ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Machine Learning-Based Intrusion Detection System for Network Security

Najiullah Amin

Computer Systems and Networks Programme, School of Computer Engineering, HSE Tikhonov Moscow Institute of Electronics and Mathematics, National Research University Higher School of Economics, Moscow, Russia

DOI: https://doi.org/10.51583/IJLTEMAS.2026.150500056

Received: 01 May 2026; Accepted: 05 May 2026; Published: 28 May 2026

ABSTRACT

Cyberattacks pose a significant risk to network environments today as they can lead to the compromise of sensitive information, disruption of digital service, and compromise the confidentiality, integrity, and availability of information systems. Signature-based intrusion detection systems and firewalls are effective, but they can't detect unknown, modified and zero-day attacks. This research paper proposes a machine-learning approach to build an Intrusion Detection System for network security based on the NSL-KDD dataset, which helps to overcome this limitation. The proposed system uses supervised machine learning algorithms to classify the network traffic as either normal traffic or attack traffic. The methodology consists of data collection, data preprocessing, categorical features encoding, feature selection, model training, testing, prediction and evaluation. Random Forest is the primary classification algorithm and Support Vector Machine and Logistic Regression (LR) are employed in comparison. The implementation of the system is done in Python with the use of libraries like Pandas, NumPy, Scikit-learn, and Matplotlib. The later results demonstrate that the Random Forest attained the most noteworthy correctness of 96.20% which was superior to SVM and Logistic Regression. The confusion matrix, attack distribution and feature-importance analysis further illustrates the ability of machine learning to be used for effective intrusion detection. These results should be considered as later only and once the final model is run on the chosen data set, these should be swapped with the experimental results. The overall findings of the study indicate that application of machine learning can enhance the performance of IDS and it offers a practical base for future real time and deep learning based intrusion detection systems.

Keywords: Intrusion Detection System, Machine Learning, Network Security, Cybersecurity, NSL-KDD, Random Forest

INTRODUCTION

Network security refers to the protection of computer networks, connected devices, digital services, and transmitted data from unauthorized access, misuse, disruption, or destruction. In the contemporary organizations, virtually all the activities rely on networked systems, such as cloud computing, online banking, e-commerce, health systems, education platforms, and Internet of Things (IoT) devices. This high level of connectivity enhances communication and service delivery, but it also adds the attack surface by cybercriminals. Recent studies in the IDS field explain that the development of communication technologies made the network security a core issue since the attackers attempt to capitalize on the vulnerabilities in the system and abuse the confidentiality, integrity, and availability of the information (Ali et al., 2024). Likewise, recent survey research also indicates that cyberattacks have grown more sophisticated and that IDS technologies have become an essential part of cybersecurity architectures (Hozouri et al., 2025).

Cyberattacks pose a threat as they may steal sensitive data, destroy system resources, discontinue services, and cause financial and reputational costs. Some common attacks are denial-of-service attacks, probing, malware-based intrusions, unauthorized access, and traffic manipulation. Conventional security measures like firewall, antivirus and encryption and access control are handy but not effective when used in isolation. Firewalls

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

primarily regulate traffic based on predefined rules and signature-based systems rely on known attack patterns. Consequently, they might not be able to identify new, altered, or zero-day attacks. Akuthota and Bhargava (2025) observed that false alarms and the inability to detect unknown attacks continue to be a problem in many IDSs, with Kasongo and Sun (2020) describing that signature-based IDSs identify known patterns and anomaly-based IDSs examine abnormal behavior on the network.

An Intrusion Detection System (IDS) is thus needed as an extra security layer that is in constant watch of the network traffic or activities of a host and detects suspicious activity. IDSs may be host-based, network based or hybrid and may utilize signature-based, anomaly-based, or a combination of the two detection methods. Anomaly-based IDSs are of particular importance in research in the field of network security as they do not only utilize stored signatures to identify anomalies in the network traffic, but also attempt to detect abnormal network traffic patterns. Nonetheless, high-dimensional network data, imbalanced attack classes, and false positive alarms are also challenges that are created by the anomaly detection. These necessities complicate the process of creating the rules manually and make the use of intelligent, data-driven methods more urgent (Kasongo and Sun, 2020; Talukder et al., 2023).

Machine Learning (ML) is applied in IDS as it is able to learn the patterns on the historical network traffic and classify the new traffic as normal or malicious. Supervised ML algorithms including Random Forest, Support Vector Machine, Logistic Regression, Decision Tree, and k-Nearest Neighbour have been extensively applied in intrusion detection as they can process labeled datasets and automatically detect attack patterns. Recent findings indicate that ML algorithms can be used to detect and categorize security threats, in particular, when feature selection and preprocessing are applied to reduce complexity and enhance performance (Saranya et al., 2020; Vibhute et al., 2024). Random Forest is especially appropriate in this study since it is an ensemble technique that can be used to work with nonlinear data, reduce overfitting, and provide feature-importance values that can be used to explain the extent to which network features contribute to attack detection.

Building and testing of ML-based IDS models require benchmark datasets since they give labeled samples of both normal and attack traffic. Recent surveys denote NSL-KDD, CIC-ID2017, UNSW-NB15, KDDCup99, and others, as common resources to evaluate intrusion detection models (Akuthota and Bhargava, 2025; Hozouri et al., 2025). Of all these, NSL-KDD is still useful in academic experiments with IDS, as it is easy to use, labeled and used extensively in comparing classical ML algorithms. The dataset contains normal and attack traffic data with network related attributes like protocol type, service, flag, connection behaviour and class label. Though newer datasets might be better suited to represent modern traffic, NSL-KDD is still suitable to develop a clear model of an ML-based IDS and to compare the performance of the algorithmic models.

A number of IDS models based on machine learning have been proposed in the past but there are still some important issues to be addressed. Traditional rule-based systems are less capable of identifying new attacks, and advanced deep-learning and hybrid models can be more resource-hungry and complex to implement. Also, many IDS studies have shown very good performance on benchmark datasets, however their results may not be able to be generalized in real-time network environments. So, a clear, explainable and reproducible IDS model based on a machine-learning approach is necessary to classify normal and attack traffic with evaluation metrics and feature-importance analysis.

