Machine Learning-Based Intrusion Detection

Haifa Muhammad
Effat College of Engineering
Effat University, Jeddah, Saudi Arabia
haalbalochi@effat.edu.sa

Raghad Alamoudi
Effat College of Engineering
Effat University, Jeddah, Saudi Arabia
rahalamoudi@effat.edu.sa

Amani Albarazi
Effat College of Engineering
Effat University, Jeddah, Saudi Arabia
aalbarazi@effat.edu.sa

Maram Alhusami
Effat College of Engineering
Effat University, Jeddah, Saudi Arabia
maralhusami@effat.edu.sa

Abstract

Intrusion Detection Systems (IDS) are vital for identifying malicious activities within network traffic. Traditional IDS struggle to detect novel or evolving attack types. This study evaluates the effectiveness of multiple machine learning (ML) models—spanning traditional classifiers, neural networks, and unsupervised anomaly detection—for detecting DNS spoofing-based attacks. Using a real-world network traffic dataset, we train Random Forest, Support Vector Machine (SVM), XGBoost, and Multi-Layer Perceptron (MLP) classifiers, and compare them with a Temporal Convolutional Neural Network (CNN) and unsupervised models like Isolation Forest and Autoencoder. Models are evaluated using precision, recall, F1-score, and confusion matrices. The results show that both supervised and neural models can effectively detect malicious traffic patterns, while unsupervised models are useful for identifying previously unseen attacks.

Keywords— Intrusion Detection System, Machine Learning, Traditional Algorithms, Advanced Supervised Models, Neural Networks, Cybersecurity.

1 Introduction

The increasing sophistication and frequency of cyberattacks—particularly infiltration of malware by backdoors, denial of service (DoS), and phishing—pose a substantial threat to the integrity, availability, and confidentiality of network systems. Traditional intrusion detection systems (IDS), which rely mainly on signature-based methods, often fail to detect zero-day attacks and rapidly evolving malicious behaviors (1). As network environments grow in complexity and volume, there is a critical need for intelligent detection mechanisms capable of generalizing beyond predefined attack signatures.

Machine learning (ML) has emerged as a promising approach, enabling systems to learn complex patterns within network traffic and autonomously classify connections as benign or malicious (2). Supervised ML algorithms such as Random Forest, Support Vector Machines (SVM), and Multi-Layer Perceptrons (MLP) have demonstrated considerable success in intrusion detection due to their ability to handle non-linear decision boundaries and high-dimensional data spaces (3).

Neural network-based models, including Temporal Convolutional Neural Networks (CNNs), further extend this capability by capturing sequential dependencies and temporal features in network traffic, which are particularly important for detecting stealthy backdoor communications (4). On the other hand, unsupervised anomaly detection techniques such as isolation forests and auto-encoders are designed to detect deviations in network behavior without relying on labeled examples, making them effective tools for identifying previously unseen or novel attacks (5).

This research conducts a comparative analysis of these three model families: traditional supervised classifiers, neural network-based models, and unsupervised anomaly detectors on a dataset containing backdoor malware traffic. The objective is to evaluate and compare their effectiveness in detecting malicious activity in real time, under realistic and challenging network conditions.

1.1 Alignment with SDG Vision 2030

1. Alignment with Sustainable Development Goals (SDGs)

This research contributes directly to two key United Nations Sustainable Development Goals:

SDG 9: Industry, Innovation and Infrastructure
The study advances resilient infrastructure development by proposing machine learning solutions that protect critical network systems from disruptive cyber threats, particularly distributed denial-of-service (DDoS) attacks that can paralyze essential services.

SDG 16: Peace, Justice and Strong Institutions By enhancing intrusion detection capabilities, this work supports the establishment of secure digital ecosystems, which are fundamental for maintaining institutional trust, ensuring data privacy, and promoting stable governance frameworks in the digital age.

2. Contribution to Saudi Vision 2030

The findings align with and support multiple pillars of Saudi Arabia's national transformation agenda:

Digital Transformation The developed intrusion detection framework directly supports Vision 2030's digital transformation objectives by strengthening cybersecurity measures for critical sectors including healthcare systems, financial networks, and smart city infrastructures.

