SCIENTIFIC CULTURE
www.sci-cult.com

SCIENTIFIC CULTURE, Vol.12, No.02, (2026), pp. 753-762
Open Access. Online & Print

DOI: 10.5281/zenodo.23984310

OPTIMIZING NETWORK SECURITY USING MACHINE LEARNING-BASED INTRUSION DETECTION SYSTEMS

PHOOL KUMARI[1], Vikash kumar[2], Twinkle kumari[3], Mahesh Singh[4], Pankaj Kumar[5], Prity Kumari[6]

[1]Assistant professor, CSE, Government Engineering College Lakhisarai, Lakhisarai pin 811306, BIHAR. Email id - phoolkumari.cusb@gmail.com, ORCID ID- 0009-0002-7908-6646
[2]Assistant professor, EEE, Power system security, RRSDCE BEGUSARAI. Begusarai,851134, Bihar. EMAIL- staarvk@gmail.com ,ORCID ID- 0009-0002-3834-8638.
[3]Assistant Professor, Computer science and engineering,. Government Engineering college Lakhisarai , Bihar. City / Pincode: 811311, Email ID: 1502twinkle@gmail.com.
[4]Assistant Professor, Department of Electrical and Electronics Engineering. Department of Electrical and Electronics Engineering, Rashtrakavi Ramdhari Singh Dinkar College of Engineering, Begusarai, Bihar. Email ID: maheshbgs512@gmail.com ORCID ID: 0000-0002-1121-7411.
[5]Assistant Professor, EEE, Power System, AIML, Rashtrakavi Ramdhari Singh Dinkar College of Engineering, Begusarai (Bihar). Email ID: pankajkr374@gmail.com ORCID ID: 0009-0002-5984-7778.*
[6] Assistant Professor, Dept. Electrical & Electronics Engineering, Rastrakavi Ramdhar Singh Dinkar College of Engineering (RRSDCE), Begusarai, Bihar-851134.City - Begusarai - 851134, Email id- prity030991@gmail.com, 0000-0002-3991-0875.

Received: 20/05/2026
Accepted: 30/06/2026

Corresponding Author: PHOOL KUMARI (phoolkumari.cusb@gmail.com)

ABSTRACT

This research deals with network security issues that result from the continued vulnerability of today's computer networks to malicious attacks and unauthorized accesses. An intrusion detection model based on machine learning is developed for distinguishing between normal and malicious network data. Different types of experiments were carried out using the NSL-KDD database, which consists of training data and testing data for evaluating the results. Multiclass attacks were labeled as binary data, where normal data was labeled as 0 and malicious data as 1. Data pre-processing included data cleansing by removing unnecessary columns, one-hot encoding, and normalization of numeric attributes. Four different machine learning algorithms were compared, namely, Decision Tree, Random Forest, Naïve Bayes, and Logistic Regression. Decision Tree algorithm proved to be the most effective, with the highest accuracy of 79.62% and F1 score of 79.71%.

KEYWORDS: Network Security, Intrusion Detection System, Machine Learning, NSL-KDD Dataset, Decision Tree, Random Forest, Cybersecurity, Binary Classification.

Copyright: © 2026. This is an open-access article distributed under the terms of the Creative Commons Attribution License. (https://creativecommons.org/licenses/by/4.0/).

1. INTRODUCTION

With the fast growth of digital communication, cloud computing, online services and interconnected network infrastructures, the importance of network security has grown significantly. Today, computer networks are essential to modern organizations, enabling them to share data, conduct financial transactions, manage business operations, healthcare systems, education platforms, and other vital communication services. But this reliance has also made organizations vulnerable to cyber attacks like denial-of-service attacks, probing attacks, unauthorized access, malware traffic and data breaches. With the increase in the frequency and sophistication of cyberattacks, traditional network security solutions are no longer effective in detecting new and evolving attack patterns. Intrusion detection systems thus have become an integral part of network security.

Intrusion Detection System is a system that is built to monitor the traffic or activities occurring on the network or system and detect suspicious or malicious activities. Traditional IDS techniques tend to rely on a set of rules or signatures. Signature-based systems work well against known attacks but are weak against unknown, modified and zero day attacks. Anomaly-based intrusion detection, on the other hand, tries to detect the abnormal behavior by learning the normal behavior of the network traffic. This makes the use of data-driven and machine learning based approaches more and more important in modern research on intrusion detection. Liu and Lang (2019) pointed out that machine learning and deep learning have been broadly adopted in IDS research due to their capability of learning the complex patterns from huge network data.

Machine learning can automatically discover the relationship between the features of network traffic and attack behavior, which is a practical approach to improving intrusion detection. Machine learning models can be trained using labelled network traffic data and then applied to new data to label it as normal or malicious traffic, rather than relying on manually written security rules. Kocher and Kumar (2021) highlighted that machine learning and deep learning techniques have proven to be very promising for intrusion detection, however, they also have the issues of data quality, feature selection, false alarm rate, and computational complexity. The challenges require considering several machine learning algorithms and finding models that are able to yield good performance with acceptable execution time.

