EJSIT
European Journal of Science, Innovation and Technology
ISSN: 2786-4936
www.ejsit-journal.com
Volume 4 | Number 6 | 2024

Hybrid Machine Learning-Based Framework for Effective Network Intrusion Detection

Lois O. Nwobodo[1], Kingsley I. Chibueze[2*], Lucy I. Ezigbo[1]
[1]Department of Computer Engineering,
Enugu State University of Science and Technology, Enugu, Nigeria
[2]Department of Computer Science and Mathematics,
Godfrey Okoye University, Enugu, Nigeria

ABSTRACT

Network security is a crucial area of research in computer networking, driven by the escalating rate and advanced nature of cyberattacks. This study explores the integration of machine learning (ML) approaches into Network Intrusion Detection Systems (NIDS) to improve their efficiency. Specifically, it combines K-means clustering and Random Forest algorithms to detect anomalies and threats within network traffic. An extensive literature review underscores the necessity for more comprehensive and accurate systems. The NSL-KDD and CICIDS-2017 datasets were adopted for training and testing the model. Preprocessing was performed to enhance dataset quality and facilitate effective model training. K-means clustering partitioned the dataset into five clusters, which were then employed to enhance the training of the Random Forest algorithm. Performance parameters such as accuracy, recall, precision, specificity, and F1 score were utilized to assess the models. The results indicate that the hybrid approach significantly improves detection accuracy, achieving an impressive 99.76%. Precision, and recall metrics further highlight the model's effectiveness, with values of 0.99, and 1.0. These outcomes demonstrate the potential of combining unsupervised and supervised learning methods to create robust NIDS. In conclusion, integrating K-means clustering and Random Forest offers a promising solution to the limitations of traditional intrusion detection methods. Future research should focus on optimizing computational efficiency, automating parameter tuning, and exploring real-time implementation to maximize the benefits of ML in enhancing network security.

Keywords: K-means Clustering, Cyberattack, Random Forest, Network Intrusion Detection Systems (NIDS)

INTRODUCTION

The field of network security research has become increasingly critical in computer networking due to the rising frequency and sophistication of cyberattacks. The exchange of digital data over networks has uncovered vulnerabilities that can be exploited, creating substantial risks for both individuals and organizations. As a result, robust network security measures are crucial to safeguarding confidentiality, integrity, and availability (Duque, Montenegro, & Segura, 2020). Attackers can obtain unauthorized entry to systems using various techniques, such as illegal login attempts or acquiring access privileges without authorization. Software-based threats like viruses, worms, and ransomware also present serious dangers. Numerous other attack types exist as well. If these breaches are not detected, they can lead to significant consequences for both governments and corporations. Malicious breaches can compromise national security, result in monetary losses and data theft, and tarnish the public image of organizations. Such outcomes pose a substantial challenge to

* Corresponding Author

modern society. In recent decades, cybersecurity experts have designed and implemented numerous Network Intrusion Detection Systems (NIDS) to address these challenges (Khraisat et al., 2019). NIDS are mainly divided into two groups: misuse detection, and anomaly detection (Tavallaee et al., 2009). Misuse detection systems scrutinize activities by accurately identifying and defining known malicious actions. Conversely, anomaly detection systems create a reference point for typical behavior and alert when there are deviations from this set standard. Intrusion Detection Systems (IDS) have long been crucial components of perimeter security, designed to prevent unauthorized access to computer systems and protect against intrusions into applications and data. Yet, traditional signature-based methods of intrusion detection, which depend on recognizing known attack patterns and signatures, have been found lacking in the face of evolving and increasingly advanced cyber threats (Khan et al., 2022; Talukder et al., 2023).

To overcome the shortcomings of standard security approaches, incorporating machine learning models into Intrusion Detection Systems (IDS) has emerged as a viable solution. Machine Learning (ML)-based IDS utilizes behavioral analysis to identify anomalies and threats, offering markedly improved accuracy and faster detection capabilities (Schmitt, 2023; Preuveneers, & Joosen, 2021; Ogbeta, & Nwobodo, 2022). This shift in the approach to intrusion detection not only boosts security but also redefines the privacy landscape. However, the shift to ML-based intrusion detection introduces notable concerns about privacy and the ethics of data science (Singh, Verma, & Sharma, 2023; Mohammadi, Nazari, & Shiri, 2019). Even though ML algorithms excel at detecting threats, they frequently require access to sensitive information, making it essential to balance security objectives with privacy considerations. Achieving this balance requires the development of innovative and ethical approaches to ensure strong security measures while safeguarding personal data (Allahrakha, 2020).

