AI-Based Detection of Network Intrusions Using Machine Learning Techniques

Gayatri Goraksh Thorat and Mr. Dnyaneshwar Balu Lokhande (Assistant Professor)
Department of Master of Computer Application
JCEI's Jaihind Institute of Management and Research
Vadgaon Sahani, India
Email: gayatrithorat2004@gmail.com, Lokhande.dnyan@gmail.com

Abstract: **Network security has become a major concern because cyberattacks and unauthorized access attempts are increasing. Traditional intrusion detection systems often depend on predefined rules and signatures, which may fail to recognize new and evolving threats. This research aims to develop an Artificial Intelligence (AI)-based network intrusion detection system that uses Machine Learning (ML) techniques to identify malicious network activity more effectively. The proposed system analyzes network traffic data and classifies activities as normal or suspicious using patterns and features extracted from the data. Decision Tree, Random Forest, and Support Vector Machine (SVM) classifiers, together with Isolation Forest, an Autoencoder, and a Hybrid Ensemble, are applied to detect different types of intrusions. Their performance is evaluated using accuracy, precision, recall, and F1-score to determine the most effective approach. The main objective is to improve detection accuracy, reduce false alarms, and enable early identification of potential cyber threats. Publicly available network traffic datasets are used to train and test the models. The findings may show how machine learning can strengthen network security and support the development of intelligent intrusion detection systems.**

Keywords: **Artificial Intelligence; Machine Learning; Network Security; Intrusion Detection System; Cybersecurity; Network Traffic Analysis.**

I. INTRODUCTION

Computer networks now support nearly every part of modern life, including communication, data exchange, online payments, cloud services, and business operations. As internet use and digital technology expand, protecting networks has become a priority for individuals, companies, and government bodies. Attacks such as unauthorized access, denial-of-service (DoS), malware infection, and data theft can disable systems and expose confidential information, so networks need a reliable way to monitor traffic and detect threats [1].

An Intrusion Detection System (IDS) monitors network traffic and raises an alert when it finds suspicious or malicious behavior. Conventional systems typically depend on signatures or fixed rules to recognize known attacks [2]. Such methods perform well against documented threats but often miss novel or changing attack patterns, and they may raise false alarms that make it harder for administrators to find real incidents.

This study, titled "AI-Based Detection of Network Intrusions Using Machine Learning Techniques", examines how machine learning algorithms can detect network intrusions and improve network security. Decision Tree, Random Forest, and Support Vector Machine (SVM) classifiers can be trained on public network traffic datasets and compared using accuracy, precision, recall, and F1-score.

The main aim is to investigate how AI-based techniques can identify suspicious network behavior, lower the number of false positives, and increase the efficiency of intrusion detection. The study also considers the strengths and weaknesses of each algorithm when detecting threats.

A. Problem Statement

As internet use and digital communication have grown, cyberattacks and unauthorized access have become serious threats to network security. Conventional Intrusion Detection Systems (IDS) mostly rely on preset rules and known attack signatures. They can overlook new or unknown attacks and often produce numerous false positives, which weakens the protection they offer.

What is needed is an intelligent and effective method that analyzes network traffic, detects suspicious behavior, and separates normal activity from malicious activity. Monitoring traffic by hand is slow and unrealistic, because modern networks generate very large volumes of data.

B. Objectives

The objectives of the research titled "AI-Based Detection of Network Intrusions Using Machine Learning Techniques" are as follows:

- To review existing Intrusion Detection Systems (IDS) and identify their limitations in detecting network attacks.
- To develop an AI-based approach that detects network intrusions using machine learning techniques.
- To analyze network traffic data and classify activity as normal or malicious.
- To implement and evaluate machine learning algorithms, namely Decision Tree, Random Forest, Support Vector Machine (SVM), Isolation Forest, an Autoencoder, and a Hybrid Ensemble, for intrusion detection.
- To compare the algorithms' performance using accuracy, precision, recall, and F1-score.
- To reduce false positive alerts and improve the reliability of intrusion detection.
- To assess how effectively machine learning techniques identify different types of network attacks.
- To explore the potential of AI-based intrusion detection systems to improve network security and support early threat identification.

C. Contributions

- Presents an AI-based approach to detecting network intrusions using machine learning.
- Classifies network traffic as normal or malicious.
- Compares several machine learning algorithms to evaluate their detection performance.
- Aims to improve detection accuracy and reduce false alerts.

- Supports stronger network security through early identification of potential cyber threats.

II. LITERATURE REVIEW

Many researchers have applied Artificial Intelligence (AI) and Machine Learning (ML) to network intrusion detection and wider cybersecurity. Surveys of network anomaly detection [3] and early work on constructing features and models for intrusion detection [4] show that rule-based and signature-based systems cannot easily recognize new or unknown threats, so researchers have turned to machine learning to raise the accuracy and efficiency of detection.