Data-driven intrusion detection has become a hot topic in the last few years, as network traffic produces a large amount of structured and semi-structured data. Chou and Jiang (2021) were able to describe that data-driven network intrusion detection methods are based on the extraction of useful patterns from the traffic data, which helps in the detection of attacks. This is because these approaches can adapt themselves as opposed to static rule based approaches. The performance of a data-driven IDS, however, will be significantly affected by the data set used, the data pre-processing technique, the data representation and the classification algorithm. So, effective IDS framework should involve proper data cleaning, categorical data encoding, scaling of numerical data, training of the model, and evaluation of the model.

In recent network scenarios like software-defined networks, IDS models based on machine learning have been used as well. Alzahrani and Alenazi (2021) developed an intrusion detection system for software-defined networks using machine learning and demonstrated that intelligent classification techniques can be used to aid network security monitoring. Likewise, Kasongo and Sun (2020) showed how feature selection and machine learning techniques can boost the performance of IDS for benchmark datasets. Their results demonstrate the importance of preprocessing and feature handling to increase the accuracy of the detection and to remove unnecessary computational overhead.

In spite of the benefits of the machine learning, intrusion detection is still a difficult problem. Data imbalance is a big concern as some attack classes can occur far less frequently than normal traffic or other attack classes. Ahsan et al. (2022) talked about the significance of tackling data imbalance in machine learning-based network intrusion detection. The imbalanced data might lead models to be good at detecting majority classes but not the minority attack classes. A second difficulty is to choose the right algorithm. Al Lail et al. (2023) compared the various machine learning models for network intrusion detection and demonstrated that various algorithms can generate varying levels of accuracy, precision, recall, and F1-score.

From these problems, this study is aimed at optimizing the network security with machine learning based intrusion detection systems. The NSL-KDD dataset is used to train and test different machine learning classifiers. The original multiclass ID problem is reduced to binary class ID problem, namely normal or attack. Four machine learning models are tested: Decision Tree, Random Forest, Naive Bayes and Logistic Regression. The

comparison of the models is based on accuracy, precision, recall, F1-score, confusion matrix and training time. The overall goal of this research is to discover an efficient and appropriate machine learning model for malicious network traffic detection and better network security.

2. LITERATURE REVIEW

Machine learning has turned out to be one of the significant techniques for enhancing network intrusion detection systems in recent studies. Traditional security mechanisms rely on fixed rules and signatures which may not be effective against new or modified attacks. To address this limitation, machine learning-based intrusion detection systems try to learn the patterns from the network traffic data and categorize network traffic into normal or malicious traffic. Comparative studies are especially useful as they can be used to find which algorithms are more effective with certain datasets, preprocessing techniques and evaluation conditions.

In this paper, Zhang et al. did a comparative study of machine learning network intrusion detection methods. They highlighted that the results from various algorithms can vary based on the data, representation of features, and categories of attacks. The study emphasized the need for considering various machine learning models instead of a single machine learning classifier. This is applicable to the current study since the current study also compares several machine learning models such as Decision Tree, Random Forest, Naive Bayes and Logistic Regression to determine the most appropriate model for binary intrusion detection (Zhang et al., 2022).

An additional key research area for intrusion detection is explainability. Patil et al. investigated the explainable artificial intelligence for intrusion detection system. They said that while the machine learning models may help to improve the detection accuracy, black-box decision making may lead to loss of confidence in the security systems. Explainable methods enable security analysts to know the reasoning behind a model's classification of specific network traffic as normal or malicious. This highlights the significance of feature importance analysis in the field of IDS, as interpretability plays a crucial role in validating model predictions and fostering trust in automated security solutions (Patil et al., 2022).

Ali et al. gave a detailed overview of machine learning approaches and issues related to IDS. They highlighted key issues in the IDS performance such as data preprocessing, feature selection, selection of classifiers, quality of the data and evaluation metrics. The study also highlighted the need for careful experimental design when developing an IDS using machine learning as the performance of the IDS could vary depending on the preprocessing and model configuration. This aligns well with the current work which adopted categorical encoding, numerical scaling, binary label conversion, and multiple evaluation metrics to develop a structured IDS framework (Ali et al., 2024).

The research on NSL-KDD dataset is still relevant as it serves as a standard dataset to compare the performance of intrusion detection methods. Alshamy and Akcayol created a machine learning algorithms based intrusion detection model based on NSL-KDD data set. They showed that the supervised learning algorithms can be used in an effective way to classify network traffic and to measure the performance of IDS. Their research also directly supports the use of the NSL-KDD dataset in machine learning-based intrusion detection experiments, as the present study also utilizes this dataset.

