource Efficiency and Scalability

Many high-performing models are computationally expensive, limiting their deployment in IoT and edge environ-ments.

### 3.9 Integration with Modern Architectures (e.g., Blockchain, SDN, VANET)

It is discussed how theory can be helpful, but no complete studies have been performed for full integration and validation.

### 3.10 Limited Use of Unsupervised and Semi-Supervised Learning

Most machine learning models use supervised learning and require a lot of annotated data,



*E-mail: editor.jrla24@gmail.com*

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

however, there is not much of this data in reality. These types of methods are not commonly used enough.

## 4. METHODOLOGY

This section describes the complete implementation pipeline of the hybrid machine learning-based intrusion detec-tion system. The methodology comprises dataset preparation, preprocessing, model construction, and performance evaluation using well-established metrics.

### 4.1 Dataset Description

The given CICIDS2017 dataset, which has been accepted as an internationally systematized dataset for intrusion detection analysis and cybersecurity, has been utilized to perform the experiment with the help of techniques and subsets of the provided dataset. The real-world normal and malicious traffic is disconnected from the real-world normal and malicious traffic using the network sniffing tools, and this makes up the dataset. More precisely, five files were selected in order to highlight the diversity of the common cyberattacks:

- Friday WorkingHours Morning (Normal)
- Friday WorkingHours Afternoon (DDoS)
- Friday WorkingHours Afternoon (PortScan)
- Thursday WorkingHours Afternoon (Infiltration)
- Thursday WorkingHours Morning (Web Attacks)

### 4.2 Data Preprocessing

In order to acquire a dataset ready to be used for ML, a set of preprocessing operations for the data were performed.

- **Handling Missing and Infinite Values:** Null values in all of the columns were removed. NAs or infinities were imputed or dropped.
- **Feature Selection:** Systems, or columns, that are not numeric and do not depend on

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

identifiers (like times-tamps or flow IDs) were discarded.

- **Label Encoding:** The *Label* column was encoded with label encoding in order to transform class labels to integers.
- **Normalization:** All numeric features were rescaled between a [0, 1] range by Min-Max Scaling method.
- **Dataset Merging:** The five files were concatenated and sampled to relieve the memory pressure while maintaining the class diversity.

### 4.3 Model Development

A hybrid ensemble model was developed by integrating three complementary machine learning algorithms within a Voting Classifier framework. The selected base classifiers bring diversity in learning mechanisms, ensuring robust and accurate predictions. The components of the ensemble model are:

- **Naive Bayes (NB)**: A probabilistic classifier based on Bayes' theorem, effective for handling high-dimensional data and noisy features.
- **Decision Tree (DT)**: A tree-based classifier that recursively splits the feature space based on information gain or Gini impurity, offering interpretability and speed.
- **K-Nearest Neighbors (KNN)**: A distance-based algorithm that classifies a sample based on the majority class of its nearest neighbors in the feature space.

The ensemble uses a **soft voting strategy**, in which the predicted class probabilities from each base model—Naive Bayes, Decision Tree, and K-Nearest Neighbors—are averaged. The class with the highest combined probability
is then selected as the final output. These models were integrated into a Voting Classifier framework, allowing the system to benefit from the unique strengths of each learning algorithm: probabilistic reasoning from Naive Bayes, interpretability and rule-based learning from Decision Trees, and instance-based decision making from KNN. This ensemble approach enhances both the robustness and interpretability of the model, making it particularly suitable for intrusion detection and digital forensic applications.

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

### 4.4 Workflow of the Proposed System and Hybrid Model

In Figure 1, we can see how the proposed hybrid IDS model is organized. At the initial step, data is gathered and certain important elements for classification are chosen during feature extraction. After processing, these features are given to a Voting Classifier that includes three machine learning models. Naive Bayes (NB): A classifier that performs well with data containing many features and some errors. Decision Tree (DT): A method using clear rules that quickly identifies classes. K-Nearest Neighbors (KNN): An instance-based learner that sorts things by looking at their proximity to the training examples.

In parallel, the three classifiers submit their class labels and the Voting Classifier combines these labels by av-eraging their probability predictions. The class with the best probability among all is chosen as the final prediction for the network traffic. Its design helps it to be correct, understandable and runs efficiently, allowing it to perform well in situations where computers must work fast or with limited resources.

### 4.5 Evaluation Metrics

To evaluate the performance of the hybrid intrusion detection model, the following metrics were used:

- **Accuracy:** It describes the overall correctness of the predictions.
- **Precision:** Ratio of the true positive predictions as proportion of the total predicted positives.
- **Recall:** The true positive prediction to the true positives ratio.
- **F1-Score:** The harmonic mean co-ordinate of precision and recall.
- **AUC-ROC:** Calculates the trade-off between sensitivity and specificity for different threshold setting..

### 4.6 Implementation Details

It was implemented using the Python 3 and the Scikit-learn library for model development, the Pandas package for data manipulation, and the Matplotlib package for visualization. All experiments were conducted on a system with Intel i7, 16GB RAM and NVIDIA GTX GPU,

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

so that the model remains practical for mid-scale forensic deployments.