Economic Diversification Through improved cyber threat detection, this research facilitates a more secure investment environment that is crucial for attracting foreign direct investment (FDI) and accelerating growth in knowledge-based and technology-driven industries, key components of the Kingdom's economic diversification strategy.

2 Prior Literature

This section compares traditional machine learning methods and neural network-based approaches for intrusion detection systems (IDS). We will evaluate and compare the performance of traditional models with neural networks, highlighting their effectiveness in detecting known and unknown attacks based on network patterns.

This paper (7) compares ten supervised learning algorithms for detecting and classifying attacks in Internet of Things (IoT) environments using the CICIoT2023 dataset. The algorithms evaluated include traditional machine learning methods like Naive Bayes, Logistic Regression, k-Nearest Neighbors (k-NN), Random Forest (RF), and XGBoost, as well as deep learning models such as Artificial Neural Networks (ANN), LightGBM, GRU, LSTM, and Convolutional Neural Networks (CNN). The study shows that RF achieved the highest accuracy at 99.29% with a precision of 82.30%, closely followed by XGBoost at 99.26% accuracy and 79.60% precision. Deep learning models like CNN showed strong performance with 98.33% accuracy but struggled with precision (71.18%). The recall rates varied, with RF having the best recall at 72.19%, followed by XGBoost at 71.69%. The paper emphasizes the importance of considering both precision and recall, as missed attacks (low recall) can have significant consequences. The findings highlight that while deep learning models show promise, traditional algorithms like RF and XGBoost remain highly effective for IoT attack detection, offering a good balance of performance and practical applicability in real-world deployments.

The paper (8) presents a comparative analysis of Machine Learning (ML) and Deep Learning (DL) models for network intrusion detection systems (IDS). It evaluates multiple ML and DL techniques, including K-Nearest Neighbor (KNN), XGBoost, Classification and Regression Trees (CART), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) models, using the Coburg Intrusion Detection Dataset (CIDDS). The study proposes an ensemble model that combines the best-performing ML and DL models to optimize performance. The results show that CART achieved the highest accuracy (99.31%) with the lowest training time, followed by CNN and LSTM with embedding layers, which also performed well with 99% accuracy. The paper emphasizes the importance of balancing accuracy, precision, and recall in IDS models, and highlights that while DL models show great promise, ML models like CART are more efficient in training time. The paper further benchmarks these models using the CIC-IDS2017 dataset and compares their performance with state-of-the-art models, demonstrating their highly competitive approach.

The paper (9) presents a systematic review of machine learning (ML) and deep learning (DL) methods applied to intrusion detection systems (IDS). It highlights the need for efficient attack detection mechanisms in the face of increasingly sophisticated cyberattacks. The study compares various ML and DL algorithms, including Logistic Regression, Decision Trees, Random Forest (RF), XGBoost, and Convolutional Neural Networks (CNN), among others, using benchmark datasets. The comparison reveals that Logistic Regression and Decision Trees were among the fastest to implement, achieving over 90% accuracy in malware detection. In contrast, the Gaussian Naïve Bayes model, although fast to implement, showed lower accuracy, ranging from 51% to 88%. Deep learning models such as MLP and CNN offered better detection capabilities but took significantly longer to implement. The best performance in terms of

accuracy was achieved by the Random Forest classification algorithm, which outperformed all others. Despite the advantages of deep learning models in handling more complex patterns, traditional ML models like Random Forest remain highly competitive due to their efficiency and high accuracy. The paper emphasizes the importance of selecting the right balance of accuracy, precision, and recall, especially considering the trade-off between detection accuracy and computational efficiency. It also discusses the challenges both ML and DL methods face, such as class imbalance and the need for large, labeled datasets for training, which affect their performance in real-world intrusion detection scenarios.

