IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026

DOI: 10.17148/IJARCCE.2026.15923

AI-Driven Network Intrusion Detection System Using Machine Learning

Rashmi[1], Abhinandan Raje Urs M B[2], Appu Gowda M P[3], Aravind Kumar P N[4], Keerthan K[5]

Assistant Professor, Department of Computer Science, East West College of Engineering, Bengaluru, India[1]

Student, Department of Computer Science, East West College of Engineering, Bengaluru, India[2]

Student, Department of Computer Science, East West College of Engineering, Bengaluru, India[3]

Student, Department of Computer Science, East West College of Engineering, Bengaluru, India[4]

Student, Department of Computer Science, East West College of Engineering, Bengaluru, India[5]

Abstract: The Network Intrusion Detection technique is an essential component of cybersecurity for detecting malicious activities in computer networks. However, signature-based system may have limitations in detecting some new or modified attacks. In this paper, an AI-based Network Intrusion Detection System has been developed that applies supervised machine learning techniques for classifying network traffic. It performs data preprocessing and feature preparation and applies Decision Tree, Random Forest, K-Nearest Neighbors (KNN), Extra Trees, and XGBoost algorithms to the CICIDS2017, CSE-CIC-IDS2018, NSL-KDD, and UNSW-NB15 datasets. The developed models are embedded in a web application that allows uploading of CSV files, traffic prediction, attack detection, and dashboard-based analysis of results.

Keywords: Network Intrusion Detection System, Machine Learning, Network Security, Cybersecurity, XGBoost, Random Forest, CICIDS2017, Network Traffic Classification.

I. INTRODUCTION

Computer networks have emerged as one of the most critical components of communication, business functions, education, finance, cloud computing, and other online services. With the growing reliance on networked systems, there has been a corresponding increase in the number of threats against these systems. Such attackers may take advantage of network weaknesses in order to engage in activities like denial of service, scanning, brute force attacks, and other malicious actions. Early detection of such activities is thus crucial in the area of network security.

Intrusion Detection System (IDS) is a security tool used to detect any suspicious behaviour or attack within the computer or network system. Traditional intrusion detection systems generally rely on pre-defined signatures or hand-crafted rules to detect the attacks that match the known pattern. While signature-based approach is useful for detecting the known attacks, it may have trouble detecting unknown patterns of attacks.

Another option is provided by machine learning because machine learning enables the system to learn from previously acquired traffic data. Unlike rules, which are specified by humans in advance, machine learning enables the system to learn what differentiates between normal behaviour and malicious attacks based on features of traffic. Thus, machine learning becomes an interesting solution for building intrusion detection systems.

This research involves the development of a Network Intrusion Detection System that utilizes AI algorithms for supervised machine learning. This Network Intrusion Detection System takes advantage of network traffic data for training and testing purposes and the models are incorporated into a web application. This web application has a very friendly user interface to facilitate uploading and analysis of network traffic data.

The primary aim of the study is to design a usable IDS solution by integrating machine learning-based classification with a user-friendly web interface. In addition, the model serves as the base of future developments for real-time traffic monitoring and security automation.

Key contributions from the study include:

1. Machine Learning approach for building Network Intrusion Detection System to classify the network traffic.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026
DOI: 10.17148/IJARCCE.2026.15923

2. Data preprocessing and preparation of the network traffic data for machine learning process.
3. Testing of five different Machine Learning Algorithms which include; Decision Tree, Random Forest, KNN, Extra Trees, and XGBoost.
4. Deployment of trained models into a web-based application for better analysis of the network traffic.
5. Representation of predictions results in an interactive dashboard for better understanding of detected network activities.

II. RELATED WORK

There has been significant research interest in the use of machine learning for intrusion detection due to the inherent challenges involved in detecting today's cyber attacks through traditional rule-based methods.

The earlier methods for detecting intrusions were based largely on signatures and rules. The usefulness of such systems is that they help detect attacks which have been identified earlier; however, their performance is contingent upon the presence and correctness of attack signatures. As attackers keep evolving their methodologies, there is a risk of missing something new using such techniques [8], [9].

IDS solutions that are based on machine learning try to solve this problem through learning of patterns from the network traffic data sets. In supervised machine learning approach, learning is done through traffic samples which are labelled and in such samples, records indicate a certain type of network behaviour. When making predictions, the model learns from the traffic sample [1], [6].

Some of the ML techniques that have been examined for intrusion detection include decision trees, random forests, support vector machines, KNN, gradient boosting techniques, and neural networks. The ensemble learning approaches are especially effective because they take advantage of information generated through different decision-making procedures [1], [6].

