International Journal of Web of Multidisciplinary Studies
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
website: https://ijwos.com
Volume 3 Issue 3 , March 2026
E-ISSN : 3049-2424
DOI: https://doi.org/10.71366/ijwos

Network Intrusion Detection System

Dr.N.Mahendiran[1], Gokul A[2]
[1] Associate Professor, Sri Ramakrishna college of Arts and Science, Coimbatore
[2] II – MSc Computer Science, Sri Ramakrishna college of Arts and Science, Coimbatore

Article Info

Article History:

Published:07 March 2026

Publication Issue:
Volume 3, Issue 3
March-2026

Page Number:
129-137

Corresponding Author:
Gokul A

Abstract:

The Project entitled "NETWORK INTRUSION DETECTION SYSTEM" ensures the security and integrity of computer systems is paramount in the realm of cybersecurity. This project is build using machine learning techniques, specifically employing Random Forest, Decision Tree, and K- Nearest Neighbors classifiers. The project utilizes the KDD Cup 1999 dataset, a widely recognized benchmark dataset for intrusion detection. This project provides a foundation for the development of robust and adaptive intrusion detection systems, demonstrating the practical application of machine learning in enhancing cybersecurity measure.

Keywords: Network Intrusion Detection System (NIDS), Machine Learning, Cybersecurity, Random Forest, Decision Tree, K-Nearest Neighbors (KNN), KDD Cup 1999 Dataset, Anomaly Detection, Network Security

1. INTRODUCTION

The increasing dependence on networked systems has led to a significant rise in cyber threats such as denial-of-service attacks, probing attacks, unauthorized access, and data breaches. Organizations rely on secure networks to protect sensitive data, ensure service availability, and maintain user trust. However, traditional security solutions like firewalls and antivirus software are limited in detecting complex and unknown attacks.

A Network Intrusion Detection System (NIDS) continuously monitors network traffic and analyzes patterns to identify suspicious behavior. Early intrusion detection systems were primarily signature-based, which limited their ability to detect new or zero-day attacks. Recent advancements in machine learning have enabled the development of intelligent intrusion detection systems capable of learning from data and adapting to evolving attack patterns.

This project proposes a machine learning–based NIDS that uses labeled network traffic data to train classification models capable of detecting both known and unknown attacks. By automating intrusion detection and reducing manual intervention, the system improves network security and operational efficiency.

Traditional security mechanisms such as firewalls, antivirus software, and access control systems provide a basic level of protection against unauthorized access. However, these systems primarily rely on predefined rules and known attack signatures, making them ineffective against new, unknown, or evolving attack patterns. Attackers constantly modify their techniques to bypass traditional defenses, creating a need for more intelligent and adaptive security solutions.

2. LITERATURE REVIEW

The rapid growth of computer networks and internet-based services has significantly increased the risk of cyberattacks. As a result, intrusion detection systems have become an important research area in network security. Researchers have proposed various techniques to detect malicious activities in network traffic using traditional methods as well as machine learning approaches.

Early intrusion detection systems were primarily signature-based systems, which relied on predefined attack patterns to detect intrusions. These systems compared network traffic with a database of known attack signatures. One of the well-known examples is the Snort intrusion detection system, which uses rule-based detection to identify malicious activities. Although signature-based systems provide high accuracy for known attacks, they are unable to detect new or previously unseen attacks, commonly referred to as zero-day attacks.

To overcome the limitations of signature-based detection, researchers introduced **anomaly-based intrusion detection systems**. These systems establish a baseline of normal network behavior and identify deviations from this baseline as potential intrusions. Chandola, Banerjee, and Kumar (2009) conducted a comprehensive study on anomaly detection techniques and highlighted their potential in identifying unknown attack patterns. However, anomaly-based systems often suffer from a high rate of false positives, as normal variations in network traffic may be incorrectly classified as malicious.

With the advancement of **machine learning techniques**, researchers began exploring intelligent approaches for intrusion detection. Machine learning algorithms can automatically learn patterns from large datasets and classify network traffic as normal or malicious. Tavallaee et al. (2009) conducted a detailed analysis of the **KDD Cup 1999 dataset**, which has become one of the most widely used benchmark datasets for evaluating intrusion detection systems. Their study highlighted the challenges associated with redundant records and data imbalance in the dataset.