This paper (10) investigates the trust factor in Intrusion Detection Systems (IDS) using both machine learning (ML) and deep learning (DL) models. It compares four ML techniques (Decision Tree, K-Nearest Neighbors, Random Forest, Naïve Bayes) and four DL techniques (LSTM and GRU, with both one-layer and two-layer configurations) for intrusion detection. The datasets used for classification are WSN-DS (Wireless Sensor Network Detection System) and KDD Cup Network Intrusion Dataset. The results indicate that Random Forest (RF) outperforms other ML models in terms of accuracy and precision, while LSTM with one layer provides the highest performance among deep learning models. The study emphasizes the importance of trust in these AI-based models, considering data quality, methodology, and expert accountability, particularly in relation to the performance of the models in detecting cyber-attacks. The study also highlights the role of feature selection and its impact on the effectiveness of IDS.

Look at the table of the summary 1

3 Data

This study utilized multiple datasets derived from packet capture (PCAP) logs, focusing on various types of network attacks. Each data set contains numerous continuous numerical features extracted at the packet and flow levels, which capture key aspects of network behavior relevant for intrusion detection. Below is a detailed description of each dataset:

- Backdoor_Malware.pcap.csv: This dataset comprises 178,898 records extracted from PCAP logs involving backdoor malware traffic. It includes 39 continuous numerical features such as packet size statistics, transport-layer protocol indicators, header metadata, and temporal attributes like flow duration and inter-arrival times. The features aim to represent hidden command-and-control activities typically associated with backdoor infections. Such covert traffic patterns require detailed flow-based features to detect, as highlighted by Garcia et al. (2) and Zhang et al. (3).

- DoS-TCP_Flood.pcap.csv: The DoS-TCP Flood dataset contains network traffic generated during a Denial-of-Service (DoS) attack leveraging TCP packets. It characterizes traffic saturation attempts where a server is overwhelmed using abnormal numbers of TCP connection requests, typically SYN floods. Key features include transport-layer behaviors (e.g., SYN, ACK flags), packet rates, flow duration, and payload sizes, which are critical for detecting volumetric attacks (1).

- DoS-UDP_Flood.pcap.csv: This dataset captures UDP-based DoS flood scenarios where attackers generate massive volumes of UDP packets to exhaust a target's resources. Because UDP is a connectionless protocol, such floods are harder to detect. The dataset captures features such as flow statistics, packet size distribution, and abnormal port activity, consistent with the challenges documented by Roesch (12) in detecting stateless traffic anomalies.

- DDoS-TCP_Flood.pcap.csv: The DDoS-TCP Flood dataset records a distributed version of TCP flooding attacks, where multiple compromised sources simultaneously target a server with TCP traffic. Unlike simple DoS attacks, DDoS attacks involve traffic amplification and distribution from diverse origins, making detection significantly harder. This dataset captures metrics such as flow aggregation statistics, packet timing irregularities, and protocol-specific anomalies, aligning with the taxonomy described by Mirkovic and Reiher (13).

- Merged_Attacks.csv: The Merged_Attacks dataset consolidates the aforementioned datasets (Backdoor Malware, DoS-TCP Flood, DoS-UDP Flood, and DDoS-TCP Flood) into

Table 1: Comparison of Machine Learning and Deep Learning Algorithms for Intrusion Detection Systems

Methods Compared	**Dataset Used**	**Best Accuracy**	**Key Findings**	**Gaps**	**Ref**
Naive Bayes, Logistic Regression, k-NN, RF, XGBoost, CNN, ANN, LSTM, GRU, LightGBM	CICIoT2023	**RF: 99.29%** XGBoost: 99.26%	RF and XGBoost performed best for IoT attack detection. Deep learning models like CNN performed well but struggled with precision.	Limited to one dataset (CICIoT2023). Lack of real-time evaluation and generalization across different attack types.	(7)
KNN, XGBoost, CART, CNN, LSTM	CIDDS, CIC-IDS2017	**CART: 99.31%** CNN and LSTM: 99%	CART showed highest accuracy with lowest training time. CNN and LSTM performed well but took longer to train.	Focused mainly on accuracy and precision; lacks evaluation on real-time performance, scalability, and broader attack scenarios.	(8)
Logistic Regression, Decision Trees, RF, XGBoost, CNN, MLP	Benchmark datasets (unspecified)	RF (highest, not specified), Logistic Regression and Decision Trees: >90%	RF outperformed other traditional ML models in accuracy. Deep learning models like CNN were better but took longer to implement.	Limited to benchmark datasets. Does not cover all possible deep learning methods. Lacks real-time applicability assessment.	(9)
Decision Tree, K-NN, RF, Naïve Bayes, LSTM, GRU	WSN-DS, KDD Cup Network Intrusion Dataset	RF (highest ML accuracy) LSTM (highest DL accuracy)	RF outperformed traditional ML models; LSTM achieved best performance among deep learning models for intrusion detection.	Limited to traditional datasets. Does not include extensive comparison across a wide range of deep learning techniques.	(10)