Deep learning techniques have also been examined for intrusion detection. Neural networks can represent intricate structures based on large amounts of data; however, they can be computationally expensive and need proper adjustment. In addition, the quality of the data used to train a model is another essential consideration independent of the chosen algorithm [3].

The current paper revolves around the construction of an effective IDS using machine learning approaches through comparison of several classification algorithms and incorporation of the trained models into a functional web application. Instead of emphasizing just the model performance, the suggested system takes into account the whole process, including traffic data preprocessing and visualization [2], [4], [5].

Explainable artificial intelligence has also been examined in intrusion detection systems to make machine learning-based decisions more understandable and trustworthy [12].

Prior research works has shown that the machine learning and deep learning approaches can be useful for detecting intrusions in networks, but still there are certain issues regarding data dependency, detecting novel attacks, and the generalization to different traffic environments [2], [5]–[7]. The above-mentioned work provides motivation for developing an IDS system that integrates traffic processing, multiple machine learning classifiers, model evaluation, and an application level interface.

Feature selection has also been researched for intrusion detection purposes, among which feature selection approaches using the UNSW-NB15 dataset and optimization techniques to detect informative network traffic features [10], [11].

TABLE I: COMPARISON OF EXISTING INTRUSION DETECTION APPROACHES

Reference	Approach / Technique	Dataset / Evaluation	Main Focus	Limitation
Fu [1]	Machine Learning	Network intrusion dataset	ML-based intrusion detection	Evaluation is focused on a specific experimental setting
Cantone *et al.* [2]	Machine Learning	Multiple IDS datasets	Cross-dataset generalization	Performance varies across datasets

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026

DOI: 10.17148/IJARCCE.2026.15923

Reference	Approach / Technique	Dataset / Evaluation	Main Focus	Limitation
Nitin *et al.* [3]	Deep Learning	Web application security / IDS studies	Deep learning for intrusion detection	Higher computational complexity and model requirements
Sabu *et al.* [4]	Machine Learning	Network traffic / DoS attacks	DoS attack detection	Focused primarily on DoS attack detection
Hutabarat and Asnar [5]	Machine Learning	Intrusion detection environment	Unknown-threat detection	Generalization to different traffic environments remains challenging
Zoppi *et al.* [6]	Supervised, Unsupervised and Meta-Learning	IDS datasets	Unknown attack detection	Detection performance depends on the learning strategy and dataset
Ahmad *et al.* [7]	Survey / Zero-Day Detection	Existing research studies	Zero-day attack detection	Zero-day detection remains an open research problem
Proposed System	Five ML classifiers	CICIDS2017, CSE-CIC-IDS2018, NSL-KDD, UNSW-NB15	Multi-dataset intrusion classification and web-based analysis	Current implementation primarily evaluates prepared network traffic data; real-time packet capture is future work

This comparison provides motivation for using machine learning as an integral part of the proposed system, as well as the significance of appropriate datasets and pre-processing. The review indicates that current research addresses various machine learning, deep learning, and combined techniques for intrusion detection; some papers specifically address certain types of attacks or certain datasets. Evaluation across multiple datasets is also critical since models can exhibit different performances depending on the dataset properties [2], [6]. The suggested system provides this evaluation by testing five classifiers on four standard datasets, as well as including the prediction phase in the web-based analysis system. Nonetheless, packet capture in real-time along with model adaptation belong to the future scope of work.

III. PROPOSED SYSTEM

The proposed system is an AI-based Network Intrusion Detection System which uses machine learning algorithms for analyzing the traffic on the network and detecting any suspicious activities.

The system comprises several key stages such as: data acquisition, data pre-processing, feature preparation, model training, model testing, prediction, and results visualization.

The overall workflow can be represented as:

Network Traffic Dataset → Data Preprocessing → Feature Preparation → ML Model → Attack Classification → Result Analysis → Dashboard

This system is made in such a way that the machine learning part is isolated from the user interface. It helps make the learned models available via backend APIs, while the frontend offers an interactive interface for the users.

A. System Architecture

The architectural elements include frontend application, backend server, machine learning component, and database/model storage elements.

Frontend offers the interface through which authentication, CSV file uploading, dashboard display, report reading, and other monitoring features can be performed. Backend processes the request sent by frontend and does the necessary processing. In the case of network traffic data being provided for analysis, backend processes the data by passing it through the preprocessing stage before it is passed to the relevant machine learning model.

