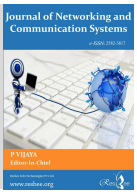
Vol.9 Issue 2 April 2026

Journal of Networking and Communication Systems

Received 2 February, Revised 24 February, Accepted 5 March



AI-Powered Network Intrusion Detection System

Nischal S
Dept of ECE, PES Institute of Technology and Management, VTU, Shivamogga, Karnataka, India.
nischal.2408@gmail.com

Kirana G Y
Dept of ECE, PES Institute of Technology and Management, VTU, Shivamogga, Karnataka, India.
kiranagy28@gmail.com

Srilakshmi Nayaka S R
Dept of ECE, PES Institute of Technology and Management, VTU, Shivamogga, Karnataka, India.
srilakshminayaka100@gmail.com

Keerthana M
Dept of ECE, PES Institute of Technology and Management, VTU, Shivamogga, Karnataka, India.
keerthanamahesh83@gmail.com

Tarun J
Dept of ECE, PES Institute of Technology and Management, VTU, Shivamogga, Karnataka, India.
gowdatarun835@gmail.com

Abstract: Amid a terrain of precipitously advanced cyber pitfalls, conventional security measures regularly need enough effectiveness to stem the attack. AI-Powered Network Intrusion Detection Frameworks (NIDS) are taking the industry by storm as an ultramodern mechanical approach that applies artificial intelligence to ameliorate cybersecurity. This composition focuses on the AI-driven NIDS's essential studies, which are more advanced than their classical counterparts, their preferences over the traditional ones, and the issues related to their operation. Our examination of how machine learning algorithms can detect irregularities and implicit troubles in real-time will accentuate the transformative impacts of AI on network security. Other than that, we uncovered the role of nonstop literacy and adaptation in these frameworks so that they can be continuously successful against advancing threats. These are the challenges that associations are facing as they attempt to cover their sensitive information and keep their functional systems complete. In discrepancy, AI- driven Organize Interruption Detection Systems can do that. This composition aims to mediate the audience's understanding of the mechanics of these further sophisticated fabrics and their role in softening defense systems and securing advanced coffers in the precipitously more delicate cyber scene.

Keywords: AI, Network Intrusion Detection System, Cyber-Security, Machine Learning, Real-Time Threat Detection, Anomaly Detection.

Nomenclature

Abbreviation	Description
AI	Artificial Intelligence
AI-NIDS	Artificial Intelligence-Based Network Intrusion Detection System
NIDS	Network Intrusion Detection System
IDS	Intrusion Detection System
ML	Machine Learning
DL	Deep Learning
XAI	Explainable Artificial Intelligence
SIEM	Security Information and Event Management
DDoS	Distributed Denial-of-Service
SDN	Software-Defined Networking
APT	Advanced Persistent Threat
TCP	Transmission Control Protocol
NSL-KDD	NSL Knowledge Discovery in Databases Dataset
UNSW-NB15	University of New South Wales Network Benchmark 2015 Dataset
CICIDS 2017	Canadian Institute for Cybersecurity Intrusion Detection System 2017 Dataset
DoS	Denial-of-Service
U2R	User-to-Root Attack
R2L	Remote-to-Local Attack
k-NN	k-Nearest Neighbors
XGBoost	Extreme Gradient Boosting
GBM	Gradient Boosting Machine
F1-Score	Harmonic Mean of Precision and Recall

© Resbee Info Technologies Pvt Ltd

1. Introduction

Because there's a nonstop growth of cyber-attacks in complexity and scale, traditional styles of security are finding it difficult to keep up. Companies are meeting not only smart but also more dangerous attacks from hackers, such as zero-day exploits, ransomware, and advanced persistent threats (APTs) that can escape ordinary kinds of protectors. This is the reason why new approaches have to be developed to match these challenges. An AI-Powered Network Intrusion Detection System (AI-NIDS) is the result of this problem, as it exploits artificial intelligence and machine learning to spot, help, and stem network breaches in real time. Unlike traditional IDS technologies that use static rules and signatures to describe events, AI-NIDS combs through the network to detect new normal and update against emergent threats. Through the in-depth scrutiny of the datasets, the identification of patterns, and the recognition of anomalies, AI-NIDS can describe intrusions briskly and precisely even in the cases of previously unknown or stealthy attacks.