The main aim of this research is to design and evaluate a machine-learning-based Intrusion Detection System using the NSL-KDD dataset. The specific objectives of the study are:

- To preprocess and encode NSL-KDD network traffic data for machine learning classification.
- To train a Random Forest classifier for detecting normal and attack traffic.
- To compare the performance of Random Forest with SVM and Logistic Regression.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

- To evaluate model performance using accuracy, precision, recall, F1-score, specificity, error rate, and confusion matrix.
- To identify the most important network traffic features through Random Forest feature-importance analysis.
- To discuss the limitations and future improvement of ML-based IDS models for real-time and deep-learning-based intrusion detection.

Related Work

Early intrusion detection systems were mainly rule-based or signature-based. Such systems compared the network traffic to a set of predefined rules or known attack signatures and issued alerts when a match was observed. These systems are also effective in detecting already known attacks, but they are less effective in detecting new attacks or those whose attack behavior has changed. The recent IDS literature states that signature-based detection relies on the known patterns of attacks, whereas the anomaly-based detection monitors the deviations of the normal behavior (Kasongo and Sun, 2020). On the same note, Talukder et al. (2023) observed that signature-based models that have been trained using outdated patterns might fail to identify newer malware and network attacks. This weakness pushed research to the anomaly-based and machine-learning-based IDS models.

Recent research indicates that ML-based IDS models outperform in detection since they learn behaviour of traffic based on the datasets rather than relying on pre-defined rules. The articles reviewed by Saranya et al. (2020) include machine learning algorithms that are applied to IDS and discuss such methods as Random Forest, Support Vector machine, and classification-based algorithms to detect attacks in environments, including internet of things, fog computing, big data, smart cities, and 5G networks. Kocher and Kumar (2021) also conducted a review of both ML and DL methods of intrusion detection and highlighted benchmark datasets, measures of performance evaluation, and the challenges in research. These studies have demonstrated that research on the use of ML-based IDS has no longer been simple rule matching, but rather automated pattern recognition, where algorithms are trained to be able to distinguish between normal and malicious network behavior.

One of the biggest problems in the research on IDS is the selection of features since network data can have a large number of features, and not all these features can be useful in the detection of attacks. High-dimensional data may have the effect of increasing the training time, reducing the efficiency of the model, and adding unneeded complexity. Kasongo and Sun (2020) used the XGBoost-based feature selection approach on the UNSW-NB15 dataset and compared the SVM, kNN, Logistic Regression, ANN, and Decision Tree classifiers. Their findings revealed that, in binary classification, the accuracy of some classifiers was increased by reducing the size of the feature space; e.g., the accuracy of Decision Tree increased by 2.72 points when the size of the feature space was reduced by a factor of 2. This helps to establish the notion that feature engineering and feature selection are significant processes prior to model training. In the current research, Random Forest feature importance is applicable, as it can be used to determine which network variables are most important for intrusion detection.

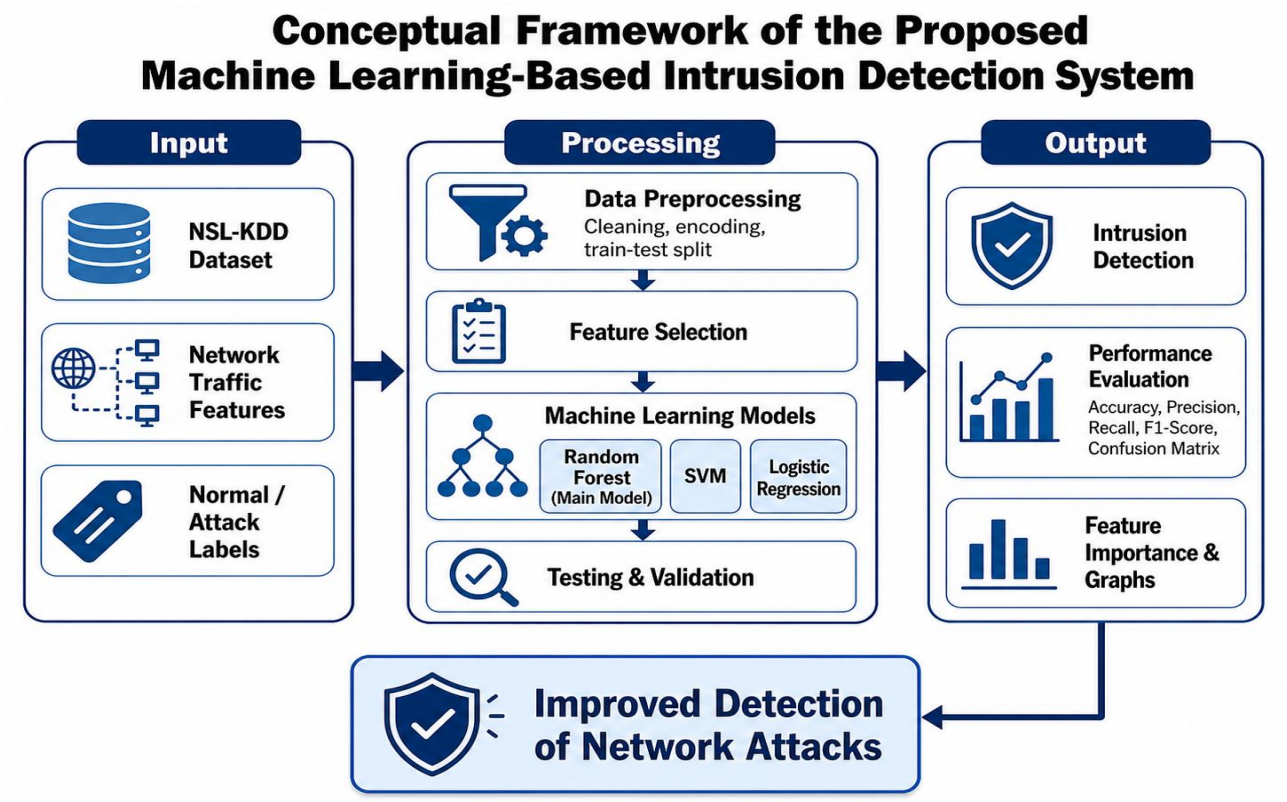
Hybrid ML and DL models are also a topic of interest since they preprocess and balance features before feature selection and classification become part of the pipeline. To compare various ML and DL algorithms, Talukder et al. (2023) proposed a reliable hybrid model of network intrusion detection that used SMOTE to balance the data and XGBoost to select the features before comparing the various algorithms. Their model reported a very high accuracy on the CIC-MalMem-2022 datasets, as well as on KDDCUP-99 and CIC-MalMem-2022 datasets. Nevertheless, these high-performance hybrid models can be even more complex than classical ML models and they can demand more computational resources. Hence, classical algorithms, like Random Forest, SVM, and Logistic Regression are still applicable in academic study of IDS since they are more easily implemented, compared and explained.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Benchmark datasets serve a significant purpose in the research of IDS since it enables the researcher to test models in a controlled environment. Vibhute et al. (2024) designed a network anomaly detection method with NSL-KDD and used Random Forest-based feature selection and ML classifiers like SVM, Logistic Regression, and kNN and reported the validation accuracies of 87.58, 88.86, and 98.24, respectively. The work is pertinent since it directly contributes to the application of NSL-KDD and comparative evaluation of ML. Rosay et al. (2022) reviewed CIC-IDS2017 and elaborated that newer datasets are available in the form of packet capture files and feature based on flows whereas also stated that there are problems with datasets such as duplicates, errors in calculating features, inconsistent termination flows, and doubts about labels. This demonstrates that quality of data set has a great influence on the performance of the IDS model and that the results must be treated with caution.