Ferdiana presented a systematic view of intrusion detection research by discussing the trends, datasets and methods used in IDS research. The review revealed that the benchmark datasets, machine learning classifiers, and performance metrics like accuracy, precision, recall, and F1-score are widely used in IDS research. It also suggested that dataset selection and data preprocessing is of utmost importance to create reliable IDS models. This helps to guide the methodological approach of the current study which adopts the benchmark dataset and the model performance assessment through standard performance measures (Ferdiana, 2020).

Another important component that impacts the performance of the ID is the data preprocessing. Umar and others investigated the impact of feature selection and normalization techniques for intrusion detection in network. They found that the preprocessing can have a significant impact on the performance of the model as irrelevant features and unscaled numerical values can impact the ability of the model to classify. This is applicable to the current study because the unnecessary columns are dropped, categorical variables are encoded and the numerical features are scaled before training the models (Umar et al., 2025).

Alharthi et al. made a comparison between the machine learning and deep learning models for binary and multiclass intrusion detection classification. They found that the decision on binary or multiclass classification has an impact on the complexity of the model and how to interpret the performance. Binary classification can be simpler to apply and can be beneficial if the primary goal is to

differentiate between normal and malicious traffic. This inspired to transform the NSL-KDD multiclass labels into binary labels for easier and faster evaluation (Alharthi et al., 2025).

Mohale and Obagbuwa tested IDS models built with machine learning techniques with explainable AI. They pointed out the significance of transparency and interpretability in cybersecurity, stating that security decisions need to be understandable to the system administrators and analysts. This justifies the incorporation of feature importance analysis in the current study and the use of Decision Tree model to not only classify but also identify the influential traffic features (Mohale & Obagbuwa, 2025).

Optimized LSTM based Deep Learning model for Anomaly Network Intrusion Detection was proposed by Dash et al. Their research shows that deep learning can be successful in higher level IDS applications, particularly in the case of sequential or complex traffic patterns. But, deep learning models may need more computational resources and training time than traditional machine learning algorithms. Hence, the current study emphasizes the lightweight and faster machine learning models and the proposed approach is more feasible for the basic and efficient implementation of IDS (Dash et al., 2025).

3. METHODOLOGY

The methodology used in this study aims to create and test an intrusion detection system (IDS) that utilizes machine learning techniques to enhance network security. The goal is to categorize the traffic into two groups, namely normal traffic and attack traffic. NSL-KDD dataset was used for this purpose as it is a widely used benchmark dataset for intrusion detection research. The entire workflow of the proposed system is as follows: Dataset selection, data loading, data preprocessing, data transformation, model training, model testing, performance evaluation and result analysis. The methodology is organized around a structured machine learning pipeline, ensuring systematic and reproducible experimental process and suitable for research-based evaluation.

3.1 Dataset Description

The data set used in this study is NSL-KDD data set which is an improved version of KDD Cup 99 data set (M Hassan Zaib, 2019). It includes network connection records that have various traffic attributes and attack categories. In this research, two data set files are used: KDDTrain+_20Percent.txt for training and KDDTest+.txt for testing. The training file used was 20 percent because of the need to have enough data to train the model, but still maintain low execution time. There are 25,192 records in the training data set and 22,544 records in the testing data set. There are 43 columns in each record, such as the network traffic features, the class label, and a difficulty-level column.

There are both numerical and categorical features in the data. The features are divided into categorical and numerical features, for example protocol_type, service, flag and duration, src_bytes, dst_bytes, count, srv_count and some traffic rate features. In the original target column, multiple attack names are listed, as well as normal traffic. Since this study focuses on a simple and efficient intrusion detection approach, the original multiclass problem was converted into a binary classification problem. Thus all the records having normal label were classified as normal class and all other attack labels were classified as attack class (M Hassan Zaib, 2019).

3.2 Data Preprocessing

In this study, data preprocessing was a crucial step since the raw datasets were not always directly applicable to the machine learning models. Initially, the data set was loaded into Python using the Pandas library. Training and testing files were loaded and checked for the number of rows, number of columns, missing values and duplicate values. It was found that there were no missing values or duplicate values in the training and testing data sets. This ensured that the data set was clean and ready for further processing.

The basic data quality check was performed and unnecessary columns were removed. After developing the new binary target variable, the class column was removed. The difficulty column was also removed since it is not an input feature that is required for intrusion detection. The num_outbound_cmds column was eliminated because it was determined to be constant and thus did not provide any valuable information for the classification process. These columns were removed and the final feature set consisted of 40 input features prior to encoding.

The target variable was created by converting the original labels into binary form. The traffic was coded as 0 for normal traffic and 1 for attack traffic. The training set contained a total of 13,449 normal records and 11,743 attack records. The testing data set contained 9,711 normal records and 12,833 attack records. This binary transformation simplified the classification task, made it faster and more appropriate to evaluate the effectiveness of machine learning based intrusion detection.

3.3 Proposed Machine Learning-Based IDS Framework

The proposed intrusion detection framework is based on supervised machine learning approach. In

this model, the training data is utilized to train the models to differentiate between normal and malicious traffic in the network. Then the models are tested on unseen data in the testing data set. The framework starts with raw network traffic data and goes through the preprocessing and feature transformation stages. The transformed features are fed to various machine learning classifiers after pre-processing. Each classifier classifies whether a network record is from normal class or attack class.