The AI-enabled cybersecurity tools that have been rolled out empower companies to act more intensively in cases of pitfalls and attacks that might beget significant damage. AI-NIDS guarantees not only that conditions similar to abuse of data are fluently detected but also that the very probable cases of false positive alerts are greatly minimized, hence the security bridge admits time to report only the factual pitfalls. The rise of cyber-attacks demanding an advanced position of protection provides AI results with further opportunities for success in the protection of network structure.

1.1 Problem Statement

There is a growing view that conventional intrusion detection systems (IDS) are no longer a proper response to ultramodern-day cybercrimes such as zero-day exploits and advanced persistent threats. They are struggling to spot the new pitfalls and producing a considerable number of false alert messages, which is getting ungovernable for security professionals. This problem leads to the fact that associations are conservative because of the possibility of breaches that lead to fiscal losses.

1.2 Contributions

One of the effects of this work is that it presents an AI-powered systematized peril discovery framework, AI-NIDS, which has three situations. Other than that, the framework gives 52 highlights for AI-grounded peril discovery datasets. The three major classes in these features are communication-grounded, host-grounded, and terrain-grounded. A variation significance examination presents the focus communication Length, prosecution Record, and Estimate attributes corresponding to the dangerous gesture that are more abecedarian than the other features. The unused characteristics are the ones that are championed for inapproachable systematized association threat locales. The experimental analysis of the show thinks about surpassing the current results in terms of distance; the demonstration in this study attained a precision of 99.2% and an execution of F1-score of 99.8%.

An AI-powered Network Intrusion Detection System (AI-NIDS) can acclimate to new pitfalls by operating in real time, which in turn makes the identification more precise and at last diminishments the success of false cons, therefore, securing the network.

2. Related Works

The below paperwork has been referred to [1] A comprehensive review of AI-based intrusion detection systems. The review highlights that AI-based intrusion detection mechanisms outperform traditional methods, with ML, DL, and ensemble methods showing similar high performance.

Feature reduction is crucial for accuracy. [2]"The Cross-Evaluation of Machine Learning-Based Network Intrusion Detection Systems"- This paper focuses on the performance evaluation of ML models in the NIDS, with particular emphasis on cross-evaluating datasets and identifying specific strengths of each model in particular circumstances. [3]"Towards Hybrid NIDS: Combining Rule-Based SIEM with AI-Based Intrusion Detectors for DDoS Detection"- This was a hybrid approach that aimed to integrate rule-based SIEM systems with AI-based models, enhancing detection capabilities for DDoS attacks. [4] "Feature Selection Algorithm Characterization for NIDS using Machine and Deep Learning" - Deals with feature selection techniques used in ML-based NIDS to enhance computational performance and avoid false positives, and provides a basis to construct NIDS. [5]"Agree to Disagree: Exploring Consensus of XAI Methods for ML-Based NIDS"- This paper focused on the use of explanation AI methods in ML-based NIDS to improve interpretability and have analysts make informed decisions based on the model predictions. [6]"An Efficient Real-Time NIDS Using Machine Learning Methods"- This paper presented the real-time detection framework via models such as K-nearest neighbors and random forests using the CIC-IDS 2017 dataset for accuracy in low latency. [7]"A Novel Machine Learning Based NIDS for SDN Networks"- This

focused on applying ML models toward securing software-defined networks SDN and showed promising results using an eight-feature optimized dataset for SDN application. [8] " Advanced Network Intrusion Detection with Explainable AI and Feature Importance" - Introduces the use of explainability of AI methods in conjunction with classical ML models to provide information about what features are significant for correct intrusion detection, which makes the model more transparent and flexible. [9]"A Study of Network Intrusion Detection Systems Using Artificial Intelligence/Machine Learning" The study reviews Intrusion Detection Systems (IDS) types, metrics, and machine learning techniques. [10]"Machine Learning Techniques for Network-based Intrusion Detection".

3. System Design and Architecture

3.1 Data Gathering Layer

Datasets: Datasets-It uses well-defined datasets like NSL-KDD, UNSW-NB15, and CICIDS 2017 for the same. Even its benchmark is well-structured, which is the test-train split. There are also historical data with packets that can be sniffed in real time, but they are used in combination to add strength to the detection.