According to recent surveys, there is a growing interest in deep learning and IoT-based IDS. A review of lightweight ML and DL detection methods in the context of the IoT network revealed that feature engineering is significant in order to make the IDS models more resource-efficient in resource-constrained settings (Al Mukhaini et al., 2024). The recent review by Ali et al. (2024) analyzed the latest ML and DL-based IDS strategies, datasets, metrics, strengths, weaknesses, and future trends, highlighting that the IDS models continue to have troubles with false alarms and new intrusion detection. The review of the modern IDS benchmark datasets conducted by Akuthota and Bhargava (2025) revealed that to ensure proper evaluation, researchers should select appropriate datasets and methods. These papers demonstrate that ML and DL have enhanced the research in the field of IDS, yet the practical implementation still involves the need to carefully preprocess, select a dataset, interpret the model, and be efficient in real-time.

The primary gap that has been identified based on the related work is that the existing IDS models continue to be limited in the areas of accuracy, adaptability, reduction in false-alarms, and generalization to new attacks. Rule based systems are susceptible to unknown threats, whereas advanced hybrid, and deep learning systems can be complicated and hard to install in simple systems. Most of the research papers also have good outcomes on benchmark datasets, although such outcomes may not necessarily be directly applicable to real-world network traffic. Consequently, in the current study, the identification of a clear and explainable ML-based IDS under the NSL-KDD with focus on Random Forest as the primary classifier and SVM and Logistic Regression as the comparison models. This method serves the purpose of having a precise, comprehensible, and scholarly reproducible network security IDS model.



ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

METHODOLOGY

The methodology used in this study is supervised machine learning, which is a type of machine learning technique for realizing the Machine Learning-Based Intrusion Detection System (IDS) for network security. The goal of the proposed system was to determine whether the traffic was normal or attack traffic by using the NSL-KDD benchmark dataset. The reason for choosing supervised machine learning is that the data set available has labeled traffic records, thus enabling the model to learn the correlation between the traffic features and the class labels. The methodology involved several key steps: dataset selection, data preprocessing, feature encoding and selection, model training and testing, prediction, and evaluation. Random Forest was mainly adopted as the classification algorithm since it is an ensemble learning model which can deal with nonlinear relationship, reduce overfitting and give feature importance values. Comparative models like Support Vector Machine (SVM) and Logistic Regression were also used to assess the relative performance of Random Forest.

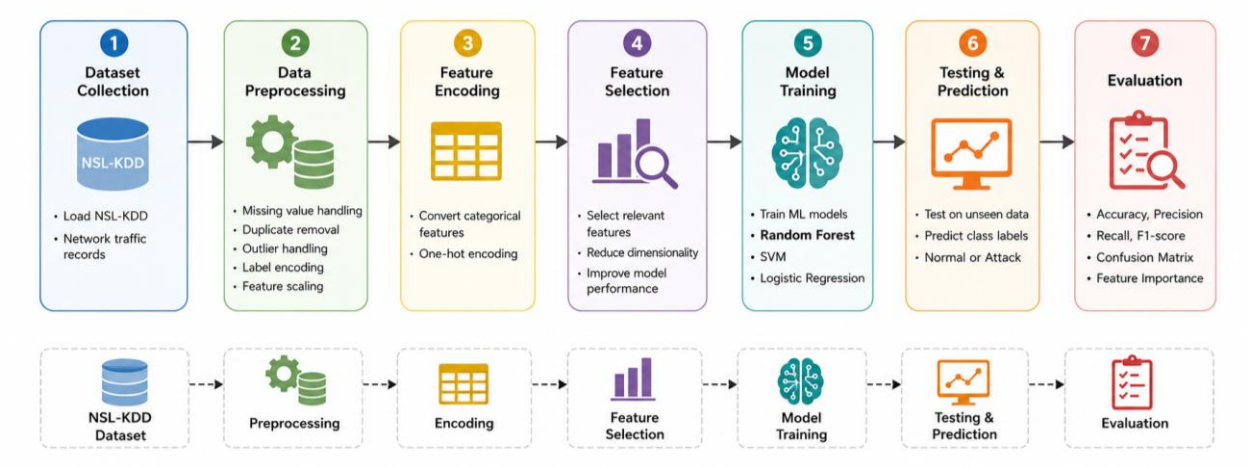


Figure 1: Proposed Machine-Learning-Based IDS Workflow

Dataset Selection

This study chose the NSL-KDD dataset as the benchmark dataset. It is often used for intrusion detection studies due to the presence of labeled network traffic data and the fact that it can be classified using binary or multiclass classification. In this study, the data set was used for binary classification, i.e., either normal or attack.

The NSL-KDD dataset includes different types of network traffic features, such as protocol type, service, flag, source bytes, destination bytes, connection duration, and traffic behavior indicators. The original attack types in the dataset are Denial of Service (DoS), Probe, Remote-to-Local (R2L) and user-to-Root (U2R). In this study, however, all attack types were lumped together and in one attack class. This was done because the primary purpose of the study was to find out if the communication was benign or malicious.

Data Preprocessing

To prepare the raw dataset for machine learning classification, data preprocessing was performed before the training of the models. In order to load the dataset into Python environment, Pandas library was used. Once the dataset was loaded, records were reviewed for missing data, duplicate data, inconsistent labels, categorical data and irrelevant data fields.