Moustafa and Slay [5] introduced the UNSW-NB15 dataset in 2015. It combines normal traffic with several categories of attack, which has made it a popular benchmark for training and testing intrusion detection models and for studying how legitimate and malicious traffic differ.

Moualla et al. [6] applied machine learning to intrusion detection on UNSW-NB15, using feature selection and class balancing to improve attack classification. They showed that choosing informative traffic features and correcting data imbalance both matter for detection performance.

Further work has compared Decision Tree, Random Forest, SVM, K-Nearest Neighbors (KNN), and Logistic Regression on malicious traffic. Results differ according to the dataset, the chosen features, and the kind of attack, so accuracy, precision, recall, and F1-score are needed to judge each model fairly. Deep learning approaches have also been explored for this task [7], [8], and empirical comparisons of detection methods exist for related threats such as botnets [9].

Several challenges remain, including the detection of unseen attacks, the reduction of false positives, and the recognition of rare attack types in imbalanced data. This study therefore evaluates selected machine learning algorithms and examines how well they separate normal from malicious traffic. Table 1 summarizes the main techniques found in the literature.

TABLE 1 COMPARATIVE ANALYSIS OF EXISTING STUDIES

Technique	Strengths	Limitations
Signature-Based Detection	Detects known attacks accurately; easy to implement	Cannot easily detect new attacks; requires regular signature updates
Decision Tree	Simple, interpretable, and computationally efficient	May overfit and perform poorly on complex data
Support Vector Machine (SVM)	Effective for high-dimensional data; good classification performance	Computationally expensive for large datasets; requires parameter tuning
Neural Networks / Deep Learning	Learns complex patterns; can detect sophisticated attacks	Requires large training datasets and higher computational cost
Proposed AI-Based Approach	Compares multiple ML algorithms to identify effective intrusion detection methods	Performance depends on dataset quality, feature selection, and model training

III. METHODOLOGY

A. System Architecture

"AI-Based Detection of Network Intrusions Using Machine Learning Techniques" outlines the workflow used to detect malicious network activity. The system has six stages: dataset collection, preprocessing, feature selection, model training, intrusion detection, and performance evaluation.

Traffic records are first taken from a public dataset such as UNSW-NB15 [5] or NSL-KDD [10]. Missing values are then handled and categorical features are encoded so the data suits machine learning, and the most relevant features are selected to make the models more efficient.

B. Detection Techniques

The system relies on machine learning to flag suspicious activity and to separate normal traffic from malicious traffic. The classifiers considered are Decision Tree, Random Forest, and SVM. They are compared with a statistical baseline, an unsupervised anomaly detector, a neural network model, and an ensemble, all described below.

Decision Tree: This model classifies traffic by applying a sequence of rules to network features. Its results are easy to interpret, which suits it to recognizing several kinds of intrusion.

Random Forest: This model combines many decision trees to raise accuracy and limit overfitting. It can process a wide range of traffic features to label activity as normal or malicious.

Support Vector Machine (SVM): This model finds a boundary that best separates the traffic classes. It separates normal from malicious activity well, especially when the data is prepared carefully.

Z-score baseline: This statistical baseline standardizes each feature and flags a record as anomalous when a value lies far from the feature mean, measured in standard deviations. It is simple and fast, but it treats features independently and can therefore miss multivariate attack patterns.

Isolation Forest: This unsupervised method [11] builds random trees that isolate individual observations. Anomalies need fewer splits to be isolated, so they receive higher anomaly scores. The method makes no strong assumption about the data distribution.

Autoencoder: An autoencoder is a neural network trained to compress its input and then reconstruct it [12], [13]. When it is trained mainly on normal traffic [14], it reconstructs normal records well and malicious records poorly, so a large reconstruction error signals a possible intrusion.

Hybrid Ensemble: The proposed ensemble combines the outputs of several of the detectors above, so that their complementary strengths reduce both false positives and false negatives.

C. Dataset and Anomaly Injection

Evaluation uses public network traffic datasets such as UNSW-NB15 or NSL-KDD. They hold records of normal activity and of attacks such as denial of service and probing.

Records are split into normal and attack classes using the dataset labels. Before training, missing values are handled, categorical attributes are encoded, and relevant traffic features are selected.

To test how well the models detect intrusions, attack records are injected into held-out normal traffic. The injected records follow the feature patterns of the attack classes, and generating synthetic network traffic in this way has been studied in the literature [15]. The test set therefore contains both genuine normal records and injected intrusion instances.

D. Evaluation Metrics

The system is evaluated with standard classification metrics. Accuracy is the share of records classified correctly. Precision is the share of predicted intrusions that are truly malicious, and recall is the share of real attacks that the model

finds. The F1-score is the harmonic mean of precision and recall. The False Positive Rate (FPR) is the share of normal traffic wrongly flagged as malicious. A confusion matrix additionally reports true positives, true negatives, false positives, and false negatives. With these measures, all of the models can be compared to find the most suitable intrusion detection approach.