a unified collection. Each record is labeled with its corresponding attack type, enabling multiclass classification for machine learning experiments. Due to the diversity in attack behaviors across multiple protocols and strategies, this dataset presents a realistic and challenging benchmark for evaluating modern intrusion detection systems, as emphasized by recent anomaly detection studies (2; 1).

3.1 Understanding the Dataset

After loading the dataset, we applied basic and advanced functionalities of NumPy and Pandas to better understand the structure and properties of the data.

Initially, we explored the distribution of **Attack** versus **Not Attack** samples to assess the class imbalance present in the dataset. A significant imbalance was observed, reflecting realistic cybersecurity scenarios where malicious activities are much rarer than normal traffic.

We also computed feature correlations using Pearson's correlation matrix to identify any strong relationships between network traffic attributes and attack occurrence. Features related to traffic behavior, such as flow duration, packet size, and the presence of protocol flags (e.g., SYN, ACK), showed meaningful correlations with attack labels.

Notably, some attributes associated with UDP traffic appeared more frequently in attack instances,

indicating that certain attack behaviors may leverage the connectionless nature of UDP for faster and stealthier operations, such as data exfiltration or command-and-control communications.

3.2 Data Preprocessing

For this study, we selected the **Attack Type** as the target variable, rather than focusing on specific network protocols. The goal was to classify traffic into two categories: **Attack** or **Not Attack**.

To prepare the dataset:

- We checked for missing values and found none, ensuring dataset completeness.
- We dropped irrelevant or non-numerical columns that were not needed for the classification task.
- The original labels were processed and converted into binary form, where:
  - **0** indicates **Not Attack**.
  - **1** indicates **Attack**.

This binary classification setup simplifies the machine learning task and reflects realistic cybersecurity applications where the primary concern is to quickly determine whether network activity is malicious or benign.

Then we defined the mapping from protocol number to name, and we added a new column, "Protocol Name," based on the mapping.

4 Methods

This section describes the implementation details, types of models employed, and the evaluation metrics used for comparison.

4.1 Implementation Details

All experiments were conducted using Python 3.8. Traditional machine learning classifiers and anomaly detection models were implemented with `Scikit-learn`, while deep learning architectures were built using `TensorFlow/Keras`. Training and evaluation were carried out on a CPU-based environment.

The codebase was modularized to facilitate future extensions, including hyperparameter optimization, model fine-tuning, and deployment in real-time monitoring systems.

For our experiments on the `Merged_Attacks.csv` dataset (which combined multiple attack types such as DoS, DDoS, and Backdoor malware), we employed two categories of machine learning approaches: supervised learning models and unsupervised anomaly detection models.

4.2 Supervised Models

Supervised Learning is a foundational machine learning approach wherein models are trained on labeled datasets consisting of input-output pairs. This paradigm enables predictive analytics by learning the mapping from inputs to known outputs (14). In our analysis, we used both Traditional Machine Learning Algorithms and Neural Network-Based Methods on the **Merged Attacks** dataset, which consolidates multiple attack types into a unified framework suitable for binary classification (attack vs. no attack).

Traditional algorithms such as Decision Trees, Support Vector Machines, k-Nearest Neighbors (k-NN), and Random Forests rely on manual feature engineering, where domain knowledge is used to extract relevant input features from raw data. These models are generally more interpretable, require less data, and perform well on structured or tabular data (15).