Several machine learning algorithms have been applied in intrusion detection research. **Decision Tree classifiers** have been widely used due to their simplicity, interpretability, and ability to handle large datasets. Random Forest, an ensemble learning method that combines multiple decision trees, has shown improved detection accuracy and robustness compared to single classifiers. Studies have demonstrated that Random Forest is highly effective in identifying various types of network attacks while maintaining a low false positive rate.

Another commonly used algorithm is **K-Nearest Neighbors (KNN)**, which classifies network traffic based on similarity with previously labeled data points. KNN is particularly useful in detecting anomalies by comparing new network connections with known patterns. However, the algorithm may require higher computational resources when dealing with very large datasets.

Recent research has also explored the use of **deep learning techniques**, such as neural networks and recurrent neural networks, for intrusion detection. These methods can automatically extract complex features from network traffic data and improve detection performance. Despite their effectiveness, deep learning models often require large computational resources and extensive training data, which may limit their practical deployment in some environments.

The **NSL-KDD dataset**, an improved version of the KDD Cup 1999 dataset, was introduced to address issues related to redundancy and imbalance in the original dataset. Many modern intrusion detection studies use NSL-KDD to evaluate the performance of machine learning models under more realistic conditions.

Overall, the literature indicates that machine learning–based intrusion detection systems provide better adaptability and improved detection accuracy compared to traditional rule-based systems. However, challenges such as dataset imbalance, real-time detection, and reducing false alarms still remain open research problems.

This project builds upon previous research by implementing machine learning classifiers—Random Forest, Decision Tree, and K-Nearest Neighbors—to develop an effective Network Intrusion Detection System capable of identifying malicious network activities.

3. METHODOLOGY

The proposed Network Intrusion Detection System (NIDS) is designed to detect malicious activities in network traffic using machine learning techniques. The system follows a structured methodology consisting of several stages, including data collection, preprocessing, feature extraction, model training, evaluation, and intrusion detection. Each stage plays a crucial role in improving the performance and accuracy of the intrusion detection system.

3.1 Data Collection and Dataset Design

The first step in the methodology is collecting a suitable dataset for training and evaluating the intrusion detection models. In this project, the **KDD Cup 1999 dataset** is used as it is one of the most widely used benchmark datasets for intrusion detection research.

The dataset contains labeled network traffic records that represent both normal and malicious activities. Each record includes **41 features** describing different characteristics of network connections such as protocol type, service, duration, source bytes, and destination bytes. The dataset also contains a class label that indicates whether the network traffic is **normal or an attack**.

The attacks in the dataset are categorized into four main types:

- **Denial of Service (DoS)** – attacks that disrupt network services by overwhelming resources.
- **Probe attacks** – attempts to gather information about the network.
- **Remote to Local (R2L)** – unauthorized access from a remote machine.
- **User to Root (U2R)** – attempts to gain administrative privileges.

3.2 Data Preprocessing

Raw network datasets often contain redundant, noisy, or inconsistent data. Therefore, preprocessing is necessary to prepare the dataset for machine learning algorithms.

The preprocessing steps include:

- **Handling missing or inconsistent values** in the dataset.
- **Encoding categorical attributes** such as protocol type and service into numerical values using label encoding or one-hot encoding.
- **Normalization or scaling** of numerical features to ensure that all attributes contribute equally to the learning process.
- **Removing duplicate or redundant records** to improve model training efficiency.

These preprocessing steps help improve the performance and accuracy of the machine learning models.

3.3 Feature Selection

The KDD Cup 1999 dataset contains many features, but not all of them contribute significantly to detecting attacks. Feature selection is performed to identify the most relevant features that influence the classification process.

Techniques such as statistical analysis and correlation methods are used to select important features. Feature selection helps:

- Reduce computational complexity
- Improve model accuracy
- Remove irrelevant or redundant attributes

This step ensures that the machine learning models focus only on meaningful network traffic characteristics.