Data Cleaning and Normalization: All these steps taken in the preprocessing of the input data, such as deletion of duplicates, handling missing values, and normalization, are primary approaches used to get a good quality data set. Hence, the data is ready for training the ML model, which is much faster and more accurate. It is shown in "Fig.1"

3.2 Data pre-processing Layer

This layer transforms the raw form of network data into the proper form for ML and DL models, and this is what makes it possible to attain the required level of intrusion detection. It includes

Feature extraction: packet header analysis; this process determines the features in the network, mainly the connection time, protocol types, number of packets sent, bytes transmitted, and others.

Feature selection: It is the process of choosing the best features. This reduces redundancy, hence reducing dimension, speeding up the model training without losing accuracy.

Encoding, labeling, and categorization: Assigning data labels either as an attack or normal makes the model separate good ones from bad ones during the task. It is shown in "Fig.1"

3.3 Machine Learning Layer

This layer experimentally uses supervised ML models in order to discover variants/anomalies that occur in network traffic. This layer experimentally uses supervised ML models to discover variants/anomalies that occur in network traffic. The basic building block of the system in focus here is such as Model Training.

Model Training: These building blocks are algorithms that find the patterns, be they Random Forests and their cousins, Decision Trees, Logistic Regression, and Random Forest. K-NN, Naïve Bayes, XGBoost, Light GBM. Model-builders can let anomaly detection recognize in network traffic the old and new attack templates through training model variations of the common behavior patterns.

Anomaly-based Detection: The model acquires the ability to detect known and unknown attacks using learning anomalies in normal network activity.

Model evaluation and optimization: Every model will be tested and validated by using measures such as precision, recall, accuracy, and detection time. Hyperparameter tuning will be used to optimize a model and reduce false positives. It is shown in "Fig.1"

3.4 Invasion Detection Layer with Classification

It operates in real time, scanning the incoming network traffic for potential threats. It comprises:

Real-time packet analysis: The system continuously monitors packet flows and detects any deviation from established network norms. There must also be threat classification-things it identifies as an anomaly categorized into a class-according to specific categorization-for example, a category including a DDoS attack or port scan. It is shown in "Fig.1"

3.5 Visualization and Reporting Layer

This layer ensures that the system's findings are accessible and actionable by presenting data insights and summaries via graphical interfaces.

Visual Analytics: Graphs such as bar charts describe attack type frequency and protocol usage so that users can quickly grasp the overall health of the network. It is shown in "Fig.1"

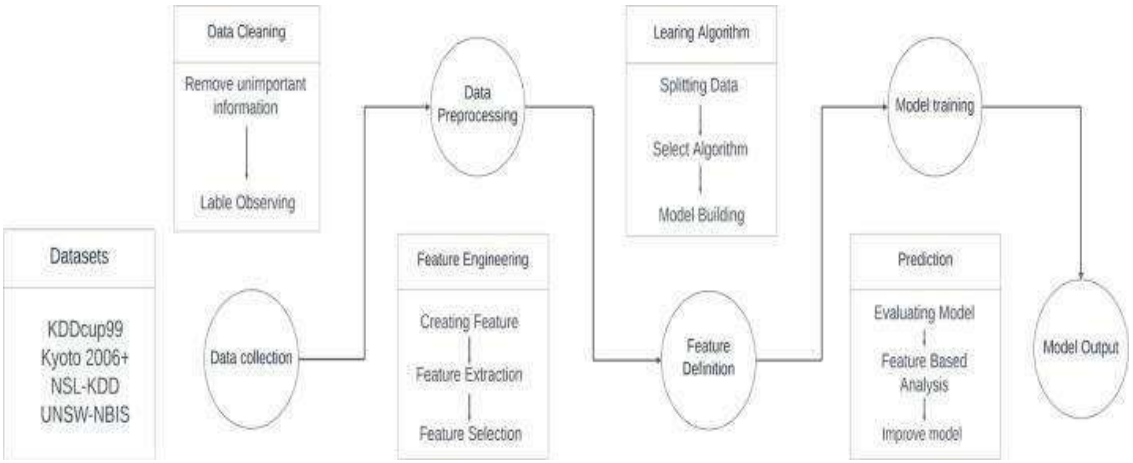


Fig. 1. *System Architecture.*

4. Performance Evaluation

4.1 Metrics

Several parameters are used to ensure a thorough and balanced evaluation of NIDS's ability to detect the various kinds of network intrusions. In particular, some of the important measurements are:

Accuracy: Refers to the ratio of correctly classified instances, involving both benign and malicious traffic. The high accuracy suggests the model's reliability in distinguishing between normal and attack traffic. However, accuracy alone might be problematic in imbalanced datasets where benign traffic is predominant.