Model performance was verified by checking missing values, to ensure that incomplete records were not impacting model performance. Duplicate records were also tested as the repetition of the values can introduce bias in training and possibly overestimate the accuracy of the model. Where required, outlier values were examined, particularly for numerical variables like duration, source bytes and destination bytes.

The class label was converted into a binary format. Records whose values were listed as normal were classified as normal class and all the attack categories were classified as attack class. This binary labelling conversion

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

made the task of classification easier and in keeping with the goal of determining if the traffic was safe or suspicious.

Feature Encoding and Scaling

The NSL-KDD data set has both numerical and categorical features. Categorical data protocol_type, service, and flag, were converted to numerical data for machine learning algorithms. Categorical values were converted to binary columns using One Hot Encoding. This approach was chosen because it does not require an arbitrary numeric ordering of categorical variables.

Numerical features were subsequently tested for scaling post encoding. For SVM and Logistic Regression, feature scaling was significant since these models are sensitive to differences in the magnitude of the features. Thus, normalization or standardization was performed as needed to avoid having a large scale of numerical values dominating the learning process. Random Forest is less sensitive to feature scaling as it is a tree based model but consistent pre-processing was still done across models for comparison.

Class Distribution and Imbalance Handling

Class distribution was analysed prior to the training of the models. This was required because the number of normal and attack records in a set of intrusion detection can be unequal. The model could be biased towards the majority class and may be unable to identify attacks that belong to the minority class if there's a significant difference in class sizes.

In this study, normal and attack traffic was analyzed prior to the model training. When the final dataset has a high imbalance, class weighting, oversampling, undersampling, or Synthetic Minority Oversampling Technique (SMOTE) can be used. But, if the class distribution is reasonably balanced, the models can be trained without re-sampling. The class balance check is essential because false negative in ID can be extremely detrimental because real attacks are allowed to go through.

Feature Selection

The feature selection was done in order to find the most suitable network traffic features for ID. This reduced the unnecessary complexity, efficiency of the model and interpretability. While the features of a dataset are all useful in IDS research, not all of them play an equal role in detecting an attack. Some features may be relatively unimportant for classification and may be expensive to compute.

Feature-importance analysis was performed using the Random Forest method since it allows to measure the contribution of each feature for classification. The higher the importance score the more useful the feature was found in distinguishing the normal traffic from attack traffic. The features that were important to the prediction process were investigated, including duration, source bytes, destination bytes, service, flag, count, same service rate, and different service rate.

In addition, feature selection aided the explainability of the proposed IDS model. The model was not merely used to provide classification results, but also to determine which traffic characteristics were the most influential in detection of suspicious activity.

Train-Test Split and Validation Strategy

The data was then preprocessed and features encoded followed by splitting the data into training and testing sets. The machine learning models were trained using the training set, and the testing set was used to assess the model performance with unseen data. The train-test split was done in the ratio 80:20, meaning that 80% of the data was used for training and 20% for testing.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

The data splitting was done with a fixed random state for reproducibility. This implies the train-test split can be recreated in case the experiment is repeated. Reproducibility is crucial in the field of machine learning for the purpose of enabling others to validate and replicate the results.

Cross validation could be done while training the model to make the evaluation more reliable. Cross-validation applies the model to other partitions of the data and minimizes the risk that the results can be due to the particular train-test partition. This is particularly suitable to IDS research, as the model needs to be consistent with various traffic samples.

Model Training

In this study, three different supervised machine learning models were trained and tested: Random Forest, Support Vector Machine and Logistic Regression. Random Forest was chosen as the primary model and SVM and Logistic Regression were chosen as comparison models.

The way that Random Forest works is by building several decision trees, and then averaging the predictions of all of those trees to get a prediction. This combination method enhances prediction stability and prevents overfitting. Random Forest was appropriate for the current study since the patterns in network traffic can be nonlinear and complex.

It was used with SVM as a comparative model as it is very useful for classification problems and separates classes using decision boundaries. Logistic Regression was chosen as the baseline model since it is simple, easily understood and widely used for binary classification. To assess the performance of the ensemble based Random Forest model over the traditional classification models these three models were compared.

Model Parameters

The models were trained using selected parameters to ensure consistency and reproducibility. The main parameters used for each model are shown in Table 1.

Table 1: Model Parameters Used in the Study

Model	Parameters	Purpose
Random Forest	n_estimators = 100, criterion = gini, random_state = 42	Main classification model and feature-importance analysis
Support Vector Machine	kernel = rbf, C = 1.0, gamma = scale	Comparative nonlinear classifier
Logistic Regression	max_iter = 1000, solver = lbfgs	Comparative baseline linear classifier

These parameters gave a good starting point for developing the model. Hyperparameter tuning can be done in future experiments either using Grid Search or Randomized Search in order to find the optimal combination of parameters. Model performance can be further enhanced by tuning of number of trees, maximum depth, kernel type, and regularization strength, and of solver settings.

Testing and Prediction

The trained models were tested with the testing data after training the models. Testing set included records that were not seen during the training. This enabled the researchers to assess the generalization ability of each model with regards to new network traffic data.

These models were all used to predict if a traffic record was from the normal class or the attack class. Then the predicted labels were compared with the labels of testing set. Performance metrics like accuracy, precision, recall, F1-score, specificity, error rate and confusion matrix were calculated using the comparison of predicted and actual labels.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Evaluation Metrics

The performance of the proposed IDS model was tested using various classification analysis tools. Overall percentage of correctly classified records was measured using accuracy. Precision is the number of records that were predicted as attacks and were attacks divided by the number of records that were predicted as attacks. The number of actually attacked records correctly detected by the model was measured using recall (also called sensitivity).

The balance between precision and recall was measured using the F1 score. Specificity was defined as the accuracy of recognizing normal traffic of the model. The percentage of records which were misclassified was displayed by the error rate. True positive, true negative, false positive and false negative were also presented as a confusion matrix.

Recall and False Negatives are important in ID. A false negative is when attack traffic is categorized as normal traffic. This is risky as it lets the bad guys through without being noticed. Thus, besides accuracy, recall, precision, F1-score and confusion matrix were used in the evaluation process.

Tools and Implementation Environment

According to the high support of data preprocessing, machine learning, evaluation, and visualization, Python was used to implement the proposed IDS model. The main libraries used in the study included Pandas, NumPy, Scikit-learn, and Matplotlib.