There are five main stages of the framework. Data acquisition is the first stage, in which NSL-KDD training and testing files will be loaded. The second phase is the pre-processing phase that involves removing unwanted columns and generating binary labels. The third step is feature transformation, which involves transforming the categorical and numerical features into a format suitable for training the model. The fourth stage is the training of the model, which involves training various machine learning algorithms with the processed training data. The evaluation stage is the fifth stage where the trained models are evaluated using the processed testing dataset and compared based on the standard performance measures.

3.4 Machine Learning Models Used

The four machine learning algorithms that were implemented and tested in this study include Decision Tree, Random Forest, Naive Bayes, and Logistic Regression. These models were chosen as they are widely used for classification problems and have a relatively quick training time.

The principle of the Decision Tree classifier is to divide the data set into smaller decision rules on the basis of the value of the features. It is easy to understand, fast, and appropriate for classification problems for structured data. Random Forest is an ensemble learning technique which combines several decision trees to achieve better classification accuracy and avoid overfitting. Naive Bayes is a probabilistic classifier that uses Bayes' theorem and has the advantage of being quickly trained. Logistic Regression is a linear classification algorithm which estimates the probability of a record belonging to a particular class.

The training data and the testing data were processed and all four models were trained and tested on the same processed data. This gave an equal comparison between the model.

3.5 Feature Transformation

The data set has a combination of categorical and numeric features, which required feature transformation. One Hot Encoding was used for the categorical attributes protocol_type, service and flag. This method transforms categorical data into binary numerical columns to be used by machine learning algorithms. Unrecognized categories in the test data were encoded safely.

The numerical features were transformed using Standard Scaling. Standard Scaling is a technique used to rescale numerical values to be at a similar scale, for models like Logistic Regression to work better. The number of features was increased from 40 to 117 after applying the One Hot Encoding and Standard Scaling. To prevent data leakage, the preprocessing pipeline was only fit on the training data and then applied to both training and testing data.

3.6 Evaluation Metrics

The accuracy, precision, recall, F1-score, confusion matrix, and training time were used to evaluate the performance of each model. Accuracy is the percentage of correct predictions out of all predictions made. Precision is the number of attack records predicted that were indeed attacks. Recall is the number of true positive attacks that were detected. F1-score is helpful for intrusion detection tasks as it can balance precision and recall. Two classes (normal and attack) were correctly and incorrectly predicted using the confusion matrix. The training time has also been measured to evaluate the computational efficiency of each model.

With such measurements, it became possible to establish not only the best performing model but also the one that represents an optimal compromise between detection and computing efficiency. The method described in the research paper is quite clear and applicable in practice when working with IDS based on machine learning technologies.

4. RESULTS

This section presents the experiment results that have been achieved using the suggested ML-based network IDS. In order to measure the efficiency of several classifiers' work in the detection of malicious activity within the network, the NSL-KDD dataset was used. The binary approach to the analysis of traffic was applied, as the normal traffic was marked by '0', while the attack traffic received '1'.

4.1 Dataset Analysis

Prior to training our machine learning algorithms, we had first carried out an analysis on the dataset so that we could get insight into the data structure, the number of entries, and the distribution of classes. In this study, the training dataset used was KDDTrain+_20Percent.txt, and the test dataset was KDDTest+.txt. The training dataset consisted of 25,192 samples, whereas the test dataset had 22,544 samples. Each sample had 43 fields in total.

Table 1: Dataset Summary

Dataset	File Used	Rows	Columns	Missing Values	Duplicate Rows
Training Dataset	KDDTrain+_20Percent.txt	25,192	43	0	0
Testing Dataset	KDDTest+.txt	22,544	43	0	0

As shown in Table 1, both the training and testing datasets were complete, with no missing values and no duplicate records. This indicates that the dataset was suitable for preprocessing and model development without requiring missing value imputation or duplicate removal.

The original dataset contained multiple attack classes along with normal traffic. For this study, the multiclass labels were converted into binary labels to simplify the intrusion detection task. All records labelled as normal were assigned to the normal class, while all other attack categories were assigned to the attack class.

Table 2: Binary Class Distribution

Class Label	Target Value	Training Count	Testing Count	Training Percentage	Testing Percentage
Normal	0	13,449	9,711	53.39%	43.08%
Attack	1	11,743	12,833	46.61%	56.92%

Table 2 shows that the training dataset contains 13,449 normal records and 11,743 attack records. The testing dataset contains 9,711 normal records and 12,833 attack records. The distribution shows that both classes are represented in sufficient numbers, making the dataset appropriate for binary intrusion detection.