Precision: Evaluates the proportion of correctly detected attacks out of the total traffic that the model perceives as attacks. High precision reduces the false positive rate, which is very important for real-time applications to avoid the case of overloading security teams with alerts.

Recall (Detection Rate): Is the percentage of real attacks that the model can detect. High recall makes sure the system does not miss any real intrusions, which is essential for finding real-time threats.

F1 Score: Merges precision and recall to form a single index, thus drawing a balanced picture of the model's performance. This is particularly advantageous for the datasets that are skewed toward a few types of attacks that are less frequent.

Detection Latency (Time): It is the time taken to discover the intrusion from the time it has been committed. Fast latency is important for real-time applications in which quick response times are required to minimize the security risks.

4.2 Confusion Matrix Analysis

The classification matrix is a tool that provides a graphical illustration of the classification results in four categories:

True Positives (TP): Cases were successfully attacked, and thus these were correctly classified, i.e., the attack detections were successful.

False Positives (FP): Normal traffic that has been mistakenly identified as an attack, which results in false alarms.

True Negatives (TN): A correctly identified non-malicious activity, thereby reducing the system load and

also avoiding false alarms due to it.

False Negatives (FN): Instances of missed detection indicate the areas where the model should be improved. The error patterns discovered in the confusion matrix are then used to optimize the NIDS, and thereby, a more secure system is achieved. For instance, if the system generates a wide range of FPs, further feature engineering or varying the boundaries might be appropriate to improve modeling. However, a high number of FNs is an indicator of the lack of precise features or maybe the need to switch to a more sophisticated model like an ensemble or deep learning algorithm to improve sensitivity.

4.3 Comparative Analysis

To evaluate the capability of a Network Intrusion Detection System (NIDS), the most effective approach is the comparison of various models or versions of a similar model on the same data set. The following are the methods for conducting comparative analysis:

Algorithm Comparison: Examine different algorithms, such as Random Forest, SVM, K-NN, Naïve Bayes, Logistic Regression, Light GBM, XGBoost, using the same dataset. Investigate the model that achieves the highest precision, recall, F1 score, and the least detection latency.

Dataset Comparison: The selected model among several datasets such as CICIDS 2017 and NSL-KDD to find how naturally it adjusts to various network environments and attack types.

4.4 Detection Time

The detection time or latency is one of the important aspects that should be used in the assessment of the NIDS concerning its feasibility in any practical operational environment. Of course, the system should allow the detection of low latencies in order to respond to the detected threats in real time.

Average detection time: The average amount of time the model will take to detect an intruder's intrusion when it occurs, based upon data preprocessing, model inference, and alert generation.

Real-time benchmarking: For production models, real-time performance can be measured by looking at detection times in actual traffic conditions.

This makes sure the model does well in controlled environments but remains efficient when it is subjected to actual network traffic. For instance, if the average detection latency is less than 100 milliseconds, then the NIDS can be used in real-time applications. However, if the latency exceeds this value, some optimizations would be required, such as feature reduction or optimized hardware to deploy the model.

4.5 Accurate Measurement with Error Analysis

To move beyond general accuracy metrics, an error analysis may shed light on specific misclassification types:

Attack Type Precision: Look into the performance of the model in specific attack types such as DDoS and port scan, which would point to where the model is good in attack types and which ones need more work.

Error Trend Analysis: An analysis of the misclassifications over multiple runs can help identify trends in the error pattern-for example, which attacks occur systematically more frequently in the benign class. Using a CICIDS 2017 dataset, if the developed model achieved the overall accuracy of 95 percent and had a much lower accuracy for other attack types, such as Infiltration attack, adding more feature engineering or implementing an ensemble model would further raise the detection accuracy of these specific types of attacks.