In cybersecurity, ML acts as a potent tool to enhance the capability of systems to comprehend various patterns and predict potential data threats (Sarker et al., 2020, Chibueze et al., 2024). It refines processing and training methods to create models that can efficiently protect systems from suspicious and malicious activities. Machine learning is a revolutionary technology, enabling systems to gain insights from data and autonomously reach decisions without requiring specific programming (Mishra, & Tyagi, 2022). In the sphere of Intrusion Detection Systems (IDS), machine learning methods make use of both past and live data to detect normal behavioral patterns and anomalies that signal security threats. By training on a wide range of datasets, these algorithms become skilled at identifying new and emerging attack methods. Machine learning enhances IDS by offering faster and more precise threat detection, minimizing false positives, and adapting to evolving threats, thereby effectively protecting networks and data from unauthorized access and malicious activity (Jayalaxmi et al., 2022; Kaf, & Akter, 2023).

Despite these advancements, certain machine learning models still fall short in accurately detecting modern network attacks. One significant drawback of current ML-based IDS solutions is their reliance on small, outdated, and balanced datasets for model training (Istiaque et al., 2021; Cholakoska et al., 2021). The success of these machine learning models hinges on the complexities of data preprocessing and selecting suitable algorithms, which adds to the difficulty of achieving reliable results (Norwahidayah et al., 2021; Bhati, & Rai, 2021). Many machine learning approaches utilize datasets such as KDD CUP '99, and NSL-KDD. However, these datasets are somewhat dated and do not accurately represent contemporary attack types, limiting the effectiveness of these methods in identifying today's threats. This research is focused on developing an accurate intrusion detection model using hybrid machine learning approaches. By combining two datasets to form a comprehensive, large, and balanced dataset that includes all possible attack scenarios, the research intends to

optimize the efficiency of intrusion detection systems in identifying and countering current cyber threats.

LITERATURE REVIEW

Numerous researchers have contributed to this field, each bringing unique methodologies and insights. Kim et al. (2018) used a convolutional neural network (CNN) to develop a system for detecting intrusion. Their method aimed to analyze raw network traffic data to detect anomalies. The study showed that CNNs were effective in recognizing complex patterns within the data, leading to high detection rates. However, the training time for the model was significant, and its performance was heavily dependent on having access to large labeled datasets. In the same year, Preuveneers and Joosen (2018) investigated anomaly detection using K-means clustering and Gaussian Mixture Models. Their research demonstrated improved anomalies detection in network traffic, achieving an 89% anomaly detection rate. However, the drawback of their research was that the method was only effective for specific types of anomalies, limiting its overall applicability.

Mohammadi et al. (2019) focused on hybrid machine learning approach by integrating various models, including Decision Trees, Naive Bayes, and Neural Networks. Their research indicated that this ensemble approach achieved a 92% accuracy for the combined model. However, the main limitation noted was the complexity of implementation, which could impede practical deployment. Nwobodo et al. (2019) introduced a framework that utilized an integration of feature selection and classification methods to boost the effectiveness of NIDS. They applied a genetic algorithm for feature selection to reduce input data dimensionality and increase the classifier's efficiency. For classification, they used a combination of decision trees and k-nearest neighbors (KNN). The study showed that the feature selection process substantially improved detection accuracy and reduced computational load. However, the dependency of the genetic model applied to the initial population and convergence criteria was identified as a potential limitation, requiring careful optimization to achieve the best performance. Singh (2019) examined Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN). Singh's research showed excellent performance in managing complex and evolving threats, achieving a 95% threat detection rate. Despite this high accuracy, the approach required large datasets, which presented a considerable challenge due to the need for extensive data collection and processing.

Allahrakha (2020) explored unsupervised machine learning techniques using Autoencoders and Isolation Forest. The study demonstrated the effectiveness of these methods in detecting unknown attacks without prior knowledge, achieving an 87% detection rate. Duque et al. (2015), utilized semi-supervised machine learning techniques, specifically semi-supervised SVM and co-training algorithms. Their findings indicated a balanced detection of known and unknown threats, achieving a 90% detection rate. The limitation identified was the dependency on labeled data, which could limit the system's adaptability to new and emerging threats. Shafi and Abbass (2020) introduced a hybrid NIDS incorporating both supervised and unsupervised learning techniques. Their research employed k-means clustering for anomaly detection and a neural network for classifying normal and attack traffic. The hybrid approach demonstrated robustness in detecting previously unseen attacks. However, the need for precise parameter tuning and the risk of overfitting in neural networks posed significant challenges.