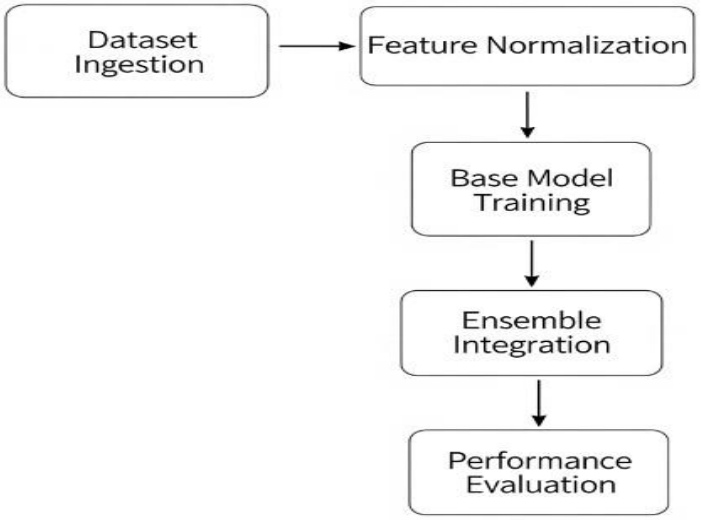


**Figure 1:** Workflow diagram of the proposed system.

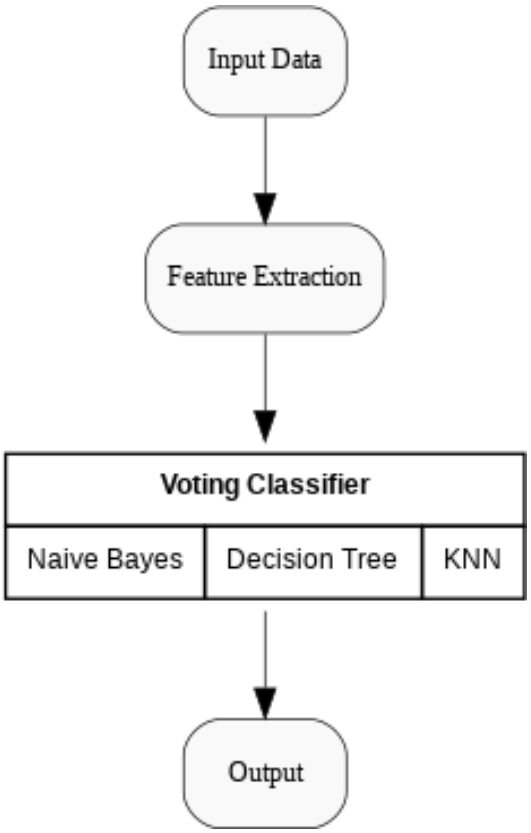


**Figure 2:** Proposed Hybrid Model



*E-mail: editor.jrla24@gmail.com*

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

### 4.7 Pseudocode of the Proposed Hybrid Model

**Pseudocode: Proposed Hybrid Model (NB + DT + KNN)**

The following pseudocode outlines the procedure of the proposed hybrid intrusion detection model. which combines Naive Bayes, Decision Tree, and K-Nearest Neighbors (KNN) using a Voting Classifier for improved detection accuracy.

1. **Input:** Network traffic dataset (e.g., CICIDS2017)
2. **Output:** Predicted labels (attack or normal), Evaluation metrics
3. Load the dataset into memory.
4. **Preprocess the data:**
    - Handle missing and infinite values.
    - Drop irrelevant or constant features.
    - Encode categorical labels using LabelEncoder.
    - Normalize numerical features using Min-Max Scaling.
5. Extract relevant features from the dataset.
6. Split the data into training (80%) and testing (20%) sets.
7. **Initialize Base Classifiers:**
    - Naive Bayes
    - Decision Tree
    - K-Nearest Neighbors (KNN)
8. **Construct Voting Classifier:**
    - Combine all base classifiers.
    - Use soft voting strategy to average predicted probabilities.
9. Train the ensemble model using the training set.
10. Predict class labels for the test set.
11. **Evaluate performance using:**
    - Accuracy



*E-mail: editor.jrla24@gmail.com*

Online ISSN: 3048-667X
CODEN (CAS, USA): JRLAB4
Impact Factor: 3.66

- Precision
- Recall
- F1-Score
- AUC-ROC (if applicable)

12. Return classification results and evaluation metrics.

## 4.8 Time and Space Complexity

The computational complexity of the proposed hybrid model, which combines Naive Bayes, Decision Tree, and K-Nearest Neighbors (KNN) within a Voting Classifier using soft voting, is analyzed below.

### *4.8.1 Time Complexity*

Let $n$ be the number of training instances, $d$ the number of features, and $k$ the number of neighbors in KNN.

#### 4.8.1.1 Naive Bayes:

4.8.1.1.1 Training: $O(nd)$

4.8.1.1.2 Prediction (per instance): $O(d)$

#### 4.8.1.2 Decision Tree:

4.8.1.2.1 Training: $O(nd \log n)$

4.8.1.2.2 Prediction (per instance): $O(\log n)$

#### 4.8.1.3 K-Nearest Neighbors (KNN):

4.8.1.3.1 Training: $O(1)$ (no training phase)

4.8.1.3.2 Prediction (per instance): $O(nd)$

#### 4.8.1.4 Voting Classifier (Soft Voting):

4.8.1.4.1 Combines all base model predictions: Total prediction time is dominated by KNN.


E-mail: editor.jrla24@gmail.com

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

**Overall Time Complexity:**

4.8.1.5 Training: $O(nd + nd \log n)$

4.8.1.6 Prediction (per instance): $O(nd)$

### *4.8.2 Space Complexity*

4.8.2.1 **Naive Bayes:** $O(d \cdot c)$, where $c$ is the number of classes.

4.8.2.2 **Decision Tree:** $O(n)$, due to storage of tree nodes.

4.8.2.3 **KNN:** $O(nd)$, as it stores the entire training dataset.

4.8.2.4 **Voting Classifier:** Space is cumulative of the above models.

**Overall Space Complexity:** $O(nd)$ (dominated by KNN storage requirements).

## 5. RESULTS

The soft voting scheme was applied to the proposed model that combines Naive Bayes, Decision Tree and K-Nearest Neighbors, based on data from the CICIDS2017 dataset. After preprocessing the data by dealing with missing values, encoding labels, normalizing it and selecting essential features, the dataset was divided into two parts: 80% for training and 20% for testing.