Model produces prediction which is processed by backend and returned to the frontend. Dashboard displays the predictions in terms of information such as types of attacks, severity of attacks, probabilities of attack, and traffic data.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026

DOI: 10.17148/IJARCCE.2026.15923

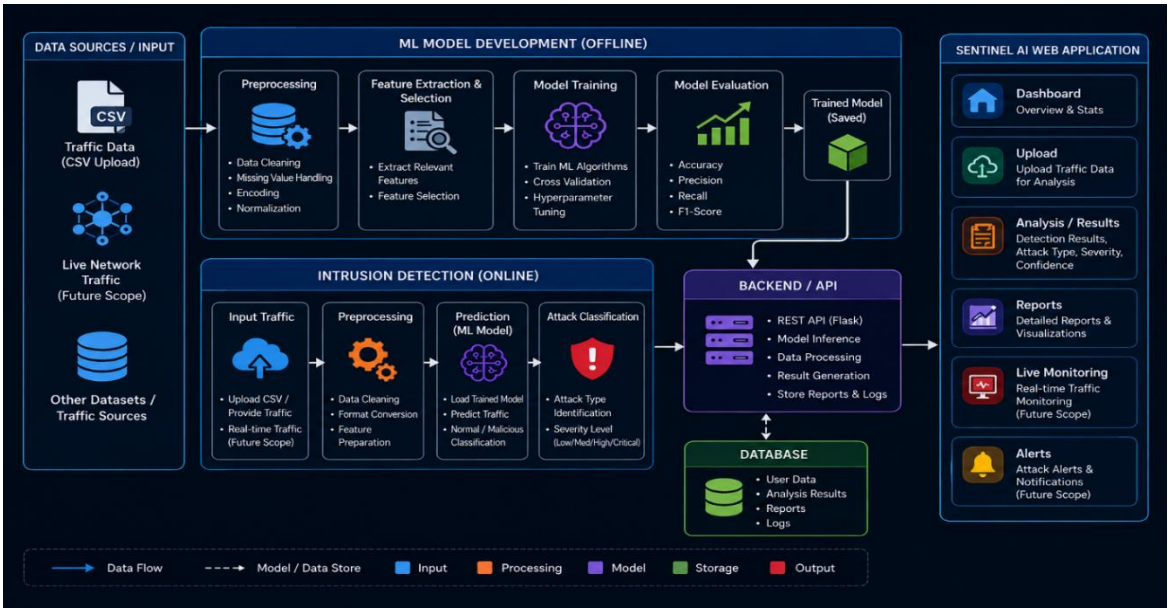

Fig. 1. Architecture of the implemented AI-driven network intrusion detection system

IV. METHODOLOGY

A. Dataset

The proposed system uses four benchmark network intrusion detection datasets for machine learning model development and evaluation: CICIDS2017 [13], CSE-CIC-IDS2018 [14], NSL-KDD [15], and UNSW-NB15 [16]. These datasets contain network traffic records representing legitimate and malicious activities and provide different characteristics of network behaviour for intrusion detection research. Using multiple datasets allows the proposed system to evaluate machine learning models across different network traffic environments rather than relying on a single dataset.

The datasets differ in terms of the number of records, input features, and attack classes. The processed datasets used in the implementation contain 2,231,806 records for CICIDS2017, 7,889,294 records for CSE-CIC-IDS2018, 147,907 records for NSL-KDD, and 145,222 records for UNSW-NB15. The number of input features used by the corresponding machine learning models is 77, 79, 39, and 31, respectively.

Data Availability: This research utilizes publicly available benchmark datasets, which include CICIDS2017, CSE-CIC-IDS2018, NSL-KDD, and UNSW-NB15, in its experiments. The corresponding dataset sources are cited in the reference list. The methodology adopted in preprocessing and training the model is presented in the manuscript.

TABLE II: DATASETS USED FOR MODEL DEVELOPMENT AND EVALUATION

Dataset	Records Used	Features	Classes	Train/Test
CICIDS2017	2,231,806	77	15	80/20
CSE-CIC-IDS2018	7,889,294	79	2	80/20
NSL-KDD	147,907	39	40	80/20
UNSW-NB15	145,222	31	2	80/20

For the model development process, the datasets were divided into training and testing subsets using an 80:20 split. The corresponding training and testing records were 1,785,444 and 446,362 for CICIDS2017; 6,311,435 and 1,577,859 for CSE-CIC-IDS2018; 118,325 and 29,582 for NSL-KDD; and 116,177 and 29,045 for UNSW-NB15. The NSL-KDD training data contains 40 encoded classes, while 36 of these classes are represented in its test split.

