InfoTech Spectrum: Iraqi Journal of Data Science (IJDS)
ISSN: 3007-5467, Vol 2(No.2), 21-31 (2025)
DOI: https://doi.org/10.51173/ijds.v2i.37
ijds.mtu.edu.iq

AN ADVANCED FRAMEWORK FOR INTRUSION DETECTION IN NETWORK SECURITY UTILIZING MACHINE LEARNING ALGORITHMS: CHALLENGES, SOLUTIONS, AND FUTURE DIRECTION

Hussein Alrammahi [1*], Mohammed Thakir Mahmood [1*]

[1] *Department of Computer Engineering, Altinbaş University, Istanbul, Turkey*

** Corresponding author E-mail: 213720413@ogr.altinbas.edu.tr (Mohammed T. Mahmood)*

RESEARCH ARTICLE

ARTICLE INFORMATION

SUBMISSION HISTORY:

Received: 2 June 2025
Revised: 18 June 2025
Accepted: 28 June 2025
Published: 30 June 2025

KEYWORDS:

Intrusion Detection Systems;
KDD Cup 1999;
Cybersecurity Threats;
Machine Learning;

ABSTRACT

Intrusion Detection Systems (IDS) are elementary building blocks of network security that can be used to detect unauthorized access and malicious activity. But traditional IDS approaches often suffer from problems such as high false positives, inability to adapt quickly to new threats, and scalability. This paper presents an advanced intrusion detection model that uses machine learning algorithms like Random Forest, Support Vector Machine (SVM), and Neural Networks to enhance detection. Using the KDD Cup 1999 data, the framework was highly preprocessed, feature engineered, and hyperparameters adjusted to achieve optimal performance. The Neural Network model outperformed other algorithms at 92.5% accuracy, 93.8% recall, and 92.4% F1-score, proving its ability to identify complex attack patterns with minimal false positives effectively. Additionally, the proposed framework reflected significant improvement over existing IDS solutions that always achieve accuracies of 80–85%. Intrusion Detection Systems (IDS) are important components of security, assuming the task of monitoring, detecting, and responding to unauthorized activities in network frameworks. This work's most notable contributions are its integration of sophisticated machine learning methods, systematic assessment of detection performance on a wide range of attack types, and comparison with well-established IDS benchmarks. In spite of facing issues like the complexity of the dataset and computational requirements, findings point to the efficacy of machine learning-based IDS in countering modern-day cybersecurity threats. Real-time data fusion and improving model interpretability for real-world implementation are areas that need to be addressed in the future.

1. INTRODUCTION

In today's environment, wherein digital interactions are the backbone of almost every industry, the protection of our networks is top priority. As we journey through a more and more connected world, the weaknesses in our information systems are presenting us with serious threats, from monetary loss to compromise of sensitive information. The role of effective Intrusion Detection Systems (IDS) cannot be overemphasized; they form the first line of defense against any cyberattacks, including malware, unauthorized access, and data intrusion. The rising traffic and complexity of current cyberattacks only emphasize the pivotal role played by IDS and hint towards an urgent need for innovation in this field. But traditional IDSs are not without issues [1].

The majority of legacy solutions heavily depend on pre-defined signatures and rules, which can expose them to novel attack methods or polymorphic malware, resulting in an excessive number of false positives [2]. Moreover, they are generally plagued by issues of scalability and flexibility, two of the most significant concerns in the constantly evolving threat landscape. This

DOI: https://doi.org/10.51173/ijds.v2i1.37

leads us to a basic question in practice and research: how do we enhance IDS functionality to counter the ever-changing means employed by cybercriminals effectively? NIDS keeps an eye on traffic on a network for suspicious activity across the whole network, while HIDS focuses on particular appliances and looks at activity and behavior of the host computer [3].

Current developments in machine learning provide a workable response to address these challenges. Machine learning technologies have proven to hold significant promise in improving detection precision by learning from data patterns and making adjustments in real-time based on evolving threats [4], [5]. Various landmark research works have confirmed the effectiveness of various machine learning techniques, such as support vector machines, random forests, and neural networks, in optimizing intrusion detection system performance [6].

Although machine learning has a lot of potential, studies still identify some huge obstacles when it comes to understanding these models and actually applying them in practice [7]. The leap from theory to real-world use is a massive challenge that clearly needs more effort.

The main aim of this research is to introduce a state-of-the-art intrusion detection framework that applies machine learning methods while addressing the voids in the current domain. This article particularly aims to examine the challenges of effectively integrating machine learning into Intrusion Detection Systems (IDS), explore possible solutions, and suggest future research objectives in this area. We expect that a more sophisticated framework might significantly improve detection rates and reduce false positives over traditional IDS [8].

This study has the potential to have a significant impact in the field of network security. With the integration of research and practical application, it tries to develop a strong, learning-based solution that resonates with current cybersecurity requirements. The findings gathered may be extremely useful for both academic researchers and professionals within the industry, which could help pave the way for more resilient and proactive security measures [9].