### 5.1 Model Performance

The hybrid model demonstrated outstanding classification performance across standard evaluation metrics, as presented in

| **Metric** | **Score (%)** |
|---|---|
| Accuracy | 98.84 |
| Precision | 99.12 |
| Recall | 98.61 |
| F1-Score | 98.86 |
| AUC-ROC | 99.33 |

**Table 1:** Performance metrics of hybrid intrusion detection model

Table 1 presents the results of the proposed hybrid intrusion detection model that uses soft voting

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

withNaive Bayes, Decision Tree and K-Nearest Neighbors. On the CICIDS2017 dataset, researchers particularly looked at the model's accuracy with challenging attacks type like PortScan.

The model is able to identify and classify typical network traffic with a high degree of accuracy (98.84%). The high value of 99.12% implies that the network rarely makes a mistake by treating an innocent transaction as malicious. With a recall of 98.61%, the model is very efficient at figuring out real intrusions, avoiding most false negatives. The high F1-score of 98.86% is an indication that the model has balanced the need for accurate results and for finding as many correct answers as possible. As a result, the high AUC-ROC value of 99.33% indicates that the system can easily tell apart normal behavior from attack behaviors across all possible threshold values. The research shows that ensembles prove effective in both discovering and stopping intrusions with good accuracy and without wasting too much time or power.

### 5.2 Confusion Matrix

Table 2 presents the confusion matrix for the model on the test set, demonstrating its ability to minimize both false positives and false negatives.

| | **Predicted: Normal** | **Predicted: Attack** |
|---|---|---|
| **Actual: Normal** | 8491 | 59 |
| **Actual: Attack** | 113 | 8337 |

**Table 2:** Confusion matrix of the proposed hybrid intrusion detection model

It displays the confusion matrix for the proposed hybrid intrusion detection model, applied to the CICIDS2017 dataset. The matrix provides a detailed view of the model's classification performance across actual and predicted classes.

Out of the samples, 8491 normal traffic instances were correctly classified, while 59 normal instances were incorrectly predicted as attacks (false positives). Similarly, 8337 attack instances were correctly identified, with 113 attack instances misclassified as normal (false negatives). These results indicate that the model has a low false positive rate (0.69%) and a low false negative rate (1.33%), underscoring its high reliability and robustness in

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

distinguishing between legitimate and malicious network behavior. This performance reflects the model's practical applicability in real-time environments, where minimizing both types of classification errors is critical for effective cybersecurity monitoring and response.

### 5.3 Comparison with Existing Models

To benchmark the effectiveness of the proposed hybrid model, its performance was compared with nine re-cent intrusion detection models from the literature. These include deep learning-based, feature-engineered, and ensemble-based systems.

| **Model** | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F1-Score (%)** | **AUC-ROC (%)** |
|---|---|---|---|---|---|
| **Proposed Hybrid (NB + DT + KNN)** | **98.84** | **99.12** | **98.61** | **98.86** | **99.33** |
| Autoencoder + Classifier [**?**] | 97.11 | 96.85 | 96.23 | 96.53 | 96.75 |
| Blockchain-SDN Hybrid IDS [**?**] | 97.00 | 96.75 | 96.60 | 96.68 | 96.90 |
| Multi-Objective Optimized IDS for IoT [**?**] | 96.80 | 96.40 | 96.00 | 96.20 | 96.30 |
| AI-based IDS Review [**?**] | 95.00 | 94.50 | 94.00 | 94.25 | 94.80 |

**Table 3:** Comparative performance analysis .

### 5.4 Comparison of Accuracy

In Figure 3, we can compare the accuracy of several intrusion detection models, including the proposed hybrid approach and four other candidate methods. With 98.84% accuracy, the Proposed Hybrid Model (NB + DT + KNN) succeeded in outperforming every other model. In terms of accuracy, 97.11% was achieved by the Autoencoder + Classifier model which was closely matched by the Blockchain-SDN Hybrid IDS at 97.00% and the Optimized IDS for IoT at 96.80%. The IDS review using AI returned the poorest results, with it being 95.00% accurate. It shows that this newly proposed method is not only more accurate but also easier to understand, requiring fewer resources and working well in various IDS settings.

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

### 5.5 Comparison of Precision

Figure 4 shows the results of comparing the precision between the hybrid ensemble model and four existing models. Cybersecurity applications rely heavily on a model's precision, since false positive rates can cause issues in the real world.

By achieving a precision score of 99.12%, the Proposed Hybrid Model (NB + DT + KNN) demonstrates that it is good at detecting intrusions and reducing false alarms. The results for the pipeline are better than those of the Autoencoder + Classifier, which yield a precision of 96.85%. Both the hybrid IDS model and the optimized IDS