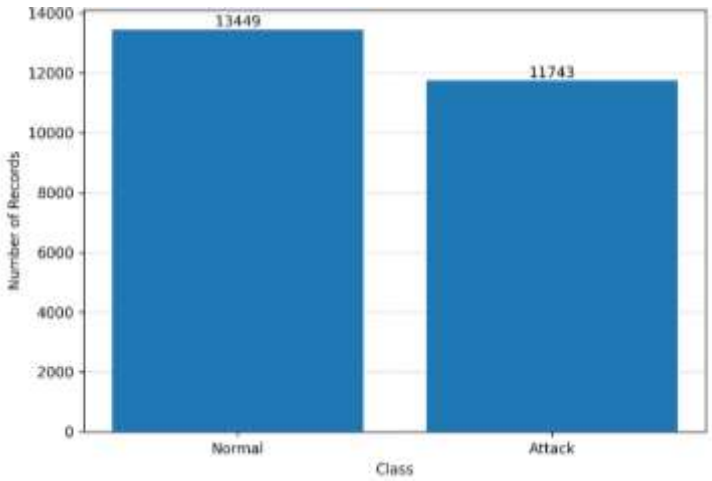


Figure 1: Training Dataset Binary Class Distribution

Figure 1 visually represents the distribution of normal and attack records in the training dataset. The graph shows that the dataset is reasonably balanced, with a slightly higher number of normal records compared to attack records. This balance is useful because highly imbalanced data can negatively affect model learning and classification performance.

4.2 Model Performance Comparison

After performing preprocessing, four different machine learning algorithms were implemented: Decision Tree, Random Forest, Naïve Bayes Classifier, and Logistic Regression Classifier. The categorical features were processed by using One-Hot Encoding technique, whereas the numerical attributes were scaled by applying Standard Scaling technique. After preprocessing, the number of attributes was increased from 40 to 117. Each algorithm was trained and tested on the same training and testing sets, respectively.

Table 3: Model Performance Comparison

Model	Accuracy	Precision	Recall	F1-Score	Training Time
Decision Tree	0.7962	0.9200	0.7032	0.7971	0.3265 sec
Random Forest	0.7801	0.9679	0.6347	0.7667	0.5797 sec
Logistic Regression	0.7508	0.9170	0.6182	0.7385	0.6791 sec
Naive Bayes	0.5626	0.9784	0.2368	0.3813	0.1444 sec

Table 3 depicts the performance evaluation for all the machine learning algorithms considered. The best accuracy was achieved by the Decision Tree classifier with 0.7962 accuracy and 0.7971 F1-score. On the other hand, the highest precision score was obtained using the Random Forest algorithm (0.9679), although recall was lower than that of the Decision Tree classifier. Logistic Regression model provided average results while the lowest F1-score was found with Naïve Bayes model.

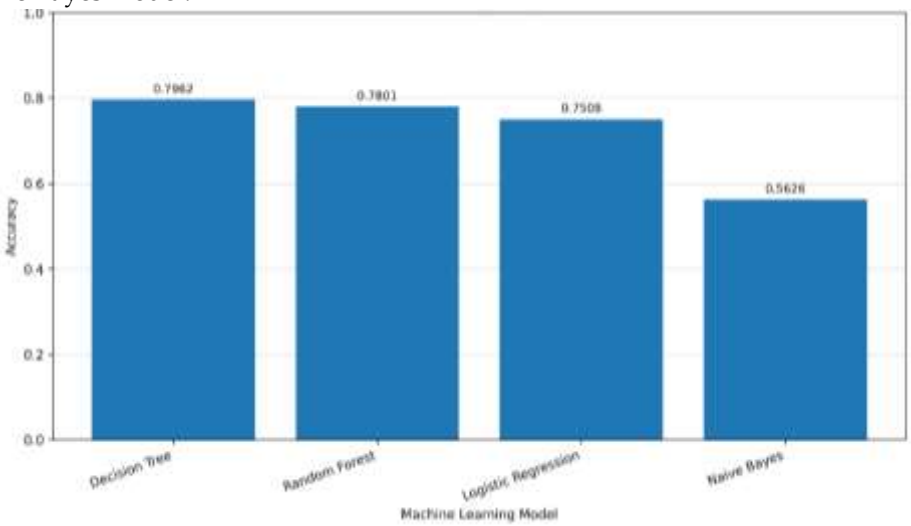


Figure 2: Model Accuracy Comparison

Figure 2 compares the accuracy values of all four machine learning models. The Decision Tree classifier achieved the highest accuracy, followed by Random Forest and Logistic Regression. Naive Bayes showed the lowest accuracy among the evaluated models.