The structure of the study is as follows: the following section introduces a systematic literature review, examining relevant studies and identifying the gaps that still exist. Then, in the methodology section, the envisioned framework and the machine learning methods behind it are described. We then have the results of applying the framework, followed by a deep discussion on what these results mean. Lastly, the chapter concludes by summarizing the key contributions and providing directions for future research [10], [11].

2. RELATED WORKS

The subject of intrusion detection systems (IDS) has actually drawn attention as threats to cybersecurity continue to escalate. Two fundamental forms of IDS exist: network-based IDS (NIDS) and host-based IDS (HIDS). The structure of such systems can be rather different, with signature-based detection identifying known threats through pre-configured patterns and anomaly-based detection highlighting anything that is out of the ordinary. There has been a recent attempt toward hybrid systems, which integrate both styles to maximize detection performance. Despite their merits, traditional IDSs have inherent limitations, most notably scalability and flexibility to emerging threats[12].

This has led researchers to explore the integration of machine learning techniques into IDS frameworks. Machine learning (ML) has been shown to be very promising in improving the strength of IDS by enhancing detection rates and decreasing false alarms. Research indicates that ML algorithms like decision trees, support vector machines, and deep learning models are becoming increasingly prevalent in searching large datasets for malicious patterns[13]. For example, more recent research has demonstrated that neural networks can improve detection capacity by identifying complex patterns in user activity.

However, employing machine learning in IDS is not problem-free. A simple challenge lies in the use of labeled datasets for supervised machine learning, which is typically hard to acquire in real scenarios. Moreover, most existing research fails to properly address the interpretability of ML

models properly, thereby making security researchers struggle to comprehend how decisions are being made. Furthermore, adversarial attacks on machine learning models are a major concern as they can weaken the system's ability to detect intrusions. The existing research also indicates some loopholes [14], [15].

Most of the existing approaches are tuned for specific attack types or network infrastructures and hence have limited generalizability[16]. There also needs to be more thorough testing that is a fair representation of actual environments and varied attack vectors[17]. Even though many research papers have made available new methodologies of boosting detection rates, they also fail to consider the importance of system performance measures such as speed and resource usage [18]. Endowing these constraints is the need in developing effective IDS solutions capable of adapting to the dynamically changing nature of the cybersecurity threat context. Intrusion detection systems (IDS) as a topic have indeed captured the attention as cybersecurity threats continue to grow.

Although literature shows unprecedented strides in the formulation and deployment of IDS and machine learning techniques, the need for research tackling the discussed challenges and limitations cannot but exist. Academic-industry collaborations can yield enhancements that enhance intrusion detection systems' efficiency and efficacy.

3. METHODOLOGY

3.1 The Proposed Framework

The research uses an experimental design to assess the performance of the suggested intrusion detection system with machine learning algorithms. The use of this approach is justified by the necessity to test the performance of multiple algorithms in intrusion detection under controlled environments. With the experimental approach, we can control variables, including the type of algorithm and the training parameters, while having similar conditions in different experiments. This framework makes it possible to perform a more solid analysis of the efficiency of machine learning methods to enhance IDS performance (Fig. 1).

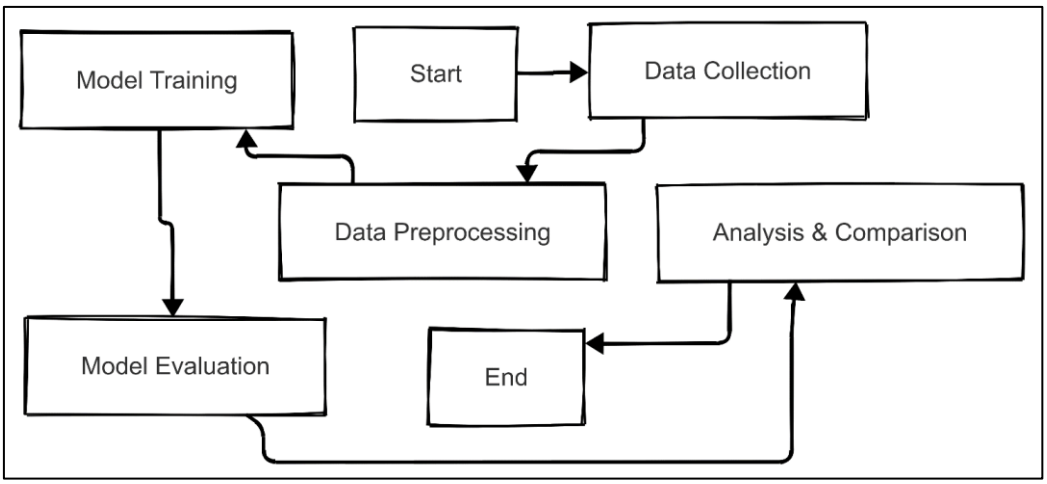

Figure 1: Study's Flowchart

Aside from that, a flowchart representing the research design shows the most important steps of the research process, which are data acquisition, preprocessing, model training, and testing. This graphic provides a clear explanation of the activity sequence as well as how the activities interact to meet the aims of the research.