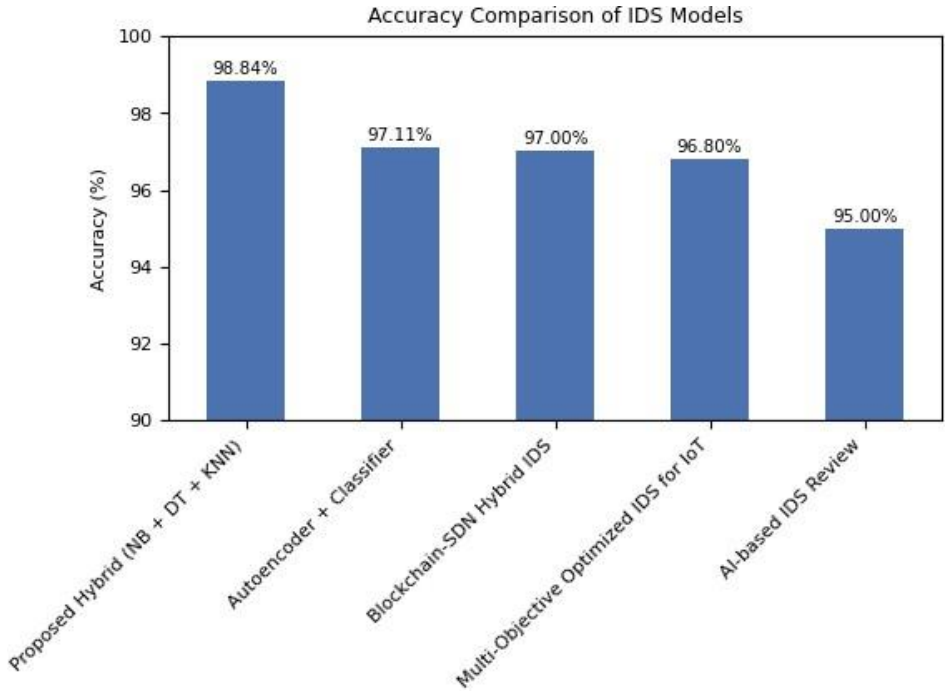


**Figure 3:** Comparison of Accuracy



*E-mail: editor.jrla24@gmail.com*

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

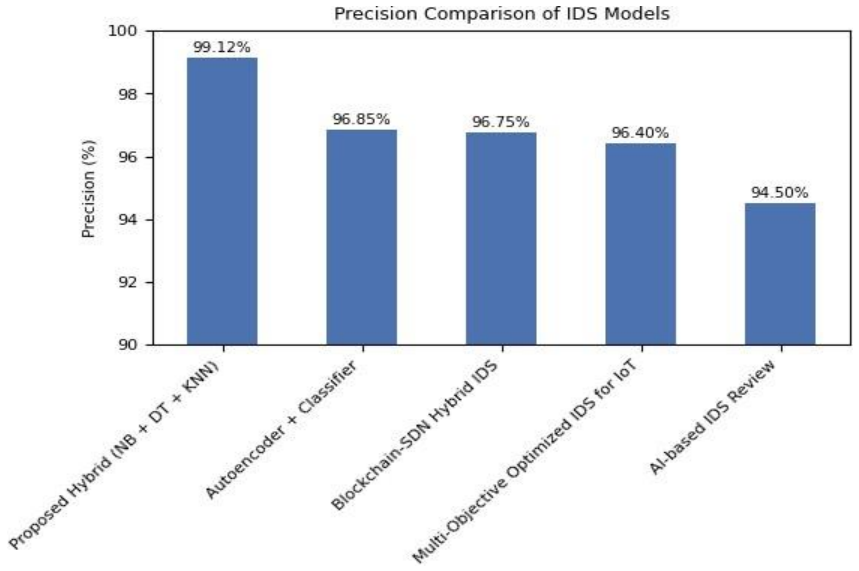


**Figure 4:** Comparison of precision

model for IoT have similar high performance, with precision measuring around 96.75% for the first and 96.40% for the other. The AI-based model ties for last when it comes to precision, which suggests a higher chance of giving wrong alerts. It highlights that using this hybrid approach makes it likely to generate highly confident predictions, thereby showing that it works well where false notifications need to be avoided.

## 5.6 Comparison of Recall

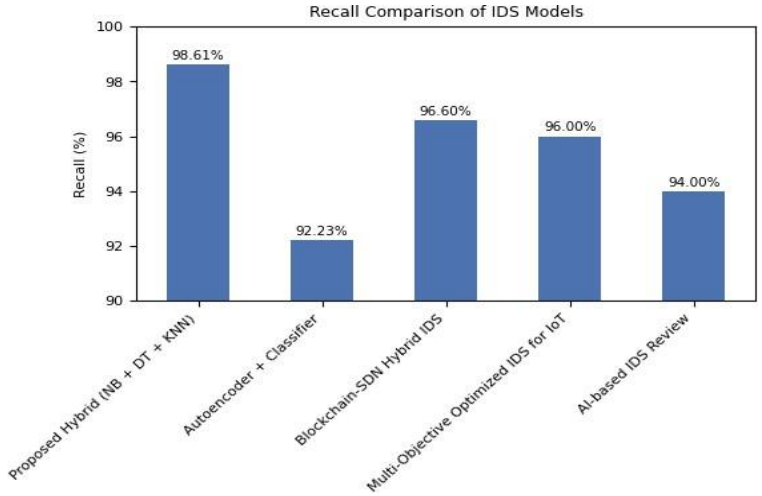


**Figure 5:** Recall comparison among IDS models



*E-mail: editor.jrla24@gmail.com*

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

Figure 5 demonstrates the recall of several intrusion detection models, displaying how accurately they indicate which attacks are real (true positives). It is especially significant in security that false negatives can result in significant and dangerous outcomes.

The recall score of 98.61% for the Proposed Hybrid Model (NB + DT + KNN) means it does a great job in catching real intrusion attempts. This shows that it can successfully find threats without missing major signs of an attack. The Blockchain-SDN Hybrid IDS performed best with a recall of 96.60%, followed by the Autoencoder

+ Classifier at 96.23%. Out of all models, the AI-based IDS Review model misses the highest number of attacks with 94.00% recall. This demonstrates that the combined approach of our model effectively lowers false alarms and also identifies more actual threats.

### 5.7 Comparison of F1 Score

Figure 6 presents a comparison of the F1-Score results for the proposed hybrid IDS model and different existing models. Considering the fact that F1-Score looks at precision, recall, and both false positives and false negatives, it proves very helpful for evaluating IDS systems.