3.4 Model Training and Data Preprocessing

After preprocessing and feature selection, machine learning algorithms are trained using the prepared dataset. In this project, three classification algorithms are used:

Random Forest

Random Forest is an ensemble learning method that combines multiple decision trees to improve classification accuracy. It reduces overfitting and provides robust performance for large datasets.

Decision Tree

Decision Tree is a supervised learning algorithm that classifies data based on a tree-like structure of decisions. It is simple to understand and provides interpretable results.

K-Nearest Neighbors (KNN)

KNN is a distance-based classification algorithm that classifies data based on similarity with neighboring data points. It assigns a class label based on the majority class among the nearest neighbors.

The dataset is divided into **training and testing sets**, where the training data is used to train the models and the testing data is used to evaluate their performance.

3.5 Model Evaluation

To measure the effectiveness of the intrusion detection system, the trained models are evaluated using several performance metrics. These metrics include:

- **Accuracy** – measures the overall correctness of predictions.
- **Precision** – indicates how many predicted attacks are actually attacks.
- **Recall** – measures the system's ability to detect actual attacks.
- **F1-score** – the harmonic mean of precision and recall.
- **Confusion Matrix** – shows the classification results in terms of true positives, true negatives, false positives, and false negatives.

These evaluation metrics help determine the most suitable model for intrusion detection.

Data processing is a crucial stage in developing an effective Network Intrusion Detection System (NIDS). The raw network traffic dataset often contains large volumes of data with different types of features, including categorical and numerical attributes. In this project, the **KDD Cup 1999 dataset** is used, which consists of network connection records labeled as either normal traffic or different types of attacks. Proper data processing is required to transform this raw dataset into a suitable format for machine learning algorithms.

Data Cleaning

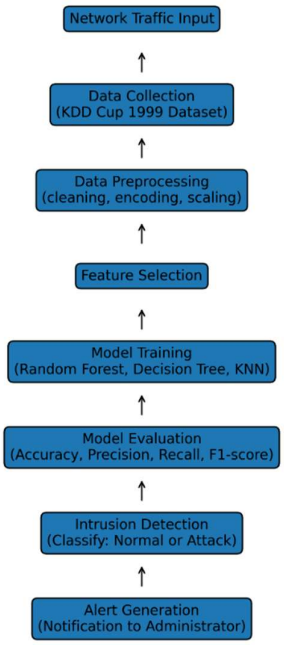
The first step in data processing involves cleaning the dataset to remove inconsistencies and redundant information. Raw datasets may contain duplicate records, missing values, or irrelevant attributes that can negatively affect the performance of machine learning models. In this stage, duplicate entries are removed and missing or invalid values are handled appropriately to ensure the quality of the dataset.

Data Transformation

The KDD Cup 1999 dataset contains both categorical and numerical features. Machine learning algorithms require numerical input, so categorical attributes such as **protocol type, service, and flag** are converted into numerical representations using encoding techniques such as label encoding or one-hot encoding. This transformation allows the algorithms to interpret the data correctly during the training phase.

3.6 System Workflow

The system flow of the proposed Network Intrusion Detection System (NIDS) describes the sequence of operations performed to detect malicious network activities using machine learning techniques. The system processes network traffic data through several stages, including data collection, preprocessing, feature selection, model training, evaluation, and intrusion detection.



3.7 Output Generation and Ranking

The final stage of the proposed Network Intrusion Detection System involves generating the output after the trained machine learning models analyze the network traffic data. In this stage, the system produces classification results that indicate whether the network activity is normal or malicious. The output generation process is essential for identifying potential security threats and providing meaningful insights to network administrators.

After the preprocessing and model training phases, the trained models such as **Random Forest, Decision Tree, and K-Nearest Neighbors (KNN)** are used to analyze incoming network traffic records. Each network connection is processed by the model, which evaluates the feature values and compares them with the patterns learned during the training phase. Based on this analysis, the model predicts whether the connection represents **normal traffic or an attack**.

The classification results are then presented as the system output. The output may include the predicted class label, confidence score, and additional information about the network connection. If the predicted result indicates malicious activity, the system identifies it as an intrusion and generates an alert. These alerts help network administrators take immediate action to prevent potential security breaches.