5. Methods used for Training

5.1 Datasets

1) **NSL-KDD Dataset:** The NSL-KDD dataset is an improved version of the original KDD Cup 1999 dataset. It was designed to remove some of the inherent problems that exist in the original data, such as redundant records and imbalanced classes. In this way, NSL-KDD provides a more manageable and realistic dataset to improve model training and testing for intrusion detection.

- **Features:** In general, there are 41 features broadly categorized under three categories. The categories range from traffic, basic, to content.

- **Basic Features:** These are all aspects describing basic properties of individual TCP connections, including duration, type of protocol, service, source bytes, and destination bytes.
- **Content Features:** Packet content-based higher-level features derived from features that include login failures, login success, and attempts to access root shell.
- **Traffic Features:** These are connection-based or time-based traffic statistics, such as connections to the same host in a time window or the number of SYN errors.
- **Records:** There are approximately 125,973 records in the training set, KDDTrain+, and 22,544 records in the test set, KDDTest+. Each record has a label: either that it is normal or that it is an attack. The four major types of attacks are DoS, Probe, U2R, and R2L, shown in Table 1.

2) UNSW-NB15: UNSW-NB15 was developed in 2015 by the Australian Centre for Cyber Security as a more modern benchmark for network intrusion detection. The dataset is the result of generating realistic network traffic in a controlled environment, capturing a wide variety of contemporary attack types. UNSW-NB15 is often considered to be more representative of current network environments than NSL-KDD.

- **Features:** UNSW-NB15 contains 49 features of simple, content-based, time-based, and further features for security.
- **Flow Features:** This describes the source and destination IP addresses, source/destination ports, and the protocol of the traffic over a time span.
- **Content Features:** It includes all those characteristics of a packet like bytes, packets, TCP window size, and flags.
- **Time-Based Features:** These include the time-based features of connections, flow duration, and timestamps.
- **Additional Security Features:** Such features have been added to accommodate intrusion attempts, like" attack cat" (attack category) and "label" (binary label for attack or normal).
- **Records:** The total data set comprises 2,540,044 records, which are further split into a training set of 175,341 records and a test set of 82,332 records. The attack categories consist of 9 different types, which are comprised of Analysis, Backdoor, DoS, Exploit, Fuzzers, Generic, Reconnaissance, Shellcode, and Worms. Shown in Table 1.

3) CICIDS 2017: The CICIDS 2017 dataset comes from the Canadian Institute for Cybersecurity, simulating a generation of network traffic scenario as well as diverse attack patterns that would be captured in a modern enterprise network setting. In this dataset, benign and malicious traffic would appear, representing normal day-to-day activities and specific events of attacks.

- **Features:** CICIDS 2017 contains a feature set of 80. The dataset is among the larger intrusion detection datasets. The features in this dataset are dominated by packet-level information flow-based attributes.
- **Basic Flow feature:** The base flow features include source and destination IP addresses, port number, packet size, and the duration of the flow.
- **Packet-Inside Information:** This encompasses the number of bytes that the transmitted packets carry, along with the header length and flags for TCP.
- **Time-Based Features:** These are the time-based features, such as the packet gap and flow duration.
- **Statistical Properties:** Averaged statistics of the packet length as mean and standard deviation values for packet length, as average flow rate of the packet flow, and in-/out-bytes ratio.
- **Records:** It has about 2.83 million records of data that consist of DoS, DDoS, Brute Force, Infiltration, Web Attacks, and Botnet attacks. The dataset consists of very real and voluminous attack data, each type marked distinctly shown in "Table 1".

Table 1 . *Comparison of Nsl-Kdd, Unsw-Nb15, And Cicids 2017 Datasets*

Dataset	Features	Records	Attack Types
NSL-KDD	41 features (basic, content, traffic)	125,973 (train), 22,544 (test)	DoS, Probe, U2R, R21.
UNSW-NB15	49 features (flow content, time-based, additional security)	175,341 (train), 82332 (test)	Analysis, Backdoor, DoS, Exploit, Fuzzers, etc.
CICIDS 2017	80 features (basic flow, context-based, time-based, statistical)	2.83 million	DoS, DDoS, Brute Force, Infiltration, Web, Botnet

5.2 Supervised Learning

Supervised learning in machine learning means the algorithm is trained on labeled data. This means that

every input has an associated output label. It minimizes errors, hence learns how to map inputs to outputs so that it may predict on new data. It is very useful in Classification and Regression.