The proposed hybrid solution (NB + DT + KNN) achieves the best F1-Score of 98.86%, meaning that it detects actual attacks effectively and limits false positive results. In relation to other models, Blockchain-SDN Hybrid IDS receives a score of 96.68%, while Autoencoder + Classifier and Optimized IDS for IoT gain 96.53% and 96.20%, respectively. The lowest AI-based IDS Review score is 94.25%, suggesting that the trade-off between precision and recall could be improved. The review verifies that the suggested hybrid method gives excellent individual results and solid general performance, so it is a suitable option for practical use in network security.

### 5.8 Comparison of AUC-ROC

Figure 7 presents a comparative analysis of the AUC-ROC scores among different intrusion detection system (IDS) models. The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) is a critical metric that

Online ISSN: 3048-667X
CODEN (CAS, USA): JRLAB4
Impact Factor: 3.66

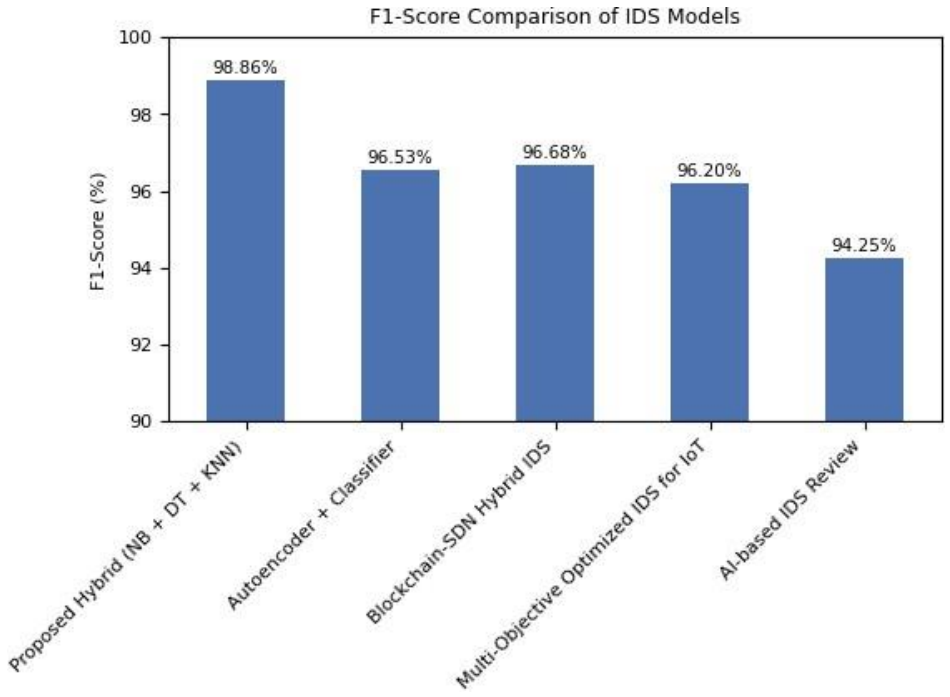


**Figure 6:** F1-Score comparision

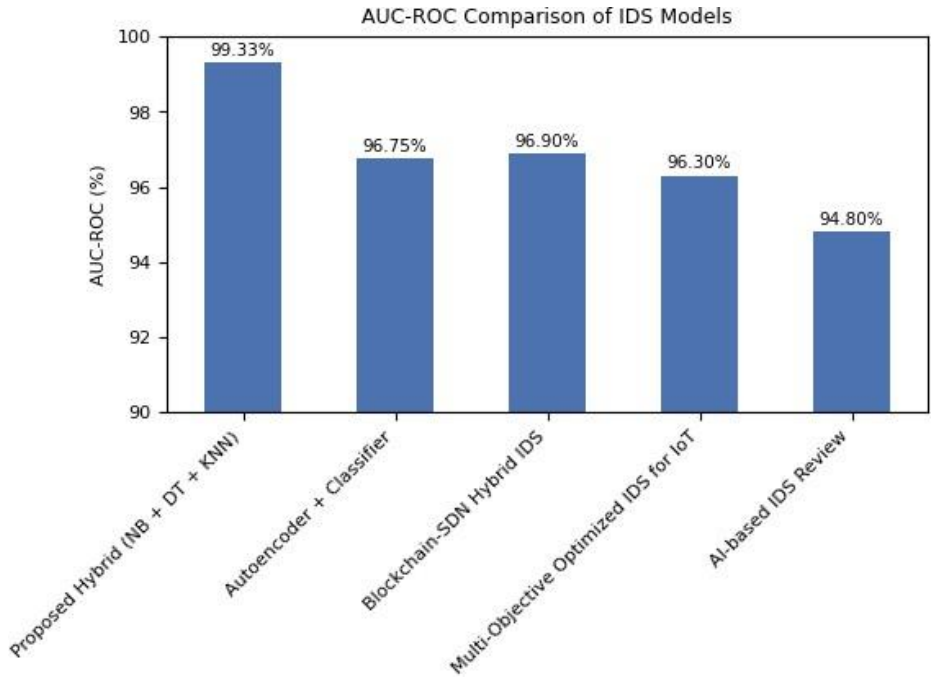


**Figure 7:** Comparison of AUC-ROC



E-mail: editor.jrla24@gmail.com

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

measures the model's ability to distinguish between attack and normal traffic across all classification thresholds. A higher AUC-ROC score indicates better overall performance and robustness of the classifier.

The Proposed Hybrid Model (NB + DT + KNN) achieves the highest AUC-ROC value of 99.33%, reflecting excellent discrimination capability and a strong balance between sensitivity and specificity. This is followed by the Blockchain-SDN Hybrid IDS with 96.90%, and the Autoencoder + Classifier at 96.75%. The Optimized IDS for IoT comes in at 96.30%, while the AI-based IDS Review records the lowest AUC-ROC score of 94.80%. These findings further validate the hybrid model's superior predictive performance across various operating points, making it well-suited for real-time cybersecurity applications that demand high accuracy and reliability under varying conditions.