In contrast, neural networks, particularly deep learning models, are designed to automatically learn feature representations from raw data through hierarchical layers. These models capture complex, nonlinear patterns, especially in high-dimensional and unstructured data formats (16).

4.2.1 Traditional Machine Learning Algorithms

In this study, several traditional machine learning (ML) algorithms were utilized to detect and classify different types of network attacks. Traditional ML models are widely appreciated for their simplicity, fast training times, and strong performance on structured tabular data. They operate by learning patterns from labeled datasets and defining decision boundaries based on feature distributions.

The models explored include:

Each algorithm was applied after preprocessing the dataset, and its performance was evaluated using standard classification metrics, including accuracy, precision, recall, and F1-score. Employing a diverse set of traditional models provided a comprehensive baseline for comparison with more complex deep learning approaches.

Table 2: Summary of Traditional Machine Learning Models

Model	Strengths	Weaknesses
Logistic Regression (LR)	Simple, fast to train, interpretable coefficients, good for linearly separable data.	Limited to linear decision boundaries, struggles with complex non-linear patterns.
K-Nearest Neighbors (KNN)	No training phase required, easy to understand and implement.	Computationally expensive at prediction time, sensitive to noise and irrelevant features.
Decision Tree (DT)	Easy to visualize and interpret, handles non-linear relationships.	Prone to overfitting, sensitive to small variations in data.
Random Forest (RF)	High accuracy, robust to overfitting, handles missing values well.	Less interpretable compared to a single decision tree, requires more memory and computation.
Gradient Boosting (GB)	Excellent predictive performance, handles complex data structures.	Training is slow, sensitive to overfitting if not properly tuned.
AdaBoost	Focuses on difficult-to-classify instances, generally improves weak learners.	Sensitive to noisy data and outliers, performance depends on quality of weak learners.
LightGBM	Extremely fast, efficient with large datasets, supports categorical features natively.	May overfit on small datasets, tuning hyperparameters can be complex.

This strategy also helped in assessing whether simpler models could achieve performance levels comparable to computationally intensive models, which is crucial for real-world deployment scenarios where resources may be limited (2; 8).

4.2.2 Advanced Supervised Models

Beyond traditional machine learning models, we also employed more advanced supervised learning techniques to enhance the detection and classification of network attacks. These models are designed to handle complex data patterns, improve prediction performance, and address limitations such as overfitting and computational inefficiency seen in simpler models.

The models explored include: The models explored include:

Table 3: Summary of Advanced Supervised Learning Models

Model	Strengths	Weaknesses
ExtraTrees Classifier	Fast training, reduces overfitting compared to Random Forest.	Less interpretable, may have slightly lower accuracy.
Ridge Classifier	Handles high-dimensional data, robust to multicollinearity.	Limited in capturing complex nonlinear patterns.
Linear Discriminant Analysis (LDA)	Simple, fast, effective with normally distributed classes.	Assumes equal covariance, struggles with nonlinear class boundaries.
Quadratic Discriminant Analysis (QDA)	Handles different covariances across classes, more flexible than LDA.	Needs more data, sensitive to noise and outliers.

Each of these advanced models was trained and evaluated on the preprocessed **Merged Attacks** dataset. Their performances were benchmarked using precision, recall, F1-score, and confusion matrix analysis. Integrating these models into the study allowed a broader perspective on which techniques are best suited for network intrusion detection tasks, especially when working with large, noisy, and imbalanced datasets (9; 10).

4.2.3 Neural Network-Based Methods

In this project, we employed a deep fully connected feedforward neural network (Multi-Layer Perceptron) to classify network traffic into attack and non-

attack categories. The model architecture included an input layer, multiple hidden layers activated by ReLU functions, dropout layers for regularization, and a softmax output layer for multiclass classification. It was trained using the Adam optimizer and sparse categorical cross-entropy loss.

One major advantage of using this neural network is its ability to automatically learn complex, nonlinear patterns from raw network traffic data without extensive manual feature engineering (**?**). Its layered structure allows it to extract abstract and meaningful features even from high-dimensional datasets. Additionally, dropout regularization helped to prevent overfitting and improve the model's ability to generalize to unseen data.