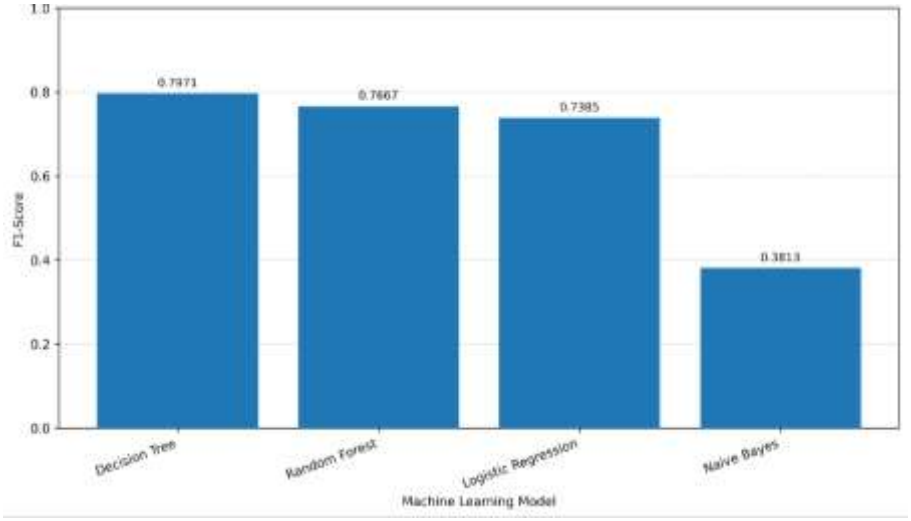


Figure 3: Model F1-Score Comparison

The performance of each model according to the F1-score is compared in Figure 3. As it has been noted earlier

that the F1-score takes into account precision and recall, thus, it makes sense to use it for intrusion detection purposes. In our analysis, the Decision Tree achieved the maximum value of F1-score.

4.3 Best Model Evaluation

Based on the model performance comparison, the Decision Tree classifier was selected as the best-performing model. It achieved an accuracy of 79.62%, precision of 92.00%, recall of 70.32%, and F1-score of 79.71%. These results indicate that the Decision Tree model was effective in classifying network traffic as normal or malicious.

Table 4: Best Model Performance

Metric	Value
Best Model	Decision Tree
Accuracy	0.7962
Precision	0.9200
Recall	0.7032
F1-Score	0.7971
Training Time	0.3265 sec

The final results obtained by the optimal model can be seen in Table 4. The Decision Tree algorithm was able to prove to be a very efficient solution in terms of accuracy, F1-score, and runtime. The training time for Decision Trees is equal to 0.3265 seconds, which makes this algorithm highly suitable for a fast intrusion detection test. A high value of precision means that most decisions made by the Decision Tree on whether an attack occurs or not were accurate.

However, the value of the recall equal to 0.7032 shows that some attacks were classified as normal behavior. Such a problem can be considered highly important from the point of view of an intrusion detection system. Therefore, although the Decision Tree performs better than any other algorithm tested, there is still much space left for improvement.

4.4 Confusion Matrix Analysis

To further analyze the classification behavior of the Decision Tree model, a confusion matrix was generated. The confusion matrix shows how many normal and attack records were correctly and incorrectly classified.

Table 5: Confusion Matrix of Decision Tree Classifier

	Predicted Normal	Predicted Attack
Actual Normal	8,926	785
Actual Attack	3,809	9,024

Table 5 shows that the Decision Tree classifier correctly classified 8,926 normal records and 9,024 attack records. It incorrectly classified 785 normal records as attacks and 3,809 attack records as normal traffic. The number of correctly classified records is higher than the number of misclassified records, which supports the overall effectiveness of the model.

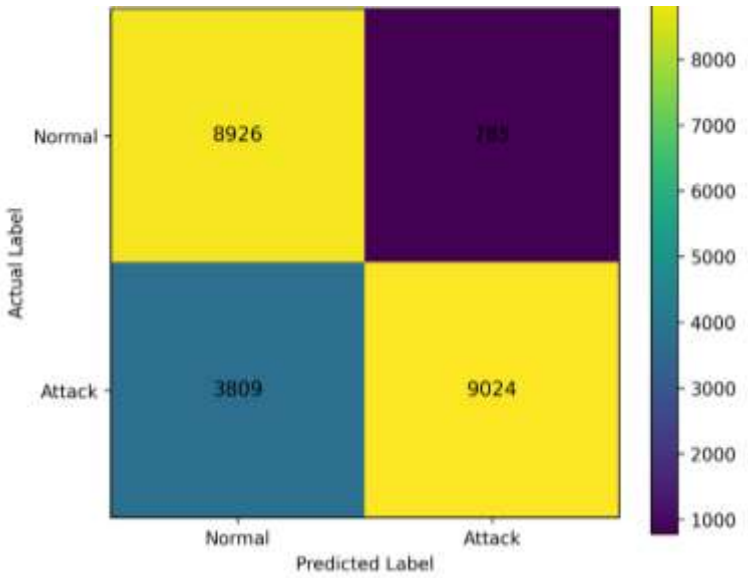


Figure 4: Confusion Matrix of Decision Tree Classifier

Figure 4 provides a visual representation of the confusion matrix. The matrix shows that the model performed strongly in detecting normal traffic and also identified a large number of attack records correctly. However, the false negative value, where attack records were predicted as normal, indicates that the model could be further improved to increase attack detection sensitivity.