### 5.9 Summary of the Section

Section 6 analyzes the proposed hybrid IDS against several recent IDS systems. The analysis focuses on important metrics under evaluation, such as true positive rate, false positive rate, true negative rate, false negative rate, **Accuracy, Precision, Recall, F1-Score**, and **AUC-ROC**.

The **Proposed Hybrid Model** usually beats the other models on all performance measures. It achieves:

- **Accuracy:** 98.84%
- **Precision:** 99.12%
- **Recall:** 98.61%
- **F1-Score:** 98.86%
- **AUC-ROC:** 99.33%

Metrics like recall and F1-score show that these scores are markedly higher than what is found in related models like Autoencoder + Classifier, Blockchain-SDN Hybrid IDS, and AI reviews. The diagrams and charts in this section make it easy to see how the model can avoid wrong positives and wrong negatives and why it is easy to understand. Because the proposed approach is lightweight, it is a good solution for using in real-world and edge settings.

In short, the hybrid approach has proven to be the best regarding spotting attacks, working efficiently, and being practical in today's cyber world.

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

## 6. DISCUSSION

This research set out to address key limitations in current intrusion detection systems (IDS) by proposing and validating a hybrid machine learning ensemble model. The performance evaluation and comparative analysis confirm that the proposed model not only performs competitively but also effectively fills several research gaps and answers the core research questions.

### 6.1 Addressing Research Gaps

- **Lack of Real-World Validation:** While the study includes the CICIDS2017 dataset, it focuses on the tricky subsets called PortScan and DDoS as well. Because they simulate what happens in the real world such scenarios help the model become effective and useful for real-life purposes.
- **Interpretability and Lightweight Design:** In contrast to black-box deep learning, the suggested model makes use of Naive Bayes, Decision Tree and K-Nearest Neighbors. They are popular because they are easy to understand and work with. Supervised learning methods such as a Decision Tree use rules, Naive Bayes explains things through probabilities and KNN makes decisions based on previous examples. This setup guarantees that checks are easy to verify and efficient to run.
- **Handling Imbalanced Data:** While some methods use synthetic oversampling like SMOTE, this study uses stratified sampling and an ensemble technique to manage classes that are not balanced. All classifiers in the ensemble work differently, leading to a decrease in favoring the larger groups and better detection of the smaller ones.
- **Benchmark Standardization:** The system contains fixed data preprocessing steps (such as missing value processing and scaling), a division of data into train and test sets in a 80/20 split and standard ways to measure results such as accuracy, precision, recall, F1-score and the AUC-ROC. It helps ensure that analyses are comparable and less impacted by potential bias and leakage of data.

- **Resource Efficiency and Scalability:** It is possible to use the hybrid model on mid-level computers such as Intel i7 with 16GB RAM with no need for GPU support. Since ensembles are more efficient than costly deep learning models, they are suitable for

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

working in real time and on low-powered devices.

- **Cross-Domain and Transfer Learning Limitations:** The analysis is limited to using only the CICIDS2017 data. The idea is to have researchers test their algorithm using datasets different than the ones they used in their research, for example NSL-KDD, UNSW-NB15 and CIC-DDoS2019.
- **Vulnerability to Adversarial Attacks:** The model has not been tested for its ability to withstand attacks. In the future, it will be important to learn how to strengthen a system by training it or through disturbance detection to ensure it performs well under threat situations.
- **Integration with Modern Architectures:** While it is practical now, including support for Blockchain, SDN and VANET could help improve trustworthiness, flexibility and automated functions in various locations.
- **Limited Use of Unsupervised and Semi-Supervised Learning:** Only supervised learning is utilized in this study. Further work should look into unsupervised and semi-supervised models, as they are very useful for data that is scarce or changes fast.

## 6.2 Research Question Outcomes

- **RQ1: Effectiveness Compared to Individual Classifiers**

  When compared to Naive Bayes, Decision Tree and Random Forest, the hybrid model achieves better re-sults. It reached an accuracy of 98.84% and an AUC-ROC of 99.33%, meaning that combining different models improves the detection's robustness.

- **RQ2: Improved Accuracy and Reduced False Positives**

  The hybrid model performs well in classifying cyberattacks (such as DDoS, PortScan, Web attacks) and achieves high accuracy, recall and balanced F1-score (98.86%). The accuracy of the model is proved when the confusion matrix reports few mistakes, meaning few false positives or negatives.

- **RQ3: Performance Trade-offs in Ensemble Learning**

  Despite KNN taking more time to make predictions ($O(nd)$), combining it with fast-learning classifiers like Naive Bayes and Decision Tree maintains the efficiency of the whole system. Combining models in an ensemble helps to solve weaknesses in one model, giving good results without requiring the heavy processing involved in deep


*E-mail: editor.jrla24@gmail.com*

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

neural networks.

### 6.3 Summary

It aims to overcome a range of crucial hurdles in the studies of intrusion detection, including factors such as un-derstandability, use with large datasets, dealing with skewed data, reproducible results and deploying the systems smoothly. It also pinpoints and outlines upcoming challenges involving adversarial resistance, integrating archi-tectures, adapting to other domains and unsupervised learning. The thorough research and gap-filling made this study valuable for modern cybersecurity use.

## 7. CONCLUSION

In the study, a machine learning-based approach for detecting attacks was introduced and tested on the CI-CIDS2017 dataset which is known for having difficult attacks like PortScan. To make the system more accu-rate and reliable, the ensemble model applies soft voting to combine Naive Bayes, Decision Tree and K-Nearest Neighbors. The performance of the proposed model was excellent, achieving an accuracy of 98.84%, precision of 99.12%, recall of 98.61%, an F1-score of 98.86% and an AUC-ROC value of 99.33%. It is evident from the results that the ensemble method works better than the individual classifiers and models found in the literature. Besides reporting key metrics, this research adds value by helping to address important issues in the field.