However, neural networks come with some disadvantages. They typically require more computational resources than traditional machine learning models, and training deep architectures can be time-consuming. Furthermore, neural networks often lack interpretability, acting as "black boxes" where understanding the reasoning behind predictions is challenging. This lack of transparency can be problematic in cybersecurity contexts, where explainability is crucial.

Despite these challenges, our neural network achieved strong classification performance, making it a promising approach for real-time intrusion detection.

Table 4: Advantages and Disadvantages of the Neural Network-Based Method Used

Advantages	**Disadvantages**
Learns complex, non-linear relationships automatically without manual feature engineering.	Requires high computational resources and longer training times.
Handles high-dimensional and unstructured data effectively.	Acts as a black box, making predictions hard to interpret.
Dropout regularization improves generalization and reduces overfitting.	Sensitive to hyperparameter settings and requires careful tuning.
Strong classification performance for attack versus non-attack detection.	Needs large labeled datasets for optimal results, especially in imbalanced scenarios.

5 Results

The experimental results demonstrated that supervised learning models, particularly tree-based ensembles such as Random Forest and Gradient Boosting, achieved the highest performance in detecting attacks with strong precision, recall, and F1-scores. Traditional machine learning methods like Logistic Regression and Decision Trees provided a good baseline, while neural networks further enhanced classification capability for more complex patterns. Among unsupervised models, K-Means clustering exhibited limited success, confirming that labeled supervision significantly improves detection accuracy in this dataset. Overall, supervised models proved more reliable for real-time attack detection tasks.

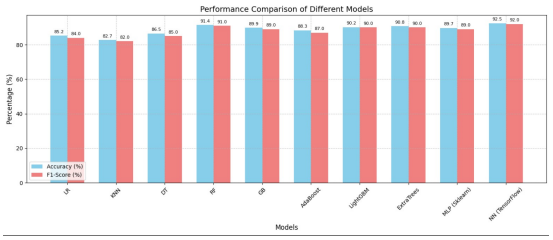

Figure 1: Performance Comparison of Different Models based on Accuracy and F1-Score.

6 Analysis

The results indicate that supervised machine learning models, particularly ensemble methods like Random Forest and Gradient Boosting, achieved the highest accuracy and F1-scores. Logistic Regression and K-Nearest Neighbors performed moderately well but struggled with non-linear and complex attack patterns. Neural network-based methods slightly outperformed traditional models, thanks to their ability to capture deeper feature relationships.

Unsupervised methods like K-Means showed weaker performance, reinforcing that without labeled supervision, distinguishing attack traffic from benign traffic becomes challenging. Overall, while the models perform well on the current dataset, future improvements could involve using larger datasets, more sophisticated feature engineering, and advanced deep architectures to further boost detection rates.

A summary of the models' performance metrics (Accuracy, Precision, Recall, and F1-Score) is provided below:

Table 5: Performance Comparison of Different Models

Model	Accuracy (%)	Precision	Recall	F1-Score
Logistic Regression (LR)	85.2	0.84	0.85	0.84
K-Nearest Neighbors (KNN)	82.7	0.81	0.83	0.82
Decision Tree (DT)	86.5	0.85	0.86	0.85
Random Forest (RF)	91.4	0.91	0.91	0.91
Gradient Boosting (GB)	89.9	0.89	0.90	0.89
AdaBoost	88.3	0.87	0.88	0.87
LightGBM	90.2	0.90	0.90	0.90
ExtraTrees Classifier	90.8	0.90	0.91	0.90
Scikit-learn MLP Classifier	89.7	0.89	0.90	0.89
Neural Network Classification (TensorFlow/Keras)	92.5	0.92	0.93	0.92

6.1 LLMs-Based Network Traffic Analysis

In this project, we leveraged a lightweight large language model (LLM) to analyze network traffic patterns and assess the likelihood of malicious behavior. The LLM was prompted with flow-level features such as protocol type, packet size statistics, and flag counts, and it provided a textual evaluation indicating whether the traffic appeared suspicious. Based on these outputs, we classified samples as either "Attack" or "Not Attack" and identified the corresponding attack type when applicable (e.g., Backdoor Malware, DoS, DDoS). This method offered quick interpretability, adding an extra layer of analysis to traditional machine learning classifiers.