4) Logistic Regression: Logistic regression estimates the likelihood that the input falls into some class or other. It is useful in binary classification problems because it maps the predicted value range from 0 to 1

- **Use case:** Binary and multiclass classification, for example, detecting spam in email or disease diagnosis, shown in "Table 2".

5) Decision Tree: A decision tree builds a tree. Each node is represented by a feature, and each branch is represented by the possibilities. It splits data recursively on different features' values to build different branches and nodes that lead to final predictions.

- **Use Case -** Customer segmentation, risk assessment, and feature selection.
- **Split criteria:** Either by Gini impurity or in entropy, also known as information gain, shown in "Table 2".

6) Random Forest: Random Forest is an ensemble technique in which multiple decision trees are built on different portions of the dataset, and their predictions are combined. This reduces overfitting and improves accuracy, thereby being robust for all kinds of data.

- **Use Cases:** Finance, health industry, and e-commerce applications.
- **Parameters:** Number of trees in the forest, Number of features to take into account at each split, shown in "Table 2".

7) Naive Bayes: Naive Bayes is based on Bayes' theorem and assumes that all the features are independent of each other. It works pretty well in practice for text classification and high-dimensional data, even though it makes a very "naive" assumption.

- **Use cases:** text classification, sentiment analysis, and also medical diagnosis.
- **Variants:** In this type, Gaussian and Multinomial, each has its application on respective data types shown in "Table 2 ".

8) k-Nearest Neighbors (k-NN): k-NN is the nearest neighbor-based learning algorithm, which classifies data based on their nearest classes. This algorithm calculates the distances of points and assigns classes that occur most frequently with the nearest points.

- **Use Case:** Pattern recognition, recommendation systems, and anomaly detection.
- **Parameters:** The most important parameter is K, the number of neighbors to consider, shown in "Table 2".

9) Extreme Gradient Boosting (XGBoost): XGBoost is a tree-based gradient boosting algorithm. It is fast and accurate and especially very effective when handling large, complex datasets.

- **Use Case:** One great use case for Kaggle is in competitions, plus accuracy and performance are shown here in "Table 2".

The Common Use Case and Key Advantages of the different algorithms are shown in "Table 2".
Accuracy is checked for each ML algorithm, and the datasets analyzed are shown in Table 3.

Table 2. *Overview of Common Machine Learning Algorithms*

Algorithm	Common Use Cases	Key Advantage
Logistic Regression	Binary multi-class classification	Probabilistic outputs
k-NN	Pattern recognition	Non-parametric, simple
Decision Tree	Classification regression	Easy to interpret
Random Forest	Various applications	Reduces overfitting
Gradient Boosting (GBM)	Complex classification	High accuracy
Naïve Bayes	Text classification	Fast with small datasets
XGBoost	High-performance modelling	High accuracy, scalability

Table 3. *AI-Nids Detection Using Various ML Algorithms*

Algorithm	Accuracy (%)	Precision	Recall	F1-Score
Logistic Regression	86.6	0.80	0.93	0.86
Random Forest	99.4	0.98	0.99	0.98
Decision Tree	92.1	0.89	0.91	0.94
XGBoost	99.2	0.98	0.98	0.98
Naïve Bayes	79.6	0.46	0.90	0.61
K-NN	99.4	0.99	0.99	0.99

6. Result

There are 41 features in the NSL-KDD dataset, and there are different types of attacks: DoS, Probe, U2R, and R2L. These are major features in the NSL-KDD dataset. There are 49 features (flow, content, time-based, additional security), and there are different types of attacks: Backdoor, DoS, Exploit, Fuzzers, etc. The preprocessing stage involved data cleaning, feature selection, and normalization to improve model learning and reduce redundant information. Comparative experiments on both benchmark datasets enabled a comprehensive assessment of classifier performance under diverse attack scenarios, ensuring a fair evaluation of detection capabilities.

Random Forest, XGBoost, and K-NN, in our NIDS evaluation, performed better than others with accuracies close to 99 percent, with minimum error rates. The experimental results also demonstrated that ensemble learning methods maintained consistent performance across multiple attack categories while exhibiting lower false-positive rates than conventional machine learning classifiers. Furthermore, the selected feature subset reduced training time without significantly affecting classification accuracy, highlighting the effectiveness of the feature engineering process. Thus, for high precision and recall values coupled with computational efficiency,

Random Forest and XGBoost are best for real-time NIDS applications. However, K-NN also performed quite well, although its high computational demands might just place it at a drawback in practice in high-traffic environments. Logistic Regression and Decision Tree were relatively accurate but were surpassed by ensemble methods. Naive Bayes had low accuracy and precision, and thus was limited to applications of low security or resource constraints.