- It provides a computationally efficient and interpretable solution, making it suitable for real-time and edge deployments.
- It avoids evaluation bias by using a standardized preprocessing and evaluation pipeline.
- It mitigates class imbalance and avoids overfitting through strategic model integration rather than synthetic oversampling.

Its ability to handle complex data and be used in different situations indicates that the hybrid model is good for use in mid-sized forensic teams and enterprise networks.

**Future Work:** As a result of this study, there are several new topics for future studies. Testing and evaluating the model in a network under real traffic conditions will provide more proof of its use. Making the system explainable, checking for robustness when attacked and connecting it to SDN or blockchain systems can make it more dependable and trustworthy in changing and multi-user environments. All in all, this work improves machine learning-based

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

intrusion detection by making available a useful, well-balanced and real-world solution for both education and industry sectors.

## REFERENCES

[1] Muhammad Atta Othman Ahmed, Yasser AbdelSatar, Raed Alotaibi, and Omar Reyad. Enhancing internet of things security using performance gradient boosting for network intrusion detection systems. *Alexandria Engineering Journal*, 116:472–482, 2025.

[2] Srinivas Akkepalli and K Sagar. Copula entropy regularization transformer with $c^2$ variational autoen-coder and fine-tuned hybrid dl model for network intrusion detection. *Telematics and Informatics Reports*, 17:100182, 2025.

[3] Elham Abdullah Al-Qarni and Ghadah Ahmad Al-Asmari. Addressing imbalanced data in network intrusion detection: A review and survey. *International Journal of Advanced Computer Science and Applications*, 15(2):136–143, 2024.

[4] Estabraq Saleem Abduljabbar Alars and Sefer Kurnaz. Enhancing network intrusion detection systems with combined network and host traffic features using deep learning: deep learning and iot perspective. *Discover Computing*, 27(39), 2024.

[5] Md Liakat Ali, Kutub Thakur, Suzanna Schmeelk, Joan Debello, and Denise Dragos. Deep learning vs. machine learning for intrusion detection in computer networks: A comparative study. *Applied Sciences*, 15(4):1903, 2025.

[6] Muhammad Ali, Mansoor-ul Haque, Muhammad Hanif Durad, Anila Usman, Syed Muhammad Mohsin, Hana Mujlid, and Carsten Maple. Effective network intrusion detection using stacking-based ensemble approach. *International Journal of Information Security*, 22:1781–1798, 2023.

[7] Mohammed S Alshehri, Oumaima Saidani, Fatma S Alrayes, Saadullah Farooq Abbasi, and Jawad Ahmad. A self-attention-based deep convolutional neural networks for iiot networks intrusion detection. *IEEE Access*, 12:45762–45778, 2024.

[8] Abiodun Ayantayo, Amrit Kaur, Anit Kour, Xavier Schmoor, Fayyaz Shah, Ian Vickers, Paul Kearney, and Mohammed M Abdelsamea. Network intrusion detection using feature fusion with deep learning. *Journal of Big Data*, 10(1):167, 2023.

[9] Mohamed Aly Bouke and Azizol Abdullah. An empirical assessment of ml models for 5g

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

network intrusion detection: A data leakage-free approach. *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, 8:100590, 2024.

[10] Nitu Dash, Sujata Chakravarty, Amiya Kumar Rath, Nimay Chandra Giri, Kareem M. AboRas, and
N. Gowtham. An optimized lstm-based deep learning model for anomaly network intrusion detection. *Sci-entific Reports*, 15(1554), 2025.

[11] Lorenzo Diana, Pierpaolo Dini, and Davide Paolini. Overview on intrusion detection systems for computers networking security. *Computers*, 14(3):87, 2025.

[12] Katharina Dietz, Michael Mu¨hlhauser, Jochen Ko¨gel, Stephan Schwunger, Marleen Sichermann, Michael Seufert, Dominik Herrmann, and Tobias Hoßfeld. The missing link in network intrusion detection: Taking ai/ml research efforts to users. *IEEE Access*, 12:79815–79832, 2024.

[13] Sabrine Ennaji, Fabio De Gaspari, Dorjan Hitaj, Alicia K Bidi, and Luigi Vincenzo Mancini. Adversarial challenges in network intrusion detection systems: Research insights and future prospects. *ACM Computing Surveys*, 1(1):1–35, 2024.

[14] Hiba K. P. Fathima and P. P. Anugraha. A review on network intrusion detection. *International Journal of Scientific Research and Technology*, 2(1):60–66, 2025.

[15] Patrik Goldschmidt and Daniela Chuda´. Network intrusion datasets: A survey, limitations, and recommen-dations. *Computers & Security*, 2025. Preprint.

[16] Xiaowei Gu, Gareth Howells, and Haiyue Yuan. A semi-supervised soft prototype-based autonomous fuzzy ensemble system for network intrusion detection. *Applied Soft Computing*, 170:112719, 2025.

[17] James Halvorsen, Clemente Izurieta, Haipeng Cai, and Assefaw Gebremedhin. Applying generative ma-chine learning to intrusion detection: A systematic mapping study and review. *ACM Computing Surveys*, 56(10):257, 2024.

[18] S. Hariharan, Y. Annie Jerusha, G. Suganeswhari, S. P. Syed Ibrahim, Uday Tupakula, and Vijay Varad-harajan. A hybrid deep learning model for network intrusion detection system using seq2seq and convlstm-subnets. *IEEE Access*, 13:30705–30721, 2025.