B. Data Preprocessing

It is necessary to pay close attention to the data preprocessing stage of the proposed approach since the quality of the data will have an impact on the quality of the models that would be trained.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026

DOI: 10.17148/IJARCCE.2026.15923

Preprocessing will be done on each dataset before training any models. The preprocessing consists of analyzing the data, filtering out invalid data, preparing the necessary features column, encoding the targets, and then the scaling as required for each machine learning model.

A simplified preprocessing pipeline is:

Raw Dataset → Data Cleaning → Feature Preparation → Label Encoding → Feature Scaling → Train/Test Dataset

Handling of missing or incorrect data takes place to ensure that such values don't cause any problems while building the model. There may be cases where the datasets used for network traffic include very large or infinite values that should be dealt with before using them for training the model.

The attack labels in the dataset are then represented in a machine-readable form. In case scaling is needed, the numerical attributes are scaled using the scaler used for training. This setup is further used for processing of new traffic data.

The same preprocessing elements that were utilized during model creation will be used during prediction so that the input traffic will be processed identically to the training dataset.

C. Feature Preparation

Network traffic has numerous attributes associated with each connection and its behaviour. However, all attributes are not equal contributors to the process of classification.

The implemented system prepares the required numerical attributes needed for classification according to the trained models. The set of attributes used for training will be kept in order to transform new traffic data into the same set of attributes as the one used for training.

The order of attributes should remain the same both for training and for prediction. If the order of attributes changes while predicting the result, it can cause incorrect results from the model.

V. MACHINE LEARNING MODELS

The proposed IDS incorporates five supervised machine learning algorithms including Decision Tree, Random Forest, K-Nearest Neighbors (KNN), Extra Trees, and XGBoost. All of these algorithms have been chosen for testing on the basis of being tree-based, ensemble, distance-based, and gradient-boosting algorithms.

A. Decision Tree

Decision Tree is a type of supervised learning approach which divides the dataset using feature values in a recursive manner. The nodes in the tree are the decision points based on features and leaves are the classes predicted from the decisions made at the nodes. In this IDS approach, the Decision Tree model will be trained using the network traffic features obtained.

B. Random Forest

Random Forest is an ensemble algorithm that utilizes decision trees for building a model and making a decision. Each tree decides separately, and the decision of each tree is then used for finding the final decision.

Regarding intrusion detection, Random Forest may be helpful because in the process of network traffic classification, several features may interact. The use of an ensemble decreases dependency on one decision tree.

In the proposed system, the trained Random Forest model is stored and then used by the backend to make decisions.

C. K-Nearest Neighbors

K-Nearest Neighbor assigns a class to the unknown instance according to the class of its neighbors in the feature space.

In case of a new record of network traffic, the algorithm finds out the closest training instances and makes predictions using their labels.

Though the KNN algorithm is easy to comprehend, its prediction cost may become higher in case the training set size is increased.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026
DOI: 10.17148/IJARCCE.2026.15923

D. Extra Trees

Another machine learning technique that involves the use of tree-based algorithms is Extra Trees or Extremely Randomized Trees, which adds further randomization in the process of building the decision tree.

It is similar to the random forest, where many trees are utilized in generating the final output. The further randomization could help in building diversified trees.

Extra Trees classifier is one of the classifiers included in the proposed system.

E. XGBoost

XGBoost is a type of gradient boosting algorithm that creates an ensemble of decision trees in a sequential manner. New trees try to correct the mistakes of the previously generated trees.

XGBoost is popularly used for tabular data and thus it can be used for network traffic classification as the input here includes mostly numerical traffic information.

This trained model of XGBoost can be included in the process of prediction for classifying incoming network traffic.

VI. MODEL TRAINING AND EVALUATION

Once pre-processed, the generated dataset is split into the training and testing sets.

The training set is employed in order for the algorithms to learn about relationships existing between network traffic features and their respective attack types. The test set is left untouched and will be employed for evaluation purposes later on.

The general training procedure is:

1. Load the network traffic dataset.
2. Clean invalid and missing values.
3. Separate input features and target labels.
4. Encode the target labels.
5. Apply the required feature transformation.
6. Divide the dataset into training and testing sets.
7. Train the selected machine learning algorithms.
8. Generate predictions using the testing data.
9. Calculate evaluation metrics.
10. Save the trained models and preprocessing components.

The models are evaluated using commonly used classification metrics such as accuracy, precision, recall, and F1-score.