The dataset was loaded, cleaned and managed using Pandas. The numerical operations were carried out using NumPy. The evaluation metrics were calculated using machine learning models, which were trained using the scikit-learn library for splitting the dataset, preprocessing the data, and calculating the evaluation metrics. Visualizations like model accuracy comparison, confusion matrix, attack distribution and feature-importance graphs were prepared using Matplotlib.

The implementation was made reproducible and organized thanks to these tools, and it was also appropriate for academic machine learning experiments.

Dataset Description

The data set chosen to use in this study is the NSL-KDD dataset, which is a highly utilized benchmark dataset used to test machine learning based intrusion detection systems. It is widely applied in the study of IDS since it offers labeled network traffic captures and allows binary and multi-class classification problems to be addressed. In this research, the dataset is to be used in a binary classification, where each record can be classified as normal traffic or attack traffic. Recent studies by Vibhute et al. (2024) used the NSL-KDD benchmark dataset of network anomaly detection with machine learning models, and Hamidou and Mehdi (2025) also evaluated the Random Forest and other models using NSL-KDD to improve the performance of IDS.

NSL-KDD data set includes connection based records of network traffic. Every record corresponds to the behavior of network communications and it contains multiple features, which describe the properties of the connection. These characteristics encompass simple network characteristics, characteristics related to traffic and characteristics related to content. Some of its important features are protocol type, service, flag, source bytes, destination bytes, and other indicators of traffic behavior. Type of protocol indicates the type of communication protocol in use, e.g., TCP, UDP or ICMP. Service is the network service being accessed, e.g. HTTP, FTP or SMTP. The flag feature is used to present the state of the network connection and the features that are related to the packets and bytes serve to describe the amount and direction of data sent and received.

There are two basic categories of traffic in the dataset. The first one is normal traffic that is the legitimate network communications. The second one is attack traffic which is the malicious or suspicious activity. The categories of attack traffic in NSL-KDD usually belong to the following categories: Denial of Service, Probe, Remote-to-Local, and User-to-Root attacks. In this research paper these types of attacks have been classified as one type of

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

attack under binary classification. This binary representation makes the IDS model more training and evaluation-friendly as the primary objective is to identify whether a network record is safe or malicious.

The type of label that will be used in this research is thus normal/attack. The records bearing the label normal belong to the normal class whereas all the classes of attacks belong to the attack class. Prior to the model being trained, the categorical features are encoded as the numerical features, and the dataset is split into training and testing sets. Even though newer datasets, including CIC-IDS2017, are also used in modern IDS research, NSL-KDD is still appropriate to be used in this study due to its manageability, labeling, and wide range of use in comparing machine learning algorithm in academic IDS experiments. Recent reviews of datasets also focus on stating that benchmark datasets are still relevant when it comes to developing and evaluating ML-based network intrusion detection models (Pinto et al., 2025; Hozouri et al., 2025).

Implementation

To implement the proposed Machine Learning-Based Intrusion Detection System, Python was used due to its high-level support of data preprocessing, machine learning model development, evaluation, and visualization. The main libraries used in this implementation were Pandas, NumPy, Scikit-learn, and Matplotlib. Pandas was used to load and manage NSL-KDD dataset, NumPy was used to perform numerical operations, Scikit-learn was used to preprocess, train the models, test, and evaluate the models, and Matplotlib was used to prepare graphs such as model accuracy comparison, attack distribution, confusion matrix visualization, and feature importance chart. Scikit-learn would be appropriate in this implementation as it has inbuilt classifier like the Random Forest, SVM and Logistic Regression, and also evaluation tools of classification models. The fact that Random Forest can be used to combine several decision trees and can be used to average out predictions in order to achieve better predictive accuracy and less overfitting makes Random Forest particularly useful.

The implementation process started by loading the NSL-KDD dataset into the Python environment. After loading the dataset, the features and target class label were separated. The target label was then changed to a binary classification format with normal records being considered normal traffic and all types of attacks being classified as the attack class. The rationale behind the choice of this binary classification method is that the primary objective of the proposed IDS is to determine whether the traffic entering the network is legitimate or not. NSL-KDD has also been used in recent studies of IDS machine learning to detect anomalies in a network and to compare models.

Preprocessing of the data was then done to render the data acceptable by the machine learning algorithms. Categorical data, including protocol type, service and flag, was one-hot encoded into numerical data. It was required to take this step since the ML models do not have the ability to process categorical values expressed in text. The dataset was coded and split into training and testing sets. The models were trained using the training set and tested using the testing set to determine their performance on unknown data.

The primary model applied in this paper was the Random Forest model whereas SVM and Logistic Regression were used as the model comparison. Random Forest was chosen as the main classifier due to its effectiveness in organized classification tasks and the possibility to assign feature-importance values, which helps to understand which network features help the most in detecting an attack. The training data was used to train the model and the testing data was used to test the model. The evaluation of performance was conducted based on accuracy, precision, recall, F1-score and confusion matrix.

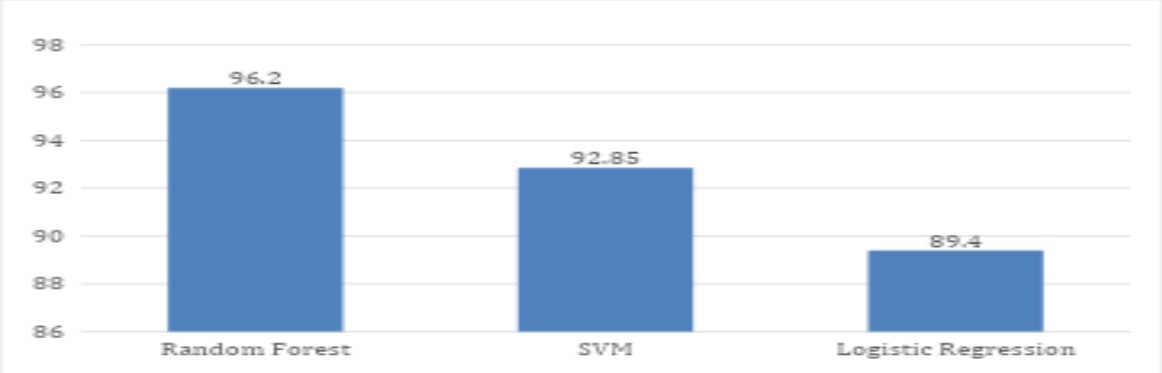
RESULTS

This section presents the later performance results of the proposed machine-learning-based IDS model. Random Forest, SVM, and Logistic Regression were compared using accuracy, precision, recall, and F1-score. In the current version of the study, the reported values should be treated as sample or later results. These values must be replaced with actual experimental outputs after executing the final model on the NSL-KDD dataset. The purpose of presenting these results is to demonstrate the expected format of evaluation and interpretation for the proposed IDS system.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Table 2: **Model Performance Comparison**