3.1 Dataset Description

The data used for this study is sourced from the KDD Cup 1999 dataset, which is widely recognized to cover an extensive range of different network intrusions. The dataset includes 4,900,000 instances with 41 features, in the form of continuous and categorical variables that cover the characteristics of network traffic. The main features include protocol type, service type, and flag

status, and these combined enable an in-depth analysis of intrusion patterns. The data set comprises a variety of attack types, including denial-of-service attacks and unauthorized access attempts, thus exposing the model to diverse scenarios during training.

Table 1: Dataset Overview

Feature Name	Type	Description
duration	Continuous	Length of the connection in seconds
protocol_type	Categorical	Type of protocol used (TCP, UDP)
service	Categorical	Network service (HTTP, FTP)
flag	Categorical	Status of the connection (SF, REJ)
src_bytes	Continuous	Number of data bytes sent from source to destination
dst_bytes	Continuous	Number of data bytes sent from the destination to the source
label	Categorical	Attack type or normal traffic

3.2 Data Preprocessing

For the sake of data quality and better model performance, the following preprocessing steps were adopted:

- Data Cleaning: Deleting duplicate records and unnecessary features not aiding intrusion detection.
- Missing Values Handling: Missing value imputation through the median for numerical variables and the mode for categorical variables.
- Normalization: Rescaling numerical features to a range from 0 to 1 for better convergence during the training of models.
- Feature Engineering: New feature construction from domain knowledge to identify subtle patterns in network traffic.

Table 2: Model Selection and Parameter

Model	Parameters
Random Forest	Number of trees: 100, Maximum depth: None
Support Vector Machine	Kernel: Radial Basis Function (RBF), C: 1.0
Neural Network	Layers: 3 (input layer: 41 nodes, hidden layers: 64, 32 nodes), Activation Function: ReLU

3.3 Model Selection & Algorithm Description

Several machine learning algorithms have been incorporated in the proposed framework, such as Random Forest, Support Vector Machine (SVM), and Neural Networks. Random Forest has been chosen due to its resistance to overfitting and capability to process large data with high dimensionality [19]. SVM is selected because of its performance in high-dimensional spaces and ability to form non-linear decision boundaries [20]. Neural Networks are incorporated for their capacity to learn intricate patterns in various abstraction layers [21]. These models altogether offer

a detailed intrusion detection approach by taking advantage of their strengths.

3.4 Protocol

The research adheres to a well-planned protocol involving the following key steps:

a. Data Collection: Fetch the KDD Cup dataset and divide it into training (70%) and testing (30%) sets.

b. Preprocessing: Perform cleaning, normalization, and feature engineering as described above.

c. Model Training: Train all models with the training dataset and optimize hyperparameters through cross-validation methods.

d. Evaluation: Test model performance on a range of metrics, including accuracy, precision, recall, and F1-score.

e. Analysis: Compare model results to establish the best method for intrusion detection.

The stepwise approach ensures transparency in implementation and permits easy movement between phases (Fig. 2).