Accuracy can be expressed as:

$$\textbf{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

where,
TP represents true positives,
TN represents true negatives,
FP represents false positives, and
FN represents false negatives.

Precision measures how many of the samples predicted as a particular class are actually members of that class:

$$\text{Precision} = \frac{TP}{TP+FP}$$

Recall measures how many of the actual positive samples were correctly detected:

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026

DOI: 10.17148/IJARCCE.2026.15923

$$\text{Recall} = \frac{TP}{TP+FN}$$

The F1-score combines precision and recall:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

These metrics are particularly relevant to intrusion detection because a model that simply achieves high overall accuracy may still perform poorly on less frequent attack classes.

For multi-class classification, precision, recall, and F1-score were calculated using weighted averaging across the classes.

VII. SYSTEM IMPLEMENTATION

The proposed IDS is implemented as a web-based application consisting of a frontend and backend layer.

A. Frontend

The frontend is the graphical user interface of the system. The frontend has pages for authentication, visualization of the dashboard, uploading of network traffic, results of the analysis, reporting, and monitoring.

The dashboard is used to visualize the results in order to avoid the need for the user to interpret the machine learning result.

The interface can display information such as:

- Total traffic records analyzed
- Normal traffic
- Detected attacks
- Attack categories
- Severity information
- Prediction confidence
- Traffic statistics
- Recent detection results

Charts and graphical components are used to make the detected traffic patterns easier to understand.

B. Backend

The backend is used as the channel of communication between the front-end and the machine learning models.

The backend accepts the uploaded CSV files, checks for validity, preprocesses the input, loads the machine learning model that has been trained, and predicts.

The general prediction flow is:

CSV Upload → Backend API → Data Validation → Preprocessing → Model Prediction → Result Processing → Frontend Dashboard

This architecture allows the machine learning models to operate independently of the user interface.

C. Prediction Pipeline

Once the user uploads the network traffic data, the backend will parse the CSV data and preprocess the data based on the preprocessing settings that were done while training the model.

The right model will take the processed features and generate a prediction. The prediction will be decoded to get the attack label using the label encoder.

The information is sent back to the frontend where it is shown to the user.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026

DOI: 10.17148/IJARCCE.2026.15923

This technique allows the same preprocessing techniques used during training to be used during the prediction phase.

VIII. RESULTS AND DISCUSSION

TABLE III: ACCURACY OF MACHINE LEARNING MODELS ACROSS DATASETS

Dataset	Decision Tree	Random Forest	Extra Trees	XGBoost	KNN
CICIDS2017	99.86%	99.85%	99.70%	99.89%	99.41%
CSE-CIC-IDS2018	100.00%	100.00%	99.96%	100.00%	99.94%
NSL-KDD	99.48%	99.67%	99.46%	99.70%	99.06%
UNSW-NB15	88.15%	90.77%	89.48%	90.79%	84.25%

The evaluation of the models was based on the accuracy, precision, recall, and F1 score. The values for the latter were obtained through the weighted average of the classes. From the results obtained, it can be seen that there is variation in performance depending on the dataset used.

In relation to the CICIDS2017 dataset, XGBoost achieved 99.89% accuracy, followed by Decision Tree with 99.86%, Random Forest with 99.85%, Extra Trees with 99.70%, and KNN with 99.41%. For the CSE-CIC-IDS2018 dataset, XGBoost, Decision Tree, and Random Forest achieved 100.00% accuracy, while Extra Trees and KNN achieved 99.96% and 99.94%, respectively. In the case of the NSL-KDD dataset, XGBoost achieved 99.70% accuracy, followed by Random Forest with 99.67%, Decision Tree with 99.48%, Extra Trees with 99.46%, and KNN with 99.06%. For the UNSW-NB15 dataset, XGBoost achieved 90.79% accuracy, followed by Random Forest with 90.77%, Extra Trees with 89.48%, Decision Tree with 88.15%, and KNN with 84.25%.

From the findings, it can be concluded that the efficiency of the model is dependent on the dataset. High levels of accuracy were obtained in CICIDS2017, CSE-CIC-IDS2018, and NSL-KDD datasets while comparatively lower accuracy was obtained on the UNSW-NB15 dataset. The above illustrates the impact of the dataset and traffic on the machine learning approach to intrusion detection systems.