4.5 Feature Importance Analysis

Feature Importance was carried out using the Decision Tree classifier for determining the influence of each feature on the classification process. Feature Importance adds clarity to the Intrusion Detection model by helping identify the most influential features in detecting any network attacks.

Table 6: Top 15 Important Features

Feature	Importance
src_bytes	0.729985
service_ecr_i	0.074766
service_http	0.054749
dst_host_same_srv_rate	0.026118
dst_bytes	0.023955
hot	0.022988
service_ftp_data	0.011425
logged_in	0.009847
dst_host_same_src_port_rate	0.008329
dst_host_srv_count	0.005099
duration	0.003818
dst_host_rerror_rate	0.003371
dst_host_srv_diff_host_rate	0.003344
dst_host_count	0.00294
count	0.00206

Table 6 presents the top 15 most important features identified by the Decision Tree classifier. These features had the highest influence on the model's decision-making process. Feature importance analysis helps explain why the model classified specific network records as normal or malicious. Overall, the experimental results show that machine learning models can be effectively applied to network intrusion detection. Among the evaluated models, the Decision Tree classifier achieved the best overall performance with an accuracy of 79.62% and an F1-score of 79.71%. The results also show that model comparison, confusion matrix analysis, and feature importance analysis provide useful insights into the performance and behavior of machine learning-based intrusion

detection systems.

5. DISCUSSION

The experimental results obtained from this study indicate that machine learning can be effectively used in network intrusion detection for traffic classification whether it is normal or malicious. The Decision Tree classifier was found to be the best model based on the overall performance with an accuracy of 79.62% and an F1-score of 79.71% among the four models evaluated. This means that the model achieved the best balance between correct classification and detection reliability as compared to Random Forest, Logistic Regression and Naive Bayes. The Decision Tree model was also trained in a relatively small time, which makes it suitable to a fast and lightweight implementation of an intrusion detection system.

One of the key findings from the results is that the Decision Tree classifier achieved a high precision score of 92.00%. This indicates that most of the records that the model predicted as attacks were indeed attacks. High precision is desirable in network security because it decreases false alarms and eliminates the need to investigate innocuous network traffic. But, the recall value is 70.32% indicating that some attack records were still identified as normal. This is a critical constraint as false negatives can be more damaging in IDS – attacks may go on undetected within the network. Hence, even though Decision Tree classifier was the best in this study, there is still need for further optimization to increase the sensitivity of attack detection.

The confusion matrix gives more information about the behavior of the model. The Decision Tree model classified 8,926 normal and 9,024 attack records correctly. It also incorrectly labeled 785 normal records as attacks and 3,809 attack records as normal. This indicates that the model was successful in detecting a significant amount of both legitimate and attack traffic, but failed to detect a certain percentage of malicious traffic. Understanding model decisions can help enhance trust and weaknesses, which is why Wang et al. (2020) highlighted the importance of explainable machine learning frameworks for intrusion detection. Feature importance analysis was also incorporated in this study to aid in interpretability, and to indicate which traffic attributes contributed most to the classification decisions.

The model comparison also reveals the different behavior of models to the same data set. Random Forest had very high precision but low recall and Naive Bayes trained fast but had poor performance in the F1 score. The results obtained by Logistic Regression were moderate and it did not outperform the tree-based models. This helps to strengthen the fact that there is a need to compare the machine learning algorithms to choose the final IDS model. Ali et al. (2025) contrasted machine learning and deep learning methods for intrusion detection, emphasizing the importance of choosing the appropriate model based on performance, complexity, and computational expense. For this research, conventional machine learning models were chosen since they are faster, easier to implement, and more appropriate to experiment with lightweight IDS.

The results are consistent with the overall studies indicating the usefulness of AI and ML in network intrusion detection. To enhance the capability of detection, Vanin et al. (2022) stated that AI and ML based IDS can learn patterns from network data, instead of relying solely on static rules. The current investigation utilizes supervised learning models trained using labeled data from NSL-KDD. The results prove that machine learning is capable of classifying network traffic, especially when proper preprocessing is used, including converting categorical, numeric attributes, and binarizing labels.

Despite the promising outcomes of the experiment, there exist some limitations. The analysis was based on the use of the benchmark dataset and conducted as a part of binary classification. Therefore, the approach can be viewed as one of the simplest realizations of IDS. Attacks can be rather complicated in practice, requiring better models. Neto et al. (2025) argue that due to the ability to detect complex patterns in big datasets, deep learning models have great potential for intrusion detection in emerging technology applications. It means that the implementation of LSTM, CNN, and other state-of-the-art models in combination with more recent datasets and practical testing may become a good focus for the future research. Overall, the results demonstrate the possibility of developing IDS models via machine learning and underline the need to enhance their quality.