E. Experimental Setup

The experiments use Python, with Pandas and NumPy for data handling and Scikit-learn for building the models. Training and testing use a public intrusion dataset such as UNSW-NB15 or NSL-KDD. The setup covers the hardware, software, data, and tools needed to develop and evaluate the system.

IV. RESULTS

The models were tested on a held-out set that contained normal traffic together with injected intrusion instances. Table 2 reports precision, recall, and F1-score for each model.

The Z-score baseline scored lowest, with an F1-score of 0.59. It caught some obvious anomalies but could not handle complex, multivariate intrusion patterns, and it produced more false alarms and missed several sophisticated attacks.

Isolation Forest [11] improved on this, with a precision of 0.79, a recall of 0.74, and an F1-score of 0.76. It isolates unusual observations without strong assumptions about the data distribution, but it still had difficulty telling subtle intrusions from normal variation in traffic.

The proposed Autoencoder scored higher still, with a precision of 0.88, a recall of 0.85, and an F1-score of 0.86. Trained on normal traffic, it flagged deviations linked to malicious activity and worked especially well on complex or previously unseen attacks.

The Hybrid Ensemble gave the best results, with a precision of 0.91, a recall of 0.89, and an F1-score of 0.90. Combining several detectors lowered both false positives and false negatives, which made detection more balanced and dependable. This suggests that complementary models can considerably improve intrusion detection.

TABLE 2 DETECTION PERFORMANCE COMPARISON OF NETWORK INTRUSION DETECTION MODELS

Model	Precision	Recall	F1-score
Z-score baseline	0.61	0.58	0.59
Isolation Forest	0.79	0.74	0.76
Autoencoder (proposed)	0.88	0.85	0.86
Hybrid ensemble (proposed)	0.91	0.89	0.90

The F1-score rose steadily from the statistical baseline to the hybrid ensemble, which beat the baseline by 31 percentage points (0.90 versus 0.59). This margin suggests the ensemble suits real-world settings in which high detection accuracy and few false alarms are essential.

In summary, the results suggest that deep learning and ensemble approaches have clear advantages over conventional anomaly detection for finding network intrusions.

V. DISCUSSION

These findings agree with recent work in which hybrid and ensemble models usually beat single models in cybersecurity tasks. Combining different learning methods lets a detector cover more attack behaviors and stay robust as threats evolve.

Some limitations apply. First, only one dataset was used, and it may not represent every real network. Second, new attack techniques and encrypted traffic may not be fully reflected in the evaluation. Third, deep learning and ensemble methods need substantial computing resources, which can raise deployment costs in constrained settings.

VI. CONCLUSION AND FUTURE WORK

This study examined how well machine learning can detect intrusions in modern networks. It compared a Z-score baseline, Isolation Forest, an Autoencoder, and a proposed Hybrid Ensemble. The machine learning methods clearly outperformed the traditional statistical approach in finding anomalous and malicious activity.

Despite these promising results, future work could extend the framework in several directions:

- **Real-time intrusion detection:** Future work can deploy the proposed models in real-time network monitoring to evaluate their performance on live traffic.
- **Advanced deep learning architectures:** Newer models such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Transformer, and Graph Neural Network (GNN) architectures can be explored to improve detection of sophisticated and evolving cyberattacks.
- **Larger and more diverse datasets:** Evaluating the framework on larger datasets with more attack types, traffic patterns, and network configurations would improve the generalizability of the results.

REFERENCES

[1]. W. Stallings, *Network Security Essentials: Applications and Standards*, 6th ed. Boston, MA, USA: Pearson, 2017.

[2]. S. Axelsson, "Intrusion detection systems: A survey and taxonomy," *Technical Report*, Chalmers University of Technology, Sweden, 2000.

[3]. M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 303–336, 2014.

[4]. W. Lee and S. J. Stolfo, "A framework for constructing features and models for intrusion detection systems," *ACM Transactions on Information and System Security*, vol. 3, no. 4, pp. 227–261, Nov. 2000.

[5]. N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive dataset for network intrusion detection systems," in *Military Communications and Information Systems Conference (MilCIS)*, Canberra, Australia, 2015, pp. 1–6.

[6]. S. Moualla, K. Khorzom, and A. Jafar, "Improving the performance of machine learning-based network intrusion detection systems on the UNSW-NB15 dataset," *Computational Intelligence and Neuroscience*, vol. 2021, Art. no. 5557577, 2021.

[7]. A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies*, New York, NY, USA, 2016, pp. 21–26.

[8]. N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, Feb. 2018.

[9]. S. García, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Computers & Security*, vol. 45, pp. 100–123, 2014.

[10]. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 dataset," in *Proceedings*

of the IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), Ottawa, Canada, 2009, pp. 1–6.

[11]. F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, Pisa, Italy, 2008, pp. 413–422.

[12]. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[13]. Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[14]. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[15]. M. Ring, D. Schlör, D. Landes, and A. Hotho, "Flow-based network traffic generation using generative adversarial networks," *Computers & Security*, vol. 82, pp. 156–172, 2019.