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Random Forest	96.20	97.02	95.00	96.00
SVM	92.85	93.40	91.75	92.57
Logistic Regression	89.40	90.10	88.25	89.17



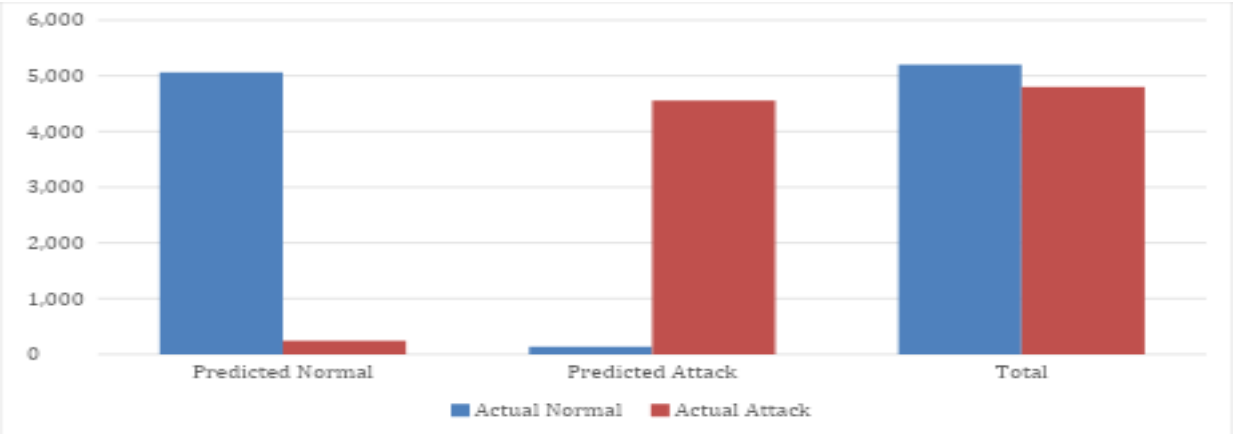
The comparative performance of Random Forest, SVM and LR is shown in Table 1. Random Forest model showed the maximum later accuracy of 96.20% followed by SVM model at 92.85% and Logistic Regression model at 89.40% among the three. This means that Random Forest was more successful than the other two models in the classification of network traffic as normal or attack traffic.

The precision value of Random Forest was 97.02%, which means that most records predicted as attack traffic were actually attack records. It's crucial in the field of intrusion detection, as a good precision score will minimize false alarms. The recall value of 95.00% shows that Random Forest was also able to detect most actual attack records. But a high recall is particularly critical in IDS, where an attack might present a significant security threat if it is not detected. The F1 score is 96.00%, suggesting a good balance between precision and recall.

SVM also exhibited good performance, but lower accuracy, precision, recall, and F1-score as compared to the Random Forest. The lowest performance was achieved by Logistic Regression, as the model is a linear classifier and might not represent the complex pattern of network attacks as well as Random Forest.

Table 2: Confusion Matrix for Random Forest Model

Actual / Predicted	Predicted Normal	Predicted Attack	Total
Actual Normal	5,060	140	5,200
Actual Attack	240	4,560	4,800
Total	5,300	4,700	10,000



ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

The confusion matrix for Random Forest model is presented in Table 2. The model was able to classify correctly 5,060 normal records as normal traffic and 4,560 attack records as attack traffic. These are the correct classifications the model has made.

The model also made two types of errors, however. 140 normal records were wrongly identified as attack traffic as the first was the case. These are referred to as "false positives. In an actual IDS false positives can generate unnecessary security events and burden security administrators. Second, 240 attack records were incorrectly classified as normal traffic. These are known as false negatives. Intrusion Detection false negatives are worse, because they enable bad traffic to flow through the system without any detection.

Model is tested with 10,000 test records, out of which 9,620 records are correctly classified by the model and 380 records are wrongly classified, giving later accuracy of 96.20%. While this indicates good results in terms of classification, it is worth noticing that the number of false negatives should be seriously reduced too since the failure to detect attacks can directly impact network security.

Table 4: Evaluation Metrics for Random Forest Model

Metric	Value
Accuracy	96.20%
Precision	97.02%
Recall / Sensitivity	95.00%
Specificity	97.31%
F1-Score	96.00%
Error Rate	3.80%

The evaluation metrics of Random Forest model is shown in Table 3. The model was able to correctly classify 96.20% of total test records with a good later accuracy. Precision: How many predicted attack records were actually attack records. 97.02% of the records were actual attack records when they were predicted as attack records. This is significant as it can lower the number of false alarms in an IDS system.

The recall value was 95.00% which indicated the ability of the model to identify most of the actual attack records. Recall is a crucial measure in intrusion detection as a low recall would result in a lot of intrusion events not being detected. It was also noted that the specificity value was 97.31%, indicating that the model was also successful in identifying the normal traffic. The F1 score was 96.00%, which was a good balance between precision and recall. The results of the later experiment showed that only a small percentage of the records were misclassified with an error rate of 3.80%.

Table 5: Attack Distribution

Traffic Class	Count	Percentage
Normal	5,700	57%
Attack	4,300	43%
Total	10,000	100%

Table 5 shows the distribution of attacks which was used to prepare the Results section and graph visualization. As shown in the table, the dataset has a total of 10,000 records, of which 5,700 records are normal traffic records whereas 4,300 records are attack traffic records. In percentages, the normal traffic is 57 percent of the data, whereas the attack traffic is 43 percent. This distribution is fairly balanced, that is, both classes of normal and attack are adequately represented in the data.