6. CONCLUSION

In this case, the focus was on developing and testing the effectiveness of an Intrusion Detection System (IDS), aimed at improving network security using machine learning algorithms. The NSL-KDD dataset was used to identify network traffic as either normal or attack. Binary labels rather than the existing multi-class labels were used as part of

simplifying the IDS and making comparisons possible. Pre-training of machine learning algorithms involved data preprocessing by eliminating irrelevant columns, encoding categorical features and scaling numerical features. Four algorithms were tested and compared; Decision Tree, Random Forest, Naive Bayes and Logistic Regression. The Decision Tree classifier recorded the highest performance metrics, including accuracy (79.62%), precision (92.00%), recall (70.32%) and F1 score (79.71%). Confusion matrix results reveal that the classifier identified 8,926 and 9,024 records as normal and attacks respectively. This experiment proves that machine learning is efficient when it comes to intrusion detection because of the ability to learn patterns from classified network traffic. Nevertheless, some attacks records were incorrectly classified as normal and thus further improvement is necessary. Other research steps to consider include tuning the hyperparameters, selecting better features and developing more sophisticated models like deep learning models, and implementing real-time IDS using up-to-date datasets.

REFERENCES

Ahsan, R., Shi, W., & Corriveau, J. P. (2022). Network intrusion detection using machine learning approaches: Addressing data imbalance. IET Cyber-Physical Systems: Theory & Applications, 7(1), 30-39.

Al Lail, M., Garcia, A., & Olivo, S. (2023). Machine learning for network intrusion detection—a comparative study. Future Internet, 15(7), 243.

Alharthi, A., Alaryani, M., & Kaddoura, S. (2025). A comparative study of machine learning and deep learning models in binary and multiclass classification for intrusion detection systems. Array, 26, 100406.

Ali, A. H., Charfeddine, M., Ammar, B., Hamed, B. B., Albalwy, F., Alqarafi, A., & Hussain, A. (2024). Unveiling machine learning strategies and considerations in intrusion detection systems: a comprehensive survey. Frontiers in Computer Science, 6, 1387354.

Ali, M. L., Thakur, K., Schmeelk, S., Debello, J., & Dragos, D. (2025). Deep learning vs. machine learning for intrusion detection in computer networks: A comparative study. Applied Sciences, 15(4), 1903.

Alshamy, R., & AKCAYOL, M. (2024). Intrusion detection model using machine learning algorithms on NSL-KDD dataset. International Journal of Computer Networks and Communications, 16(6).

Alzahrani, A. O., & Alenazi, M. J. (2021). Designing a network intrusion detection system based on machine learning for software defined networks. Future Internet, 13(5), 111.

Chou, D., & Jiang, M. (2021). A survey on data-driven network intrusion detection. ACM Computing Surveys (CSUR), 54(9), 1-36.

Dash, N., Chakravarty, S., Rath, A. K., Giri, N. C., AboRas, K. M., & Gowtham, N. (2025). An optimized LSTM-based deep learning model for anomaly network intrusion detection. Scientific Reports, 15(1), 1554.

Ferdiana, R. (2020, November). A systematic literature review of intrusion detection system for network security: Research trends, datasets and methods. In 2020 4th International Conference on Informatics and Computational Sciences (ICICoS) (pp. 1-6). IEEE.

Kasongo, S. M., & Sun, Y. (2020). Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. Journal of Big Data, 7(1), 105.

Kocher, G., & Kumar, G. (2021). Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges. Soft Computing, 25(15), 9731-9763.

Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. applied sciences, 9(20), 4396.

M Hassan Zaib. (2019). NSL-KDD [Data set]. Kaggle. https://www.kaggle.com/datasets/hassan06/nslkdd

Mohale, V. Z., & Obagbuwa, I. C. (2025). Evaluating machine learning-based intrusion detection systems with explainable AI: enhancing transparency and interpretability. Frontiers in Computer Science, 7, 1520741.

Neto, E. C. P., Iqbal, S., Buffett, S., Sultana, M., & Taylor, A. (2025). Deep learning for intrusion detection in emerging technologies: a comprehensive survey and new perspectives. Artificial Intelligence Review, 58(11), 340.

Patil, S., Varadarajan, V., Mazhar, S. M., Sahibzada, A., Ahmed, N., Sinha, O., ... & Kotecha, K. (2022). Explainable artificial intelligence for intrusion detection system. Electronics, 11(19), 3079.

Umar, M. A., Chen, Z., Shuaib, K., & Liu, Y. (2025). Effects of feature selection and normalization on network intrusion detection. Data Science and Management, 8(1), 23-39.

Vanin, P., Newe, T., Dhirani, L. L., O'Connell, E., O'Shea, D., Lee, B., & Rao, M. (2022). A study of network intrusion detection systems using artificial intelligence/machine learning. Applied Sciences, 12(22), 11752.

Wang, M., Zheng, K., Yang, Y., & Wang, X. (2020). An explainable machine learning framework for intrusion detection systems. IEEE access, 8, 73127-73141.

Zhang, C., Jia, D., Wang, L., Wang, W., Liu, F., & Yang, A. (2022). Comparative research on network intrusion detection methods based on machine learning. Computers & Security, 121, 102861.