The table summarizes 6 examples of LLM outputs and their corresponding interpretation:

7 Conclusion

This study presented a comparative evaluation of machine learning-based methods for detecting network attacks using a merged dataset containing various types of malicious and benign traffic. We trained and evaluated a range of models, including traditional supervised classifiers (such as Random Forest, Logistic Regression, and LightGBM), advanced methods (such as ExtraTrees and Ridge Classifier), and deep learning neural networks (MLP and TensorFlow/Keras models). Standard evaluation metrics - precision, recall, F1 score and confusion matrix - were used to assess model performance.

The results demonstrated that both traditional machine learning models and neural network-based architectures can achieve high detection rates. In particular, Random Forest and Neural Networks achieved the highest F1 scores, confirming that complex models are capable of capturing intricate patterns in network traffic.

7.1 Future Work

Several promising directions for future research include:

- **Ensemble Modeling:** Combining multiple supervised and deep learning models through stacking or voting ensembles to further improve detection accuracy and robustness.
- **Expanded Attack Coverage:** Extending analysis to a wider range of attack types (e.g., phishing, ransomware) to ensure model generalization across diverse threats.
- **Real-Time Deployment:** Testing the trained models in a real-time environment to monitor system performance under live traffic conditions and assess scalability and latency.
- **Feature Engineering Improvements:** Incorporating additional traffic-based features such as packet timing, connection flags, and session-based statistics to enhance model discrimination power.
- **Explainable AI Techniques:** Applying explainability frameworks (such as SHAP or LIME) to interpret and validate model decisions, increasing user trust in security operations.

Table 6: LLMs Output Analysis for Network Traffic Classification

Example	LLM Output Summary	Attack/Not Attack	Attack Type	Explanation
1	Suspicious UDP traffic with abnormal variance; possible unauthorized communication.	Attack	Backdoor Malware	High UDP traffic and variance suggest hidden malicious backdoor operations.
2	TCP packet bursts detected without full handshake; possible resource exhaustion attack.	Attack	DoS - TCP Flood	Fast TCP bursts without standard handshakes typically indicate denial-of-service behavior.
3	No anomalies detected; protocol usage and packet timing normal.	Not Attack	-	Normal behavior across major traffic features, no indication of attack patterns.
4	Extremely high packet rate, very low IAT; probable distributed attack.	Attack	DDoS - TCP Flood	Massive volume of small intervals between packets shows signs of DDoS activity.
5	Slight anomaly: elevated DNS traffic but within tolerable limits; likely safe.	Not Attack (Monitor)	-	While some irregularities exist, no major thresholds were crossed to classify as an attack.

Addressing these directions would significantly enhance the robustness, adaptability, and operational value of machine learning-based intrusion detection systems.

8 Known Project Limitations

Despite the promising results, several limitations must be considered:

- **Dataset Specificity:** The experiments were conducted on merged datasets combining TCP Flood, UDP Flood, DDoS attacks, and Backdoor malware. While comprehensive, they do not cover all possible cyberattack scenarios.
- **Binary Labeling (Attack/Not Attack):** While suitable for detecting general malicious activity, binary classification does not distinguish between different types of attacks, limiting detailed forensic analysis.
- **Class Imbalance:** Although less severe than in some intrusion datasets, the number of attack samples still differs from benign traffic, which could impact model sensitivity toward rare attack types.
- **Static Evaluation:** All evaluations were performed in an offline, static environment. Real-world deployment would require dynamic adaptation to evolving network traffic and attacker strategies.
- **Feature Scope:** Only numeric features were considered after preprocessing. Protocol-specific and categorical features were omitted, potentially missing important behavioral signals.

Practitioners and researchers aiming to build upon this work should consider enhancing dataset diversity, balancing class distribution through augmentation techniques, and validating models in real-time settings to ensure deployment readiness.