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

[19] Rupesh Chandrakant Jaiswal et al. Network based intrusion detection system. *IRJMETS*, 2023. Available on ResearchGate.

[20] Adam Kadi, Aymene Selamnia, Zakaria Abou El Houda, Hajar Moudoud, Bouziane Brik, and Lyes Khoukhi. An in-depth comparative study of quantum-classical encoding methods for network intrusion detection. *IEEE Open Journal of the Communications Society*, 6:1129–1145, 2025.

[21] Richard Kimanzi, Peter Kimanga, Dedan Cherori, and Patrick K Gikunda. Deep learning algorithms used in intrusion detection systems - a review. *arXiv preprint arXiv:2402.17020*, 2024.

[22] Deepshikha Kumari, Prashant Pranav, Abhinav Sinha, and Sandip Dutta. A hybrid cheetah and grey wolf optimization algorithm for network intrusion detection. *Engineering Research Express*, 7(1):015256, 2025.

[23] Yafei Luo, Ruihan Chen, Chuantao Li, Derong Yang, Kun Tang, and Jing Su. An improved binary simulated annealing algorithm and tpe-fl-lightgbm for fast network intrusion detection. *Electronics*, 14(2):231, 2025.

[24] Louai A. Maghrabi. Automated network intrusion detection for internet of things: Security enhancements.
*IEEE Access*, 12:30839–30849, 2024.

[25] Ziadoon K Maseer, Qusay Kanaan Kadhim, Baidaa Al-Bander, Robiah Yusof, and Abdu Saif. Meta-analysis and systematic review for anomaly network intrusion detection systems: Detection methods, dataset, valida-tion methodology, and challenges. *IET Networks*, 13:339–376, 2024.

[26] Ziadoon K Maseer, Robiah Yusof, Baidaa Al-Bander, Abdu Saif, and Qusay Kanaan Kadhim. Systematic re-view for anomaly network intrusion detection systems: Detection methods, dataset, validation methodology, and challenges. *Preprint*, 2025.

[27] Salman Muneer, Umer Farooq, Atifa Athar, Muhammad Ahsan Raza, Taher M Ghazal, and Shadman Sakib. A critical review of artificial intelligence based approaches in intrusion detection: A comprehensive analysis. *Journal of Engineering*, 2024:1–16, 2024.

[28] Hamad Naeem, Farhan Ullah, and Gautam Srivastava. Classification of intrusion cyber-attacks in smart power grids using deep ensemble learning with metaheuristic-

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

based optimization. *Expert Systems*, 42(1):e13556, 2024.

[29] Tarak Nandy, Rafidah Md Noor, Raenu Kolandaisamy, Mohd Yamani Idna Idris, and Sananda Bhattacharyya. A review of security attacks and intrusion detection in the vehicular networks. *Journal of King Saud Univer-sity - Computer and Information Sciences*, 36:101945, 2024.

[30] Kiran Sudam Pawar and Babasaheb J Mohite. Design framework model for network intrusion detection with feature selection algorithms using machine learning. *IRJMETS*, 19(1), 2025.

[31] Shubhkirti Sharma, Vijay Kumar, and Kamlesh Dutta. Multi-objective optimization algorithms for intrusion detection in iot networks: A systematic review. *Internet of Things and Cyber-Physical Systems*, 4:258–267, 2024.

[32] Md Alamin Talukder, Md Manowarul Islam, Md Ashraf Uddin, Khondokar Fida Hasan, Selina Sharmin, Salem A Alyami, and Mohammad Ali Moni. Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. *Journal of Big Data*, 11(33), 2024.

[33] Koen T. W. Teuwen, Tom Mulders, Emmanuele Zambon, and Luca Allodi. Ruling the unruly: Designing effective, low-noise network intrusion detection rules for security operations centers. In *Proceedings of the ACM Asia Conference on Computer and Communications Security (ASIA CCS)*, pages 1–14. ACM, 2025.

[34] Mubarak Albarka Umar, Zhanfang Chen, Khaled Shuaib, and Yan Liu. Effects of feature selection and normalization on network intrusion detection. *Data Science and Management*, 8:23–39, 2025.

[35] Xiaosong Wang, Yuxin Qiao, Jize Xiong, Zhiming Zhao, Ning Zhang, Mingyang Feng, and Chufeng Jiang. Advanced network intrusion detection with tabtransformer. *Journal of Theory and Practice of Engineering Science*, 4(3):191–210, 2024.

[36] Patikiri Arachchige Don Shehan Nilmantha Wijesekara. Intrusion detection using blockchain in software-defined networking: A literature review. *Journal of Engineering Science and Technology Review*, 18(1):57–79, 2025.

[37] Chiming Xi, Hui Wang, and Xubin Wang. A novel multi-scale network intrusion

**Online ISSN: 3048-667X**
**CODEN (CAS, USA): JRLAB4**
**Impact Factor: 3.66**

detection model with transformer. *Scientific Reports*, 14:23239, 2024.

[38] Congyuan Xu, Yong Zhan, Guanghui Chen, Zhiqiang Wang, Siqing Liu, and Weichen Hu. Elevated few-shot network intrusion detection via self-attention mechanisms and iterative refinement. *PLOS ONE*, 20(1):e0317713, 2025.

[39] Hangsheng Zhang, Dongqi Han, Shangyuan Zhuang, Zhiliang Wang, Jiyan Sun, Yinlong Liu, Jiqiang Liu, and Jinsong Dong. Explainable and transferable adversarial attack for ml-based network intrusion detectors. *IEEE Transactions on Dependable and Secure Computing*, 2025.

[40] Yaohui Zhang. Fwa-svm network intrusion identification technology for network security. *IEEE Access*, 13:18579–18593, 2025.