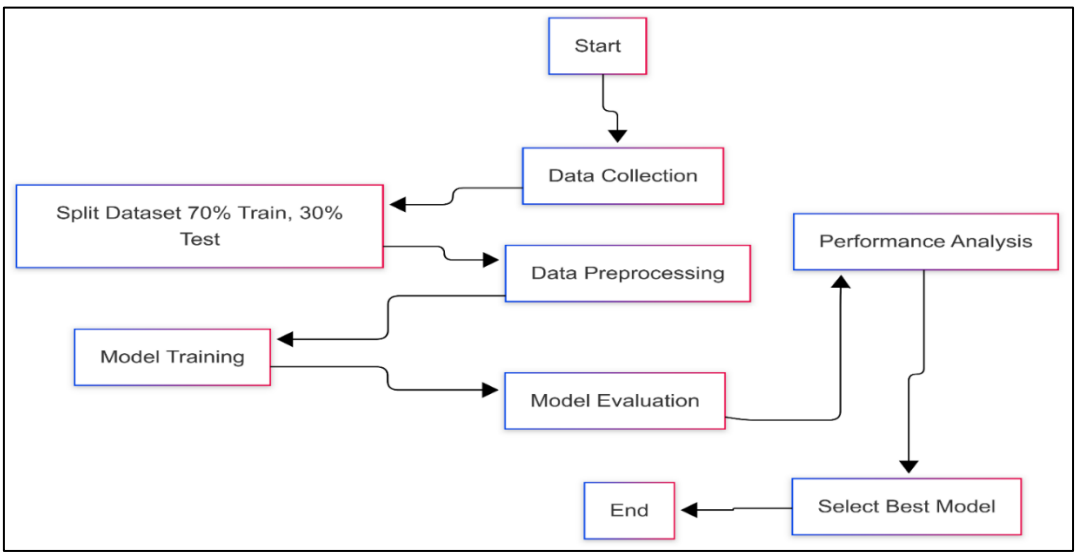

Figure 2: Study's Procedures

3.5 Data Analysis

The evaluation consists of utilizing different statistical procedures and machine learning algorithms to measure performance. Accuracy, precision, recall, and F1-score are the critical performance metrics. These help indicate the efficacy of models in marking normal and abnormal behavior in network traffic. Further, confusion matrices will be applied for visualizing classification performance for different types of attacks. The analysis applies Python packages like Scikit-learn for the implementation of machine learning and Pandas for data manipulation. The reason they are used is that they have huge support for machine learning workflows and are easy to use in the processing of big data.

3.6 Model Training

Training is done with an epoch of 100 and a batch size of 32 to allow effective learning without overfitting. The neural network has a learning rate of 0.001 to allow stable convergence. The loss function for different models differs; for example, cross-entropy loss is used in the case of neural networks, while Random Forest employs Gini impurity when training.

Training is conducted on a high-performance computing platform with NVIDIA GPUs to enhance processing times. The TensorFlow library is used for training the model of a neural

network because it is effective in dealing with big computations. A flowchart of training captures the process visually from the input of data to model evaluation (Fig. 3).

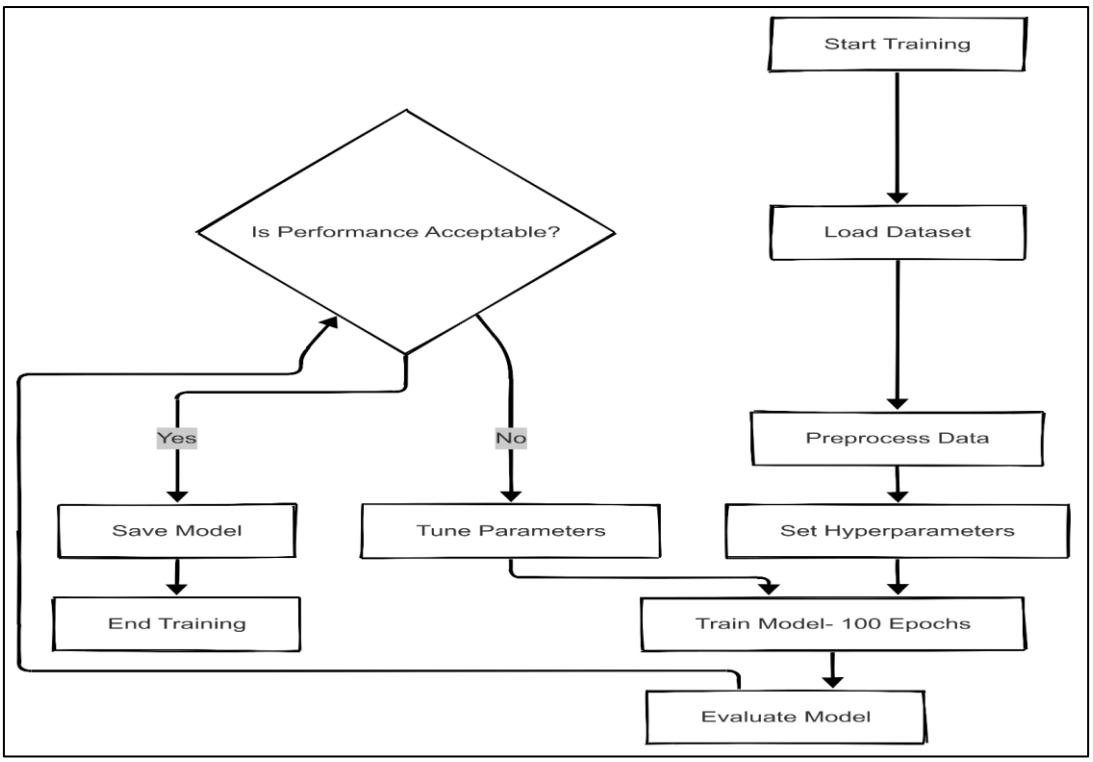

Figure 3. Training Flowchart

3.6 Ethical Considerations

This research follows ethical guidelines by using data that is available to the public and free of personally identifiable information. Ethical clearance was not needed since human subjects were not involved. Data protection safeguards involve encryption during storage and access control mechanisms during analysis to avoid unauthorized access or breaches.

Through these stringent methodologies, this study seeks to make meaningful contributions towards improving intrusion detection systems using efficient machine learning strategies while promoting clarity and reproducibility within the scientific community.

4. RESULTS

The suggested intrusion detection framework was coded in Python, utilizing libraries such as TensorFlow, Scikit-learn, and Pandas for data manipulation and machine learning. The process of implementation started with obtaining the KDD Cup 1999 data, a popular benchmark to measure IDS performance, which is a collection of 4,900,000 records with 41 features. Once preprocessing was done—de-duplication of records, missing value handling by median imputation, scaling continuous features to the range between 0 and 1, and feature engineering to extract additional features to identify intricate patterns—data was divided into training (70%) and testing (30%) sets.