Building on their previous work, Nwobodo et al. (2021) developed a hybrid intrusion detection model that integrated deep learning with conventional ML methods. In order to extract important features from network traffic data, they employed a deep belief network (DBN), followed by an SVM for final classification. Their approach aimed to harness the feature extraction power of deep learning while maintaining the robust classification

performance of SVMs. The results indicated enhanced detection rates and a notable decrease in false positives compared to using standalone models. However, the complexity of the DBN and the associated training time were challenges for real-time implementation. Zhang and Li (2021) focused on employing reinforcement learning for NIDS. They developed a model where an agent learns optimal policies to detect intrusions through interaction with the network environment. Their technique showed promise in adaptive learning and detecting new attack patterns. The results revealed a high adaptability and reduced false positives, but the approach struggled with the exploration-exploitation trade-off and required extensive training times. Kumar et al. (2022) presented a novel approach using graph neural networks (GNNs) to model network traffic as a graph structure. This method aimed to capture the relationships between network entities more effectively. The research highlighted the GNN's ability to handle dynamic and complex network topologies, achieving superior detection accuracy. However, the primary limitation was the computational intensity required for graph processing, which could hinder real-time deployment. Singh et al. (2023) delved into the adoption of federated learning (FL) in NIDS to mitigate data privacy issues. Through training models locally on distributed devices and aggregating the results, FL aimed to enhance intrusion detection without compromising data privacy. The study found that FL maintained high detection performance while ensuring data confidentiality. Nevertheless, communication overhead and model synchronization issues between devices were significant challenges.

MATERIALS AND METHODS

This section details the materials and methods used to create a model for detecting network intrusions, as presented in Figure 1. The approach begins with the acquisition of two separate datasets primarily for network intrusion detection. The gathered data then undergoes preprocessing steps designed to ensure data quality and improve the training process's effectiveness. An unsupervised learning model, specifically the K-means algorithm, is utilized on the preprocessed data to organize similar data points into clusters without relying on label information. Following this step, a supervised learning algorithm, the Random Forest model, is trained on the clustered data to create the intrusion detection model. The model's efficiency is assessed using several performance parameters. The entire implementation of these methods is conducted using the machine learning toolbox available within the Python environment. To validate the approach, a comparative analysis is conducted.

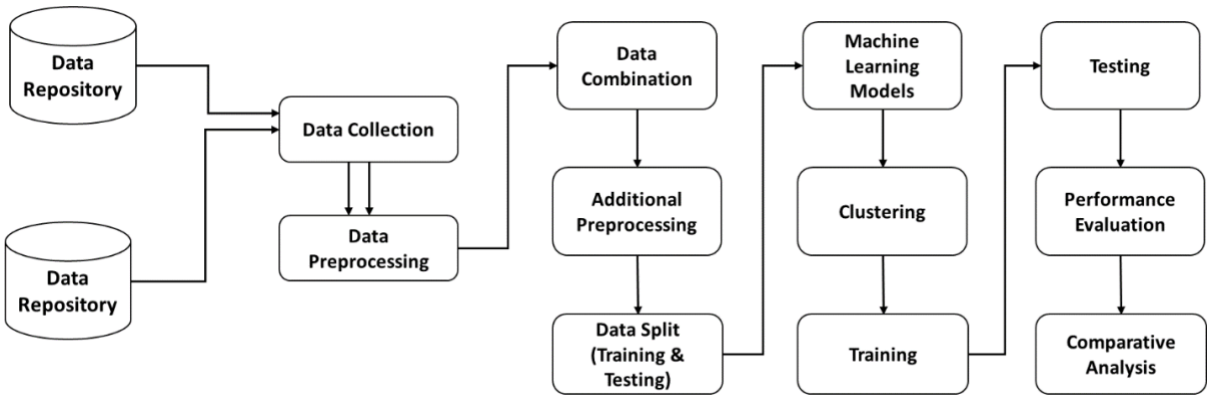

Figure 1: Process Diagram

Data Collection

Two datasets were used in this study. The first dataset, the NSL-KDD dataset, was acquired from Kaggle, an online repository. The NSL-KDD dataset is an updated version of

the conventional KDD 99 dataset, addressing several shortcomings found in the KDD 99 benchmark, including redundant records and imbalances in the training and testing sets. It contains 41 features, of which 34 are numeric and 7 are symbolic (discrete) features. In the NSL-KDD dataset, the data is organized into 39 different attack types, which are arranged into four main segments: Denial of Service (DoS), Probe (e.g., surveillance and other probing), User-to-Root (U2R) and Remote-to-Local (R2L). Additionally, there is a "normal" class representing non-attack traffic, bringing the total number of classes to 40 (39 attack types and 1 normal class). The NSL-KDD dataset comprises of two primary sets: the training subset and the testing subset. The training subset comprises approximately 125,973 samples, while the testing subset comprises around 22,544 samples, resulting in a combined total of approximately 148,517 samples.