Weighted Precision, Recall, and F1-Score also help to strengthen the accuracy measurements. XGBoost has reached 99.88% of Weighted Precision, 99.89% Weighted Recall, and 99.88% of Weighted F1-Score on CICIDS2017. In CSE-CIC-IDS2018, XGBoost, Decision Tree, and Random Forest have reached 100.00% of all three measurement values. On NSL-KDD, XGBoost has reached 99.66% weighted precision, 99.70% weighted recall, and 99.68% weighted F1-score. On UNSW-NB15, XGBoost obtained 90.78% weighted precision, 90.79% weighted recall, and 90.78% weighted F1-score.

To ensure computational effectiveness, KNN was tested using up to 50,000 samples for training and 20,000 samples for testing, while the other classifiers were tested on the prepared data sets. This should be taken into account when comparing computational performance of KNN with other classifiers.

The experimental results illustrate the capabilities of classification that have been provided by different machine learning techniques used in the proposed intrusion detection system. The comparison of accuracy, precision, recall, and F1-score gives a much more detailed perspective on model performance than only accuracy.

F1-score becomes especially significant as it could happen that classes included in an intrusion dataset differ a lot in terms of number of examples they include. This means that a model might show a good overall accuracy but will not be able to identify some less frequent attack classes.

A. Confusion Matrix

A confusion matrix is a tabular representation used to evaluate the performance of a classification model by comparing the actual class labels with the predicted class labels. The diagonal values represent correctly classified instances, whereas the off-diagonal values represent incorrect predictions among different traffic classes.

The confusion matrix of the XGBoost model on the CICIDS2017 dataset provides a class-level view of the classification performance.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026

DOI: 10.17148/IJARCCE.2026.15923

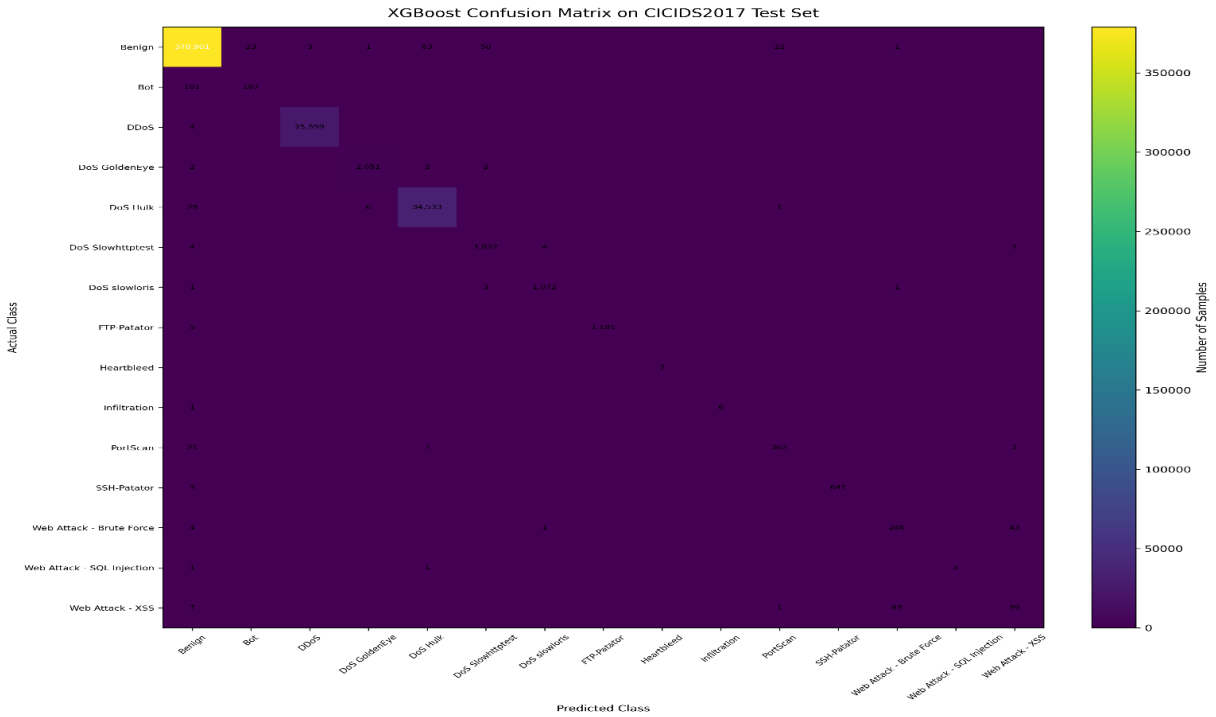

Fig. 2. Confusion matrix of the XGBoost model on the CICIDS2017 test set.