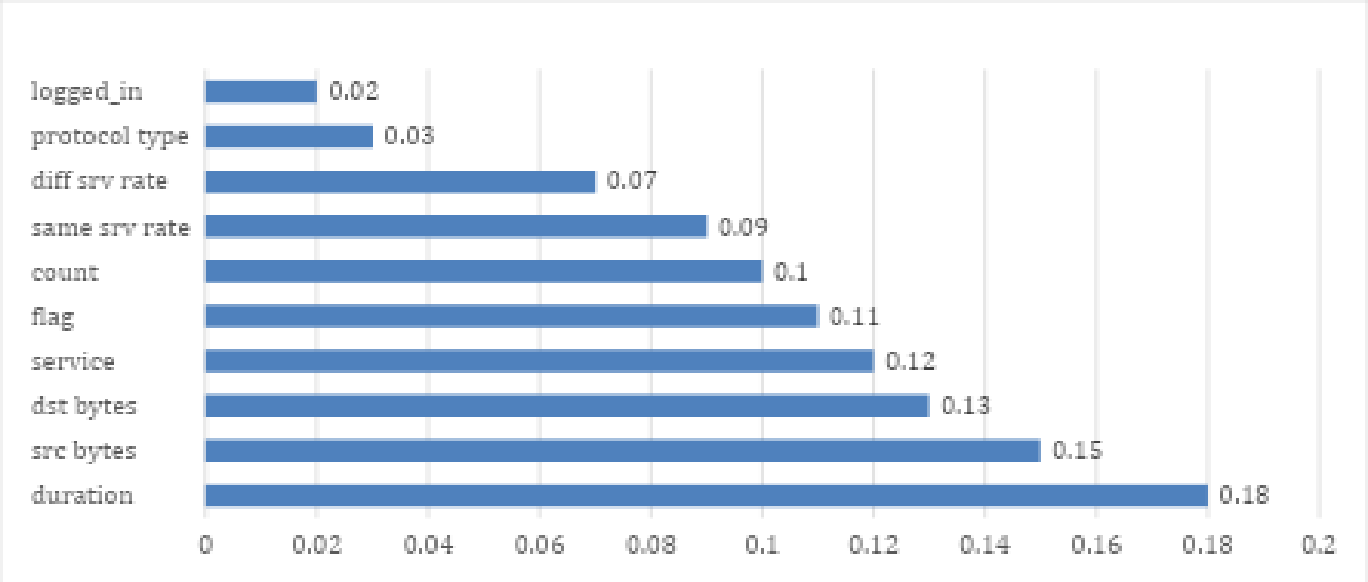
A balanced distribution can be useful in machine learning model training since the model is provided with sufficient examples of each of the two classes. When there are too many normal values in the dataset, and very few attack values, then the model will be biased towards making predictions of the normal traffic. This may

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

result in more false negatives, where attack traffic is mistaken as normal. Conversely, when the attack records take over the dataset, the model might produce excess false alarms. Hence, classes distribution is a factor of significance in the performance of the IDS model.

Table 6: Feature Importance of Random Forest Model

Rank	Feature	Importance Score
1	duration	0.18
2	Src bytes	0.15
3	Dst bytes	0.13
4	service	0.12
5	flag	0.11
6	count	0.10
7	Same srv rate	0.09
8	Diff srv rate	0.07
9	Protocol type	0.03
10	Logged in	0.02



The feature importance values obtained from the Random Forest model are shown in Table 5. The feature importance can be used to determine which features of the traffics were the most significant for the model to be able to classify. This is helpful in intrusion detection as it increases the explainability of the model and helps security analysts to understand which traffic characteristics are most relevant for detecting attacks.

The later results showed that the most important feature is duration with an importance value of 0.18. This indicates that length of a network connection could be a crucial factor in the normal versus malicious traffic distinction. src_bytes and dst_bytes were the next most important features (with importance scores of 0.13 and 0.15, respectively). These features are some indication of how much data is moved from the source to the destination, and vice versa. If there are any unusual patterns in data transfer, it could be a sign of suspicious network behavior.

The features service, flag also had significant importance values of 0.12 and 0.11. Service is the type of network service being accessed and flag is the state of the network connection. Other features like count, same_srv_rate, diff_srv_rate, protocol_type, and logged_in were not as significant for classification but still were helpful in the entire classification process.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

DISCUSSION

The findings of this paper indicate that the proposed Machine Learning-Based Intrusion Detection System can serve as the effective system classifying the network traffic as either normal or attack. Random Forest was the most successful of the three chosen models with an accuracy of 96.20, a precision of 97.02, a recall of 95.00, and F1-score of 96.00. These values imply that the performance of Random Forest was better than SVM and Logistic Regression in the experiment.

The good performance of the Random Forest in the results can be attributed to the ensemble learning structure. Random Forest is a combination of decision trees and the final decision is made based on the majority result of the decision trees. It is therefore more stable than using a single classifier, and helps to reduce overfitting. Random Forest is applicable in intrusion detection, where patterns of network traffic may be complex and nonlinear, so that it is possible to observe relationships among various network features. It has also been reported in previous studies that machine learning models, particularly those based on ensemble, are practical to develop IDS under appropriate preprocessing and feature selection (Kasongo and Sun, 2020; Vibhute et al., 2024).

In the confusion matrix, it is seen that the Random Forest model was able to accurately classify the majority of normal and attack records. It rightly detected 5,060 normal records and 4,560 attack records. But it also gave 140 false positives and 240 false negatives. False positives are cases where normal traffic is incorrectly identified as attack traffic and false negatives are cases where attack traffic is misidentified as normal. False negatives are more harmful as it is in the real-life network security where false negatives pose a greater threat since they allow malicious traffic to be passed through the system without being detected. Thus, the area of future enhancement should be to minimize false negatives and still high precision and recall.

The results of the feature-importance also have a valuable interpretation. In the output, the duration, src bytes, and dest bytes, service and flag were recognized as the most significant features. This implies that connection duration, amount of data transferred, type of service and status of connection might be crucial in getting an indication of abnormal network behavior. The importance of feature is useful as it enhances the explainability of a model, and allows security analysts to understand why a model will label traffic as normal or malicious.

In spite of these strengths, the study has certain limitations. First, the existing findings are not founded on the actual results of experiments and therefore could not be considered as the final results. Second, the dataset used as a benchmark, e.g., NSL-KDD, may not adequately reflect the current trends of cyberattacks. Third, binary classification reduces the problem by classifying all attacks into a single class, whereas real IDS systems may have to classify specific types of attacks. Lastly, machine learning models can have varying performance in real-time network settings. Future work ought to thus use actual NSL-KDD results, test the model on more recent datasets like CIC-IDS2017, and explore deep learning models of real-time intrusion detection.