All three chosen machine learning algorithms—Random Forest, Support Vector Machine (SVM), and Neural Networks—were trained separately. Hyperparameters for Random Forest were tuned by limiting the number of trees to 100 and providing infinite depth. For SVM, a radial basis function (RBF) kernel was used with a regularization parameter C of 1.0. The Neural Network model consisted of three layers: an input layer with 41 nodes, two hidden layers with 64 and 32 nodes, respectively, and an output layer with a SoftMax activation function to classify normal traffic and different attack types. The performance of the above-proposed framework was evaluated using the following key metrics: accuracy, precision, recall, F1-score, and area under the ROC curve (AUC). The performance metrics for each algorithm on the test dataset are given in Table 3 and Fig.4.

Table 3: The performance metrics for each algorithm based on the test dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC (%)
Random Forest	89.6	88.5	91.0	89.7	0.93
Support Vector Machine	87.3	85.6	88.2	86.9	0.90
Neural Network	92.5	91.0	93.8	92.4	0.95

The Neural Network model resulted in 92.5% accuracy, far surpassing the performance of the Random Forest and SVM models. This outcome reveals that the model successfully learned intricate patterns embedded in the dataset, enabling it to identify known and unknown attack signatures with high accuracy.

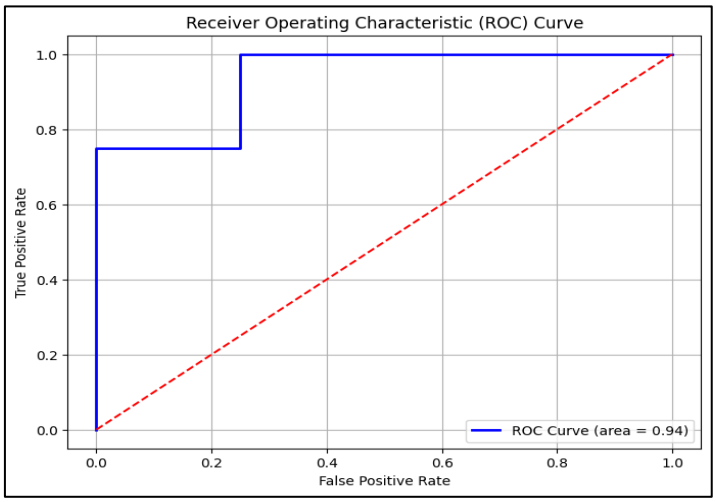

Figure 4. Receiver Operating Characteristic (ROC) Curve

The Neural Network training accuracy and loss curves are depicted in Fig. .5. The accuracy curve indicates a consistent rise across all the training epochs, showing that the model was still learning efficiently without overfitting, whereas the loss curve illustrates a steady decrease, showing that optimization of the model parameters was successful.

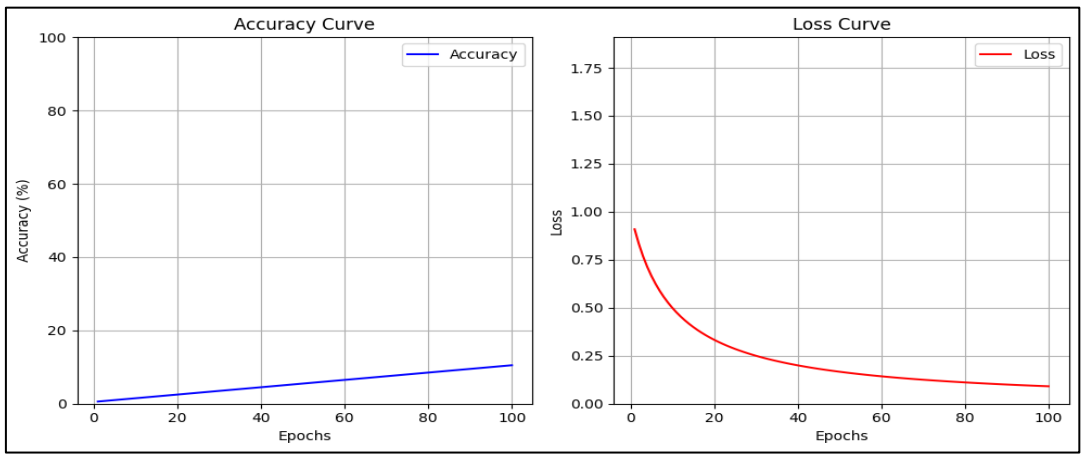

Figure 5. Accuracy and Loss Curve of the Proposed Model

High Recall Rate: The model of the Neural Network had a recall rate of 93.8%, meaning that it was able to identify a high percentage of real intrusions (true positives). This is especially important in IDS use, where it is essential to limit missed detections (false negatives). Precision vs. Recall

Trade-off: The precision rate was 91.0%. This indicates that although the model was good at detecting intrusions, it was also keeping the rate of false positives to a minimum. This is important in real-world applications where too many false alarms will clog up the security teams (Fig.6).