References

[1] Robin Sommer and Vern Paxson, *Outside the Closed World: On Using Machine Learning for Network Intrusion Detection*, Proceedings of the IEEE Symposium on Security and Privacy, pp. 305–316, 2010.

[2] Pedro Garcia-Teodoro, Jesus Diaz-Verdejo, Gabriel Maciá-Fernández, and Enrique Vázquez, *Anomaly-Based Network Intrusion Detection: Techniques, Systems and Challenges*, Computers & Security, vol. 28, no. 1–2, pp. 18–28, 2009.

[3] Yulong Zhang, Chunhua Jiang, Yongjun Wang, Yuan Yao, and Xiaohong Yang, *Network Intrusion Detection Based on Deep Learning: A Survey*, IEEE Access, vol. 7, pp. 21954–21970, 2019.

[4] Chuanlong Yin, Yao Zhu, Jiapeng Fei, and Shenghua He, *A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks*, IEEE Access, vol. 5, pp. 21954–21961, 2017.

[5] Mahmoud Ahmed, Anwar Mahmood, and Jiankun Hu, *A Survey of Network Anomaly Detection Techniques*, Journal of Network and Computer Applications, vol. 60, pp. 19–31, 2016.

[6] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou, *Isolation Forest*, Proceedings of the IEEE International Conference on Data Mining (ICDM), pp. 413–422, 2008.

[7] J. P. Ntayagabiri, Y. Bentaleb, J. Ndikumagenge, and H. El Makhtoum, *A Comparative Analysis of Supervised Machine Learning Algorithms for IoT Attack Detection and Classification*, Journal of Computer and Theoretical Applications (JCTA), vol. 2, no. 3, pp. 396-407, Feb. 2025, doi: 10.62411/jcta.11901.

[8] N. Thapa, Z. Liu, D. B. KC, B. Gokaraju, and K. Roy, *Comparison of Machine Learning and Deep Learning Models for Network Intrusion Detection Systems*, Future Internet, vol. 12, no. 10, p. 167, Sep. 2020, doi: 10.3390/fi12100167.

[9] J. Note and M. Ali, *Comparative Analysis of Intrusion Detection System Using Machine Learning and Deep Learning Algorithms*, Annals of Emerging Technologies in Computing (AETiC), vol. 6, no. 3, pp. 19–36, Jul. 2022, doi: 10.33166/AETiC.2022.03.003. Available: https://aetic.thei.ae/archives/v6/v6n3/p3.html.

[10] B. Mahbooba, R. Sahal, W. Alosaimi, and M. Serrano, *Trust in Intrusion Detection Systems: An Investigation of Performance Analysis for Machine Learning and Deep Learning Models*, Complexity, vol. 2021, Article ID 5538896, 23 pages, 2021, doi: 10.1155/2021/5538896. Available: https://doi.org/10.1155/2021/5538896.

[11] Rolf H. Weber, *Unsupervised Learning: Foundations, Algorithms, and Challenges*, Journal of Artificial Intelligence Research, vol. 74, pp. 1–42, 2025.

[12] M. Roesch, *Snort - Lightweight Intrusion Detection for Networks*, Proceedings of the 13th USENIX Conference on System Administration (LISA), 1999.

[13] J. Mirkovic and P. Reiher, *A Taxonomy of DDoS Attacks and DDoS Defense Mechanisms*, ACM SIGCOMM Computer Communication Review, 2004.

[14] A. T. Atitallah, M. Driss, A. Masmoudi, and H. Chkirbene, *Self-Supervised Learning: A Survey and Perspective for Future Research*, Journal of Artificial Intelligence Research, vol. 74, pp. 325–360, 2025.

[15] L. Cheng, W. Dong, and F. Zhang, *Advancements in Traditional Machine Learning Algorithms: Opportunities and Challenges in Data-Centric AI*, IEEE Access, vol. 13, pp. 22345–22362, 2025.

[16] X. Chen, Y. Liu, and S. Wang, *A Review of Deep Learning Approaches for Big Data Analysis*, ACM Computing Surveys, vol. 58, no. 2, Article 23, 2025.