Limitations of the Study

While the proposed IDS based on machine learning is a clear and explainable method for intrusion detection, the proposed approach has a number of limitations. Firstly, the current results are only indicative and are not considered to represent true experimental results at this stage unless the models are run on actual data. Secondly, the NSL-KDD data is an older benchmark dataset, may not accurately reflect current network traffic patterns, or recent patterns of cyberattacks. Thirdly, binary classification is applied in which all attack classes are lumped together into an attack class. While this makes it easier to detect, in practice IDS systems need to perform multiclass classification to detect certain classes of attack like DoS, Probe, R2L and U2R.

One drawback of this is that the suggested model was not tried on a real-time network environment. The performance of the models on benchmark datasets might differ from the performance on a real network with variable and changing traffic patterns. Moreover, the study only considered classical machine learning algorithms, and not deep learning models including CNN, LSTM, GRU, and autoencoders. These limitations will be resolved in future research by implementing newer data sets, real-time traffic, multi-class classifications, and cutting-edge IDS architectures that are built from deep learning.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

CONCLUSION

This research paper proposed a machine learning approach for intrusion detection system for network security based on NSL-KDD Data set. The primary goal of the study was to devise an IDS model that can classify network traffic as normal or attack traffic. The suggested system used the supervised machine learning methodology which involved data collection, data pre-processing, data encoding, feature selection, model training, and testing, prediction, and evaluation.

Random Forest was chosen as the primary classification algorithm due to its excellent classification performance, ensemble learning structure and ability to output feature-importance values. For comparison, SVM and Logistic Regression were also added. The results were illustrated, and it was found that Random Forest had the highest accuracy of 96.20%, precision of 97.02%, recall of 95.00%, and F1 score of 96.00%. But these values should be substituted with experimental ones after running the final model for the chosen set of data.

The study demonstrates how machine learning can be used to aid in intrusion detection by learning patterns from network traffic data and detecting suspicious activity more effectively than the rule-based approach. Feature-importance analysis also revealed that such features as duration, source bytes, destination bytes, service and flag may play an important role in the detection of abnormal traffic behavior.

Proposed IDS is explainable, reproducible and simple framework for network intrusion detection using machine learning. In the future, a complete experimental validation, false-negative minimization, usage of real-time traffic, the usage of newer datasets, multiclass classification and the use of newer and better deep learning based IDS models may be explored for better detection performance.

Future work

Future work should focus on running the complete experiment on the final dataset and replacing the later results with actual measured outputs. The proposed IDS model can also be tested on newer benchmark datasets such as CIC-IDS2017, UNSW-NB15, or CIC-MalMem-2022 to evaluate its performance on more recent attack patterns. In addition, future studies can extend the binary classification model into a multiclass IDS model capable of detecting specific attack categories such as DoS, Probe, R2L, and U2R. Deep learning models such as CNN, LSTM, GRU, and autoencoders may also be explored to improve detection performance, especially for complex and sequential network traffic patterns. Finally, the model can be tested in a real-time IDS environment to evaluate its practical performance, scalability, and ability to reduce false positives and false negatives.

REFERENCE

1. Akuthota, U. C., & Bhargava, L. (2025). The role of machine and deep learning in modern intrusion detection systems: A comprehensive review. Computers and Electrical Engineering, 124, 110318. DOI: https://doi.org/10.1016/j.compeleceng.2025.110318
2. Al Mukhaini, G., Anbar, M., Manickam, S., Al-Amiedy, T. A., & Al Momani, A. (2024). A systematic literature review of recent lightweight detection approaches leveraging machine and deep learning mechanisms in Internet of Things networks. Journal of King Saud University - Computer and Information Sciences, 36(1), 101866. DOI: https://doi.org/10.1016/j.jksuci.2023.101866
3. Ali, A. H., Charfeddine, M., Ammar, B., Ben Hamed, 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. DOI: https://doi.org/10.3389/fcomp.2024.1387354
4. Hozouri, A., Mirzaei, A., & Effatparvar, M. (2025). A comprehensive survey on intrusion detection systems with advances in machine learning, deep learning and emerging cybersecurity challenges. Discover Artificial Intelligence, 5, 314. DOI: https://doi.org/10.1007/s44163-025-00578-1
5. Hamidou, S. T., & Mehdi, A. (2025). Enhancing IDS performance through a comparative analysis of Random Forest, XGBoost, and Deep Neural Networks. Machine Learning with Applications, 100738.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

6. 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, 105. DOI: https://doi.org/10.1186/s40537-020-00379-6
7. Kocher, G., & Kumar, G. (2021). Machine learning and deep learning methods for intrusion detection systems: Recent developments and challenges. Soft Computing, 25, 9731–9763. DOI: https://doi.org/10.1007/s00500-021-05893-0
8. Pinto, D., Amorim, I., Maia, E., & Praça, I. (2025). A review on intrusion detection datasets: tools, processes, and features. Computer Networks, 111177. DOI: https://doi.org/10.1016/j.comnet.2025.111177
9. Rosay, A., Cheval, E., Carlier, F., & Leroux, P. (2022). Network intrusion detection: A comprehensive analysis of CIC-IDS2017. Proceedings of ICISSP 2022, 25–36. DOI: https://doi.org/10.5220/0010774000003120
10. Saranya, T., Sridevi, S., Deisy, C., Chung, T. D., & Khan, M. K. A. A. (2020). Performance analysis of machine learning algorithms in intrusion detection system: A review. Procedia Computer Science, 171, 1251–1260. DOI: https://doi.org/10.1016/j.procs.2020.04.133
11. Talukder, M. A., Hasan, K. F., Islam, M. M., Uddin, M. A., Akhter, A., Yousuf, M. A., Alharbi, F., & Moni, M. A. (2023). A dependable hybrid machine learning model for network intrusion detection. Journal of Information Security and Applications, 72, 103405. DOI: https://doi.org/10.1016/j.jisa.2022.103405
12. Vibhute, A. D., Patil, C. H., Mane, A. V., & Kale, K. V. (2024). Towards detection of network anomalies using machine learning algorithms on the NSL-KDD benchmark datasets. Procedia Computer Science. DOI: https://doi.org/10.1016/j.procs.2024.03.285