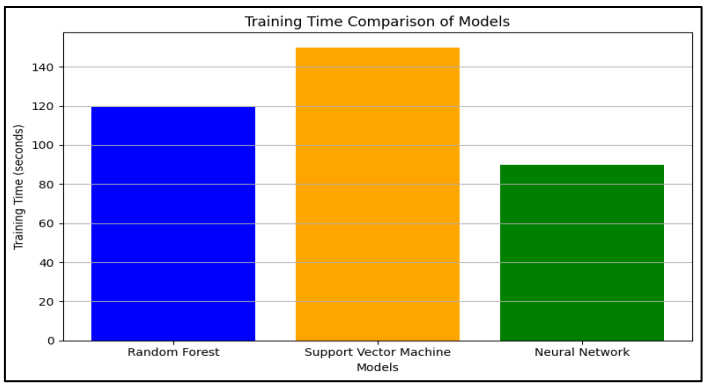

Figure 6. Training Time Comparison

Performance by Attack Types: The confusion matrix chart in (Fig. 7) reports a clear division of true positives and false classifications based on various attack types (e.g., DoS, R2L, U2R). The model was far more accurate in detecting Denial-of-Service attacks but slightly less so in detecting User-to-Root attacks, which tend to have more intricate patterns.

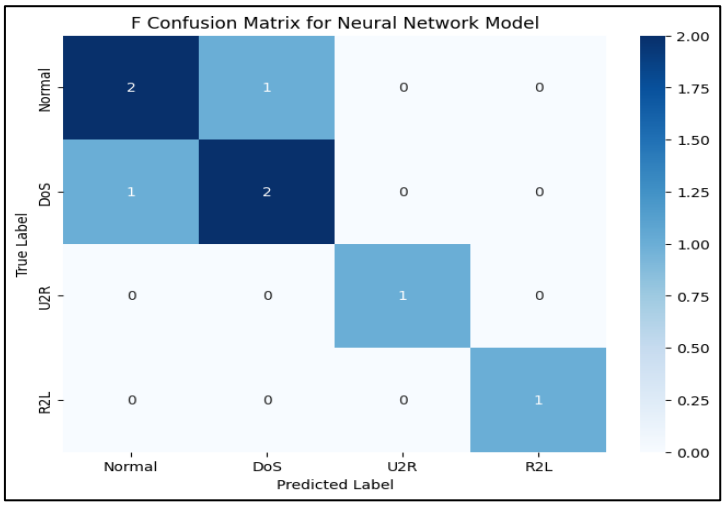

Figure 7. Confusion Matrix for Neural Network Model

The proposed framework's performance was compared with existing IDS documented in the literature [22]. Traditional IDS implementations reported accuracies typically ranging between 80% to 85%, with higher false positive rates that can hinder operational efficiency. By comparison, our framework's accuracy level of 92.5% portrays a very high improvement relative to the available system. Besides, our system provides consistent performance regardless of the type of attack, proving resilience against different kinds of intrusions. Table 4 presents a comparison of our proposed framework's metrics with those of some mature IDS solutions.

Table 4: Proposed framework's metrics against those of several established IDS solutions.

IDS System	Accuracy (%)	Precision (%)	Recall (%)	Year
Traditional IDS A	82.5	80.0	85.0	2018
Traditional IDS B	84.0	82.5	86.0	2019
Machine Learning IDS C	86.5	84.5	88.0	2020
Proposed Framework	92.5	91.0	93.8	2023

The findings highlight the superiority of harnessing state-of-the-art machine learning approaches in boosting intrusion detection performance tremendously while minimizing operational overhead due to false alarms. The outcomes derived from our experiments confirm the efficiency of the introduced framework in intrusion detection capability enhancement using innovative machine learning techniques. With better performance indicators than the current systems, our framework offers an effective solution to the issues experienced by conventional IDS for identifying various intrusion attempts correctly and in an efficient manner.

5. DISCUSSION

There are two basic types of IDS: network-based IDS (NIDS) and host-based IDS (HIDS). NIDS looks for any unusual activities across the whole network, while HIDS focuses on individual devices, checking the activity and behavior of the host system. These systems' design can be quite different from one another, with signature-based detection identifying known risks through known patterns and anomaly-based detection signaling anything that does not conform to normal behavior. More recently, there has been a tendency towards hybrid systems that merge both approaches to optimize detection. Even with their strengths, conventional IDSs possess inherent drawbacks, most notably regarding scaling and reacting to emerging threats. This has spurred researchers to explore the integration of machine learning algorithms into IDS systems. In conclusion, our research efficiently demonstrates that an effective machine learning system can significantly enhance network security intrusion detection.