The confusion matrix offers a more elaborate perspective on how the classifier model behaves. Correctly classified instances are displayed on the diagonal, whereas incorrectly classified instances are displayed on the non-diagonal entries. Fig. 2 shows the confusion matrix of the classification done by XGBoost in the CICIDS2017 dataset. The diagonal elements of the matrix are the correctly classified instances, while the non-diagonal elements are the misclassified instances of one traffic type to another. It is evident from the confusion matrix that there are many instances along the diagonal, in agreement with the high accuracy rate of 99.89%.

B. Attack Detection Results

The developed application allows converting the predictions obtained from machine learning algorithms to the user-friendly format. In addition to just showing class label values, the application is able to provide information regarding the type of the predicted attack and its detection.

For example, the system can categorize traffic into classes such as:

- Normal Traffic
- DDoS
- Port Scan
- Brute Force
- SQL Injection
- Other attack categories represented in the trained model

IX. DISCUSSION

This implementation demonstrates the integration of machine learning techniques with a web-based intrusion detection system.

First of all, an important feature of the suggested approach is that the machine learning technique is built into an application and not tested in isolation as a prototype. In this way, it becomes possible for the network traffic to be fed into the application by a user and the predictions produced by the application to be provided in a more comprehensible form.

Secondly, another essential point about this implementation is that there are several machine learning algorithms employed. As they all learn in a different manner, it makes sense to evaluate them through comparison.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026
DOI: 10.17148/IJARCCE.2026.15923

However, intrusion detection based on machine learning also has certain drawbacks. The effectiveness of such a system largely depends on the quality and richness of the training data set. In case the training data do not fully reflect actual network conditions, the model may perform poorly when exposed to traffic patterns that differ substantially from the training data.

False detections are another problem that needs to be addressed. In a real-life setting, both false positives and false negatives matter. False positives can lead to being flooded with warnings, while false negatives mean that malicious traffic goes unnoticed.

Thus, the suggested system is rather a machine learning-based detection framework than an alternative to all other methods of network security.

X. LIMITATIONS

However, there are a number of problems in the current system.

Firstly, machine learning algorithms learn from the available benchmark network traffic data. These datasets cannot represent all possible network traffic conditions or attack scenarios.

Secondly, the analysis of the traffic is performed on the basis of CSV data which means that the traffic is required to be collected and transformed into a form acceptable for machine learning algorithms.

Thirdly, machine learning algorithms might output wrong classifications of traffic patterns which are significantly different from training data.

Finally, there are other requirements related to real-time analysis of the traffic including continuous traffic capturing and feature extraction, resource allocation, alerts processing, and communication of the monitoring and the analysis components.

XI. FUTURE SCOPE

The above mentioned system can be extended in many ways.

A. Real-time Network Packet Capturing and Prediction

The system may be enhanced to perform continuous capturing of network packets and generating predictions without any manual uploading of CSV files.

B. SIEM

The IDS may be extended for integration with Security Information and Event Management systems so that events detected by it could be correlated with other security systems' log files.

C. Automated Alarming

Automated alarming in case of exceeding of the threshold levels by suspicious activities may be performed by future versions of the system. The alarm may be delivered via an email message, via messengers or in the security dashboard.

D. Explainable AI

The explainable AI methods can be applied to make more understandable why a particular network packet was recognized as malicious.

E. Continuous Training of Models

The system can be further enhanced for periodic training with newly captured network traffic to improve model performance.

F. Advanced Attack Detection

Further development may focus on deep learning, anomaly detection, and other approaches to identify attacks not seen before.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026
DOI: 10.17148/IJARCCE.2026.15923

XII. CONCLUSION

In this study, an AI-based Network Intrusion Detection System was developed for network traffic detection and classification using supervised machine learning techniques. The proposed system integrates data preprocessing, feature preparation, machine learning classification, backend prediction services, and a web-based interface.

The following five machine learning techniques were tested and evaluated on four datasets used for benchmarking purposes in intrusion detection research including CICIDS2017, CSE-CIC-IDS2018, NSL-KDD, and UNSW-NB15 datasets: Decision Tree, Random Forest, K-Nearest Neighbors (KNN), Extra Trees, and XGBoost.

Models that have been trained were added into the developed web application, thus allowing users to upload traffic data and receive classification results using the dashboard. This means that the project allows for not only machine learning models comparison but also provides an interface for analyzing network traffic.

There are certain limitations associated with the use of benchmark datasets, CSV-based traffic input, and the absence of real-time packet capture in the current version of the system. Further improvements can involve real-time monitoring of traffic, integration with SIEMs, alerting, prediction explanation, continuous model training, and unknown attacks detection techniques.