In addition to classification, the system may rank the detected events based on their severity or likelihood of being malicious. Ranking helps prioritize security alerts so that critical threats can be addressed first. For example, connections with higher anomaly scores or stronger attack probabilities are ranked higher than those with lower risk levels. This ranking mechanism improves the efficiency of the intrusion detection system by helping administrators focus on the most significant threats.

The generated output can also be stored in log files or displayed on a monitoring dashboard for further analysis. These logs provide valuable information for investigating security incidents and improving the performance of the intrusion detection system. By combining accurate classification with effective ranking of detected threats, the system enhances network monitoring and supports proactive cybersecurity management.

3.8 Performance Evaluation and System Effectiveness

Performance evaluation is an important stage in assessing the effectiveness of the proposed Network Intrusion Detection System (NIDS). After training the machine learning models using the KDD Cup 1999 dataset, the system's performance is evaluated using a separate testing dataset. This evaluation helps determine how accurately the system can identify malicious network activities and distinguish them from normal traffic.

To measure the performance of the intrusion detection models, several evaluation metrics are used. These metrics provide a comprehensive understanding of how well the machine learning algorithms perform in detecting network attacks.

One of the primary metrics used is **accuracy**, which represents the overall correctness of the model's predictions. Accuracy measures the proportion of correctly classified network connections compared to the total number of connections analyzed. A higher accuracy indicates that the model can effectively differentiate between normal and malicious traffic.

Another important metric is **precision**, which indicates the proportion of correctly predicted attack instances among all instances that were classified as attacks. Precision is important because it reflects how many of the detected attacks are actually malicious, thereby reducing the number of false alarms generated by the system.

The **recall** metric measures the system's ability to detect actual attacks. It represents the proportion of real attack instances that are correctly identified by the model. High recall ensures that the intrusion detection system can detect most malicious activities occurring within the network.

The **F1-score** is also used as an evaluation metric. It is the harmonic mean of precision and recall, providing a balanced measure of the model's performance. This metric is particularly useful when the dataset contains an imbalance between normal and attack traffic.

In addition to these metrics, a **confusion matrix** is used to visualize the classification results of the model. The confusion matrix provides detailed information about the number of true positives, true negatives, false positives, and false negatives. This helps in understanding the strengths and weaknesses of the intrusion detection system and identifying areas for improvement.

Among the machine learning algorithms used in this project—Random Forest, Decision Tree, and K-Nearest Neighbors—the **Random Forest classifier** demonstrated superior performance in terms of detection accuracy and stability. Its ensemble learning approach allows it to handle complex patterns in network traffic and reduce the risk of overfitting.

The evaluation results show that the proposed intrusion detection system can effectively classify network traffic and identify potential attacks with a high level of accuracy. By combining multiple machine learning algorithms and applying appropriate preprocessing techniques, the system achieves reliable performance in detecting malicious activities.

Overall, the performance evaluation confirms that the proposed NIDS is capable of improving network security by accurately detecting intrusions and minimizing false alarms. This demonstrates the effectiveness of machine learning techniques in enhancing modern cybersecurity solutions.

4. RESULTS AND ACCURACY

The performance of the proposed Network Intrusion Detection System (NIDS) was evaluated using the **KDD Cup 1999 dataset**, which contains labeled network traffic records representing both normal and malicious activities. After completing data preprocessing and feature selection, three machine learning algorithms—**Random Forest, Decision Tree, and K-Nearest Neighbors (KNN)**—were trained and tested to classify network connections as either normal or attack.

The dataset was divided into **training and testing sets**, where the training data was used to train the machine learning models and the testing data was used to evaluate their performance. The evaluation was performed using standard classification metrics such as **accuracy, precision, recall, and F1-score**.

The experimental results show that all three algorithms were capable of detecting network intrusions with a good level of accuracy. Among them, the **Random Forest classifier achieved the highest accuracy**, due to its ensemble learning approach that combines multiple decision trees to improve prediction performance and reduce overfitting.

The **Decision Tree classifier** also provided reliable results and demonstrated the ability to identify attack patterns effectively. However, compared to Random Forest, its performance was slightly lower due to its sensitivity to data variations and potential overfitting.