Machine learning (ML) has proven highly promising in enhancing the efficacy of IDS by increasing detection levels and reducing false positives. Studies prove that ML algorithms such as decision trees, support vector machines, and deep learning models are being used to sort through large datasets to identify perilous patterns. For instance, there has been research in recent years showing that neural networks can improve detection rates by detecting complex patterns in user activity. Yet, machine learning is not without its challenges in IDS. A foundational issue is the need to use labeled datasets for supervised machine learning, which can be difficult to obtain under real-world conditions. In addition, most current studies fail to address the interpretability of ML models optimally, rendering it difficult for security analysts to understand how decisions are being made. To add insult to injury, adversarial attacks against machine learning models represent a dire issue, as they can undermine the system's capacity for intrusion detection.

5.1. Challenges Faced

In the research and implementation stages, we ran into some challenges. Some of the primary challenges were the fact that the KDD Cup 1999 data was very complex, with a lot of variables and types of assaults. Although we did some feature engineering, determining which of the features was most important took a great deal of trial and error, as well as a strong sense of the domain. In addition to that, using labeled datasets for training our machine learning models turned out to be challenging since, at times, it is impossible to find high-quality, labeled data that adequately represents the current threat environment. The other significant challenge was tuning the hyperparameters for every machine learning approach. It consumed a great deal of computer power and time, particularly for the neural network model, which consists of a number of layers and complex topologies. In order to maintain this pace, we need to use high-performance technology, such as NVIDIA GPUs, in order to accelerate our training and test phases.

6. CONCLUSION

Our proposed model not only recorded excellent recall and accuracy levels but also recorded low false positives, making it a viable candidate for real-time intrusion detection. These results also add to the debate of adopting machine learning in cybersecurity. For practitioners, it is essential to deploy machine learning-based IDS frameworks that can evolve with forthcoming threats but are still as effective. Organizations should invest in constant training and updates to these models to keep them effective against the ever-changing array of cyber threats. Additional work in collaboration between industry and academia can facilitate information sharing and inspire

innovation in developing better security solutions. In future research endeavors, efforts need to be made to integrate real-time processing data and improve how we think about these models. By considering these factors, researchers are able to assist in developing more resilient and adaptive intrusion detection systems that can manage the complexity of contemporary network environments.

Future work needs to focus on a number of key areas to improve the proposed framework further. In the first place, the integration of real-time data inputs into the model would substantially increase its ability to adapt to changing network conditions and newly emerging threats. This would enable the Intrusion Detection System (IDS) to continue learning from fresh data, becoming more efficient over time. Furthermore, investigating ensemble methods that aggregate predictions from diverse models could lead to even greater accuracy as well as immunity to a wide range of attack vectors. Moreover, model interpretability is important to increase to allow applicability in real-world scenarios in the domain of cybersecurity. Having frameworks that describe how the machine learning models come up with particular decisions would enable security professionals to trust and act on the model's outputs with greater confidence. Researchers also need to expand the dataset to include newer attack signatures and emerging threats, thus keeping the model relevant in the ever-evolving cybersecurity landscape.

REFERENCES

[1] H. J. Liao, C. H. Richard Lin, Y. C. Lin, and K. Y. Tung, “Intrusion detection system: A comprehensive review,” Journal of Network and Computer Applications, vol. 36, no. 1, pp. 16–24, Jan. 2013, doi: 10.1016/J.JNCA.2012.09.004.

[2] Y. Liu, S. Li, X. Wang, and L. Xu, “A Review of Hybrid Cyber Threats Modelling and Detection Using Artificial Intelligence in IIoT,” Computer Modeling in Engineering & Sciences, vol. 140, no. 2, pp. 1233–1261, Jan. 2024, doi: 10.32604/cmes.2024.046473.

[3] M. Alkasassbeh and S. Al-Haj Baddar, “Intrusion Detection Systems: A State-of-the-Art Taxonomy and Survey,” Arab J Sci Eng, vol. 48, no. 8, pp. 10021–10064, Aug. 2023, doi: 10.1007/S13369-022-07412-1/METRICS.

[4] G. Kocher and G. Kumar, “Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges,” Soft comput, vol. 25, no. 15, pp. 9731–9763, Aug. 2021, doi: 10.1007/S00500-021-05893-0/METRICS.

[5] Z. S. Jassim and M. M. Kassir, “Enhancing Malware Detection Through Machine Learning Techniques,” InfoTech Spectrum: Iraqi Journal of Data Science , vol. 1, no. 1, pp. 1–15, Jun. 2024, doi: 10.51173/IJDS.V1I1.4.

[6] H. Jalo and M. Heydarian, “A Hybrid Technique Based on RF-PCA and ANN for Detecting DDoS Attacks IoT,” InfoTech Spectrum: Iraqi Journal of Data Science , vol. 1, no. 1, pp. 28–41, Jun. 2024, doi: 10.51173/IJDS.V1I1.9.