Conflict of Interest: The authors declare that there is no conflict of interest regarding the publication of this paper.

AI-Assisted Editing: This manuscript was edited for language and formatting using ChatGPT (OpenAI). All technical content, experimental results, data, references, and system implementation details were verified by the authors.

REFERENCES

[1]. R. Fu, "Design and Implementation of Network Intrusion Detection System based on Machine Learning," in *2025 International Conference on Intelligent Systems and Computational Networks (ICISCN)*, Bidar, India, 2025, pp. 1–6, doi: 10.1109/ICISCN64258.2025.10934502.

[2]. M. Cantone, C. Marrocco, and A. Bria, "Machine Learning in Network Intrusion Detection: A Cross-Dataset Generalization Study," *IEEE Access*, vol. 12, pp. 144489–144508, 2024, doi: 10.1109/ACCESS.2024.3472907.

[3]. V. Nitin, Aarti, and V. N. Thatha, "Deep Learning for Web Application Security: A Comprehensive Survey on Intrusion Detection Systems," in *2025 2nd International Conference on Intelligent Systems for Cybersecurity (ISCS)*, Gurugram, India, 2025, pp. 1–7, doi: 10.1109/ISCS69371.2025.11386278.

[4]. I. R. Sabu, S. Saju, E. A. M. Anita, and T. Sowmya, "Detection of DoS Attacks Using Machine Learning Based Intrusion Detection System," in *2024 IEEE International Conference on Contemporary Computing and Communications (InC4)*, Bangalore, India, 2024, pp. 1–9, doi: 10.1109/InC460750.2024.10649128.

[5]. C. Hutabarat and Y. Asnar, "Development of Machine Learning Module in Intrusion Detection System for Unknown Threat Detection," in *2024 IEEE International Conference on Data and Software Engineering (ICoDSE)*, Gorontalo, Indonesia, 2024, pp. 114–119, doi: 10.1109/ICoDSE63307.2024.10829880.

[6]. T. Zoppi, A. Ceccarelli, T. Puccetti, and A. Bondavalli, "Which Algorithm Can Detect Unknown Attacks? Comparison of Supervised, Unsupervised and Meta-Learning Algorithms for Intrusion Detection," *Computers & Security*, vol. 127, p. 103107, 2023, doi: 10.1016/j.cose.2023.103107.

[7]. R. Ahmad, I. Alsmadi, W. Alhamdani, and L. Tawalbeh, "Zero-Day Attack Detection: A Systematic Literature Review," *Artificial Intelligence Review*, vol. 56, pp. 10733–10811, 2023, doi: 10.1007/s10462-023-10437-z.

[8]. A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges," *Cybersecurity*, vol. 2, art. no. 20, 2019, doi: 10.1186/s42400-019-0038-7.

[9]. S. Kumar, S. Gupta, and S. Arora, "Research Trends in Network-Based Intrusion Detection Systems: A Review," *IEEE Access*, vol. 9, pp. 157761–157779, 2021, doi: 10.1109/ACCESS.2021.3129775.

[10]. S. M. Kasongo and Y. Sun, "Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset," *Journal of Big Data*, vol. 7, art. no. 105, 2020, doi: 10.1186/s40537-020-00379-6.

[11]. H. Alazzam, A. Sharieh, and K. E. Sabri, "A Feature Selection Algorithm for Intrusion Detection System Based on Pigeon Inspired Optimizer," *Expert Systems with Applications*, vol. 148, art. no. 113249, 2020, doi: 10.1016/j.eswa.2020.113249.

[12]. B. Mahbooba, M. Timilsina, R. Sahal, and M. Serrano, "Explainable Artificial Intelligence (XAI) to Enhance Trust Management in Intrusion Detection Systems Using Decision Tree Model," *Complexity*, vol. 2021, art. no. 6634811, 2021, doi: 10.1155/2021/6634811.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026
DOI: 10.17148/IJARCCE.2026.15923

[13]. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *Proc. 4th International Conference on Information Systems Security and Privacy (ICISSP)*, Funchal, Portugal, 2018, pp. 108–116, doi: 10.5220/0006639801080116.

[14]. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018)," Canadian Institute for Cybersecurity, 2018. [Online]. Available: https://registry.opendata.aws/cse-cic-ids2018/

[15]. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," in *Proc. IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, Ottawa, Canada, 2009, pp. 1–6, doi: 10.1109/CISDA.2009.5356528.

[16]. N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Canberra, ACT, Australia, 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942.