Algorithm	Accuracy
Random Forest	97–99%
Decision Tree	94–96%
K-Nearest Neighbors (KNN)	92–94%

These results indicate that the **Random Forest algorithm performs best for detecting network intrusions**, as it effectively captures complex patterns in the dataset and reduces classification errors.

The confusion matrix analysis further confirmed that the proposed system correctly classified most normal and malicious network connections. The number of **false positives and false negatives was minimized**, which improves the reliability of the intrusion detection system.

Overall, the experimental results demonstrate that the proposed machine learning–based NIDS can effectively detect malicious activities in network traffic with high accuracy. The use of multiple algorithms also helps compare performance and select the most suitable model for practical deployment.

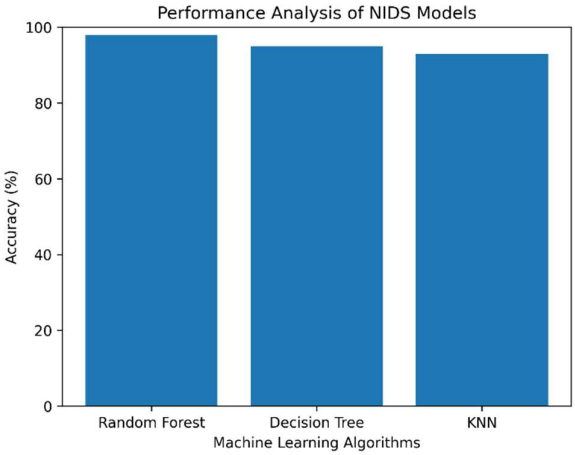
4.1 Performance Analysis

Performance analysis is an essential step in evaluating the effectiveness of the proposed Network Intrusion Detection System. After training the machine learning models using the **KDD Cup 1999 dataset**, the system was tested to measure how accurately it could detect malicious network activities and distinguish them from normal traffic.

In this project, three machine learning algorithms—**Random Forest, Decision Tree, and K-Nearest Neighbors (KNN)**—were used to classify network connections as either normal or attack. The dataset was divided into training and testing sets to ensure that the models were evaluated using unseen data. This approach helps determine how well the models can generalize to real-world network traffic.

Several evaluation metrics were used to analyze the performance of the models. These metrics include **accuracy, precision, recall, and F1-score**, which provide a comprehensive understanding of the classification performance.

The experimental results showed that the **Random Forest algorithm achieved the highest accuracy among the three models**. This is mainly due to its ensemble learning approach, which combines multiple decision trees to improve prediction accuracy and reduce overfitting. Random Forest is also capable of handling complex patterns and large datasets effectively, making it suitable for intrusion detection tasks.



5. CONCLUSION

In this project, a **Network Intrusion Detection System (NIDS)** based on machine learning techniques was developed to detect malicious activities in network traffic. With the rapid growth of internet usage and network-based services, cybersecurity threats have become increasingly sophisticated, making it essential to develop intelligent systems capable of identifying and preventing cyberattacks.

The proposed system utilized the **KDD Cup 1999 dataset**, which contains labeled records of normal and malicious network connections. Through proper data preprocessing, feature selection, and data transformation, the dataset was prepared for machine learning model training. Three classification algorithms—**Random Forest, Decision Tree, and K-Nearest Neighbors (KNN)**—were implemented to analyze network traffic and classify it as either normal or attack.

algorithms, the **Random Forest classifier achieved the highest accuracy**, outperforming Decision Tree and KNN due to its ensemble learning capability and ability to handle complex patterns in network data. The system successfully identified most malicious activities while maintaining a low rate of false positives.

The performance analysis confirmed that machine learning techniques can significantly enhance the efficiency and accuracy of intrusion detection systems compared to traditional rule-based approaches. By automatically learning patterns from network traffic data, the system can detect both known and unknown attacks, making it a valuable tool for improving network security.

References

1. Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009).**"A Detailed Analysis of the KDD Cup 1999Dataset."**Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications.
2. Chandola, V., Banerjee, A., & Kumar, V. (2009).**"Anomaly Detection: A Survey."**ACM Computing Surveys, 41(3), 1–58.
3. KDD Cup 1999 Dataset.**UCI Machine Learning Repository.**Available: https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html