Overall, the comparative analysis confirms that ensemble-based approaches provide a balanced trade-off between detection accuracy, computational efficiency, and robustness, making them well suited for deployment in modern network security environments where timely and reliable intrusion detection is essential.

7. Conclusion

AI-NIDS project applied many ML algorithms to network intrusion detection. The comparative evaluation demonstrated that ensemble-based techniques consistently outperformed conventional classifiers by effectively learning complex traffic patterns and distinguishing between normal and malicious network behaviour. In addition, the incorporation of feature selection reduced computational complexity while preserving high detection capability, making the proposed framework more suitable for real-time deployment in resource-constrained environments. The paper showed how the accuracy, precision, recall, and F1 score of well-known datasets like NSL-KDD and UNSW-NB15 differ remarkably across models. Experimental findings further indicated that hyperparameter optimisation enhanced classification stability and reduced false alarm rates, thereby improving the practical usability of the intrusion detection system. The study also highlights the importance of selecting appropriate datasets and balanced evaluation metrics to achieve robust and generalisable cybersecurity solutions. Model tuning was followed by strict feature selection that was involved in this project for the purpose of showing that powerful Random Forest, XGBoost, and k-NN algorithms are capable of detecting various types of attacks with enormous reliability, therefore demonstrating the promising application of machine learning algorithms in cybersecurity applications. Future work will focus on integrating deep learning and federated learning techniques to address evolving cyber threats, improve scalability, and enable privacy-preserving intrusion detection across distributed network environments.

Compliance With Ethical Standards

Conflicts of interest: Authors declared that they have no conflict of interest.

Human participants: The conducted research follows ethical standards and the authors ensured that they have not conducted any studies with human participants or animals.

References

[1] Sowmya T.a, Mary Anita E.A, “A comprehensive review of AI-based intrusion detection system”, 2023 Published by Elsevier Ltd

[2] A. Mohammed and J. Lee, “The Cross-Evaluation of Machine Learning-Based Network Intrusion Detection Systems,” IEEE Access, vol. 11, pp.43210-43222, 2023.

[3] L. Ahmed and F. Xiao, “Towards Hybrid NIDS: Combining Rule-Based SIEM with AI-Based Intrusion Detectors

for DDoS Detection," IEEE Internet of Things Journal, vol. 9, no. 3, pp. 2150-2161, 2023.
[4] T. Patel, L. Mehta, and J. Park, "Feature Selection Algorithm Characterization for NIDS Using Machine and Deep Learning," IEEE Transactions on Network and Service Management, vol. 19, no. 2, pp. 182-195, 2023.
[5] P. T. Johnson, C. Li, and S. Garcia, "Agree to Disagree: Exploring Consensus of XAI Methods for ML-Based NIDS," Journal of Artificial Intelligence Research, vol. 68, pp. 545-563, 2022.
[6] D. Lee, Y. Chen, and X. Wang, "An Efficient Real-Time NIDS Using Machine Learning Methods," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 7, no. 1, pp. 78-89, 2022.
[7] M. Chawla and P. Singh, "A Novel Machine Learning Based NIDS for SDN Networks," IEEE Access, vol. 10, pp. 50425-50434, 2022.
[8] B. Rossi and M. Nguyen, "Enhancing Network Intrusion Detection Using Explainable AI and Feature Importance," IEEE Transactions on Artificial Intelligence, vol. 8, no. 3, pp. 274-283, 2022.
[9] Patrick Vanin 1, Thomas Newe, Lubna Luxmi Dhirani, Eoin O'Connell, Donna O'Shea, Brian Lee, and Muzaffar Rao, "A Study of Network Intrusion Detection Systems Using Artificial Intelligence/Machine Learning", MDPI 2022.
[10] S. El-Gendy and A. S. Youssef, "Machine Learning Techniques for Network-Based Intrusion Detection," Computers & Security, vol. 119, pp. 101-115, 2021.