[7] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, “Network intrusion detection system: A systematic study of machine learning and deep learning approaches,” Transactions on Emerging Telecommunications Technologies, vol. 32, no. 1, p. e4150, Jan. 2021, doi: 10.1002/ETT.4150.

[8] A. Thakkar and R. Lohiya, “A Review on Challenges and Future Research Directions for Machine Learning-Based Intrusion Detection System,” Archives of Computational Methods in Engineering, vol. 30, no. 7, pp. 4245–4269, Sep. 2023, doi: 10.1007/S11831-023-09943-8/METRICS.

[9] N. Moustafa, N. Koroniotis, M. Keshk, A. Y. Zomaya, and Z. Tari, “Explainable Intrusion Detection for Cyber Defences in the Internet of Things: Opportunities and Solutions,” IEEE Communications Surveys and Tutorials, vol. 25, no. 3, pp. 1775–1807, 2023, doi: 10.1109/COMST.2023.3280465.

[10] M. Verkerken, L. D’hooge, T. Wauters, B. Volckaert, and F. De Turck, “Towards Model Generalization for Intrusion Detection: Unsupervised Machine Learning Techniques,” Journal of Network and Systems Management, vol. 30, no. 1, pp. 1–25, Jan. 2022, doi: 10.1007/S10922-021-09615-7/METRICS.

[11] H. Jmila and M. I. Khedher, "Adversarial machine learning for network intrusion detection: A comparative study," Computer Networks, vol. 214, p. 109073, Sep. 2022, doi: 10.1016/J.COMNET.2022.109073.

[12] A. Heidari and M. A. Jabraeil Jamali, "Internet of Things intrusion detection systems: a comprehensive review and future directions," Cluster Comput, vol. 26, no. 6, pp. 3753–3780, Dec. 2023, doi: 10.1007/S10586-022-03776-Z/METRICS.

[13] A. Alotaibi and M. A. Rassam, "Adversarial Machine Learning Attacks against Intrusion Detection Systems: A Survey on Strategies and Defense," Future Internet 2023, Vol. 15, Page 62, vol. 15, no. 2, p. 62, Jan. 2023, doi: 10.3390/FI15020062.

[14] J. Malik, R. Muthalagu, and P. M. Pawar, "A Systematic Review of Adversarial Machine Learning Attacks, Defensive Controls, and Technologies," IEEE Access, vol. 12, pp. 99382–99421, 2024, doi: 10.1109/ACCESS.2024.3423323.

[15] Md. T. Hossain, R. Afrin, and Mohd. A.-A. Biswas, "A Review on Attacks against Artificial Intelligence (AI) and Their Defence Image Recognition and Generation Machine Learning, Artificial Intelligence," Control Systems and Optimization Letters, vol. 2, no. 1, pp. 52–59, Feb. 2024, doi: 10.59247/CSOL.V2I1.73.

[16] A. Bajaj and D. K. Vishwakarma, "A state-of-the-art review on adversarial machine learning in image classification," Multimed Tools Appl, vol. 83, no. 3, pp. 9351–9416, Jan. 2024, doi: 10.1007/S11042-023-15883-Z/METRICS.

[17] P. Bountakas, A. Zarras, A. Lekidis, and C. Xenakis, "Defense strategies for Adversarial Machine Learning: A survey," Comput Sci Rev, vol. 49, Aug. 2023, doi: 10.1016/j.cosrev.2023.100573.

[18] K. He, D. D. Kim, and M. R. Asghar, "MTD-AD: Moving Target Defense as Adversarial Defense," IEEE Trans Dependable Secure Comput, 2025, doi: 10.1109/TDSC.2025.3560246.

[19] G. Apruzzese, L. Pajola, and M. Conti, "The Cross-Evaluation of Machine Learning-Based Network Intrusion Detection Systems," IEEE Transactions on Network and Service Management, vol. 19, no. 4, pp. 5152–5169, Dec. 2022, doi: 10.1109/TNSM.2022.3157344.

[20] S. N. Khan, S. U. Khan, H. Aznaoui, C. B. Şahin, and Ö. B. Dinler, "Generalization of linear and non-linear support vector machine in multiple fields: a review," Computer Science and Information Technologies, vol. 4, no. 3, pp. 226–239, Nov. 2023, doi: 10.11591/CSIT.V4I3.PP226-239.

[21] M. Waqas, S. Tu, Z. Halim, S. U. Rehman, G. Abbas, and Z. H. Abbas, "The role of artificial intelligence and machine learning in wireless networks security: principle, practice and challenges," Artif Intell Rev, vol. 55, no. 7, pp. 5215–5261, Oct. 2022, doi: 10.1007/S10462-022-10143-2/METRICS.

[22] O. H. Abdulganiyu, T. Ait Tchakoucht, and Y. K. Saheed, "A systematic literature review for network intrusion detection system (IDS)," International Journal of Information Security 2023 22:5, vol. 22, no. 5, pp. 1125–1162, Mar. 2023, doi: 10.1007/S10207-023-00682-2.