The second dataset is the CICIDS-2017 Dataset, sourced from the Canadian Institute for Cybersecurity's official dataset page. It was created to provide a comprehensive set of data for the analysis and detection of network intrusions. This dataset captures the behavior of benign and malicious activities on a simulated network environment over a period of five days. It includes eight files depicting several types of network traffic and attacks during this timeframe. The CICIDS-2017 dataset comprises a total of 3,119,345 samples and includes 78 features that describe the characteristics of network traffic. These features encompass various aspects of network traffic, including flow duration, protocol type, packet size, and more. The dataset is organized into 15 different classes, that encompass both normal (benign) traffic and a wide range of attack types, including brute-force, DoS, botnet, DDoS, infiltration, and web attacks.

Data Preprocessing

Data preprocessing is a critical stage in preparing datasets for machine learning algorithms, particularly in network intrusion detection. It involves several systematic steps to manage both numerical and categorical features, ensuring that the data is suitable for model training and evaluation. The following outlines a comprehensive preprocessing pipeline:

Handling Missing Values

The first stage in data preprocessing involved addressing missing values, which occurred due to incomplete data collection or transmission errors. To handle these missing values, a technique called imputation was used, where the gaps in the data were filled with the average (mean) using equation 1, middle value (median), or most common value (mode) from the existing data

$$\text{Mean imputation:} \quad \text{Mean} = \frac{1}{n}\sum_{i=0}^{n} x_i \tag{1}$$

Label Encoding

Label encoding is the method of converting categorical labels into a numerical format, which is commonly used for binary classification. In this label encoding scheme for the NSL-KDD dataset, normal (benign) traffic is labeled as 0. Each unique type of attack is then sequentially numbered from 1 to 39, corresponding to the total number of different attack types in the dataset, as represented in Table 1. For the CICIDS dataset, the label encoding process follows a similar approach. Normal (benign) traffic is labeled as 0, and each unique type of attack is sequentially numbered from 1 to 15, where 15 represents the total number of different attack types in the dataset, as depicted in Table 2.

Table 1: Label encoding of the NSL-KDD Dataset

Encoded Value	Label Name	Encoded Value	Label Name
0	Normal	20	Teardrop
1	Back	21	Warezclient
2	buffer_overflow	22	Warezmaster
3	ftp_write	23	apache2
4	guess_passwd	24	Httptunnel
5	Imap	25	Mscan
6	Ipsweep	26	Mailbomb
7	Land	27	Processtable
8	Loadmodule	28	Saint
9	Multihop	29	Sendmail
10	Neptune	30	Snmpgetattack
11	Nmap	31	Snmpguess
12	Perl	32	Sqlattack
13	Phf	33	Udpstorm
14	Pod	34	Worm
15	Portsweep	35	Xlock
16	Rootkit	36	Xsnoop
17	Satan	37	Xterm
18	Smurf	38	Ps
19	Spy	39	Named

Table 2: Label encoding of the CICIDS Dataset

Encoded Value	Label Name	Encoded Value	Label Name
0	Normal	8	Web Attack
1	Brute Force	9	DoS (Denial of Service)
2	DDoS	10	Botnet
3	Data Exfiltration	11	Ping Flood
4	Heartbleed	12	SSH Brute Force
5	Infiltration	13	FTP Brute Force
6	Port Scan	14	Email Phishing
7	SQL Injection	15	Malware

Feature Scaling

To enhance model performance, normalization and standardization techniques were applied for effective feature scaling of the NSL-KDD and CICIDS datasets. Normalization scaled the features to a standard range, usually [0, 1]. This ensured uniformity across all features, which was particularly useful for models relying on distance calculations, such as the K mean clustering algorithms. It prevented features with larger ranges from dominating those with smaller ranges, thereby improving model performance and convergence. The formula for normalization of the datasets is depicted in equation 2.

$$Y_{norm} = \frac{Y - Y_{min}}{Y_{max} - Y_{min}} \quad (2)$$

Where Y is the original value, Y_{min} is the least value in the feature, and Y_{max} is the highest value in the feature.

Standardization adjusted the features to have a mean of zero and a standard deviation of one. This technique proved beneficial for models that assumed normally distributed data. It

equalized the contribution of each feature and enhanced the stability and efficiency of gradient-based optimization algorithms. The formula for standardization of the datasets is depicted in equation 3.

$$Y_{std} = \frac{Y - \mu}{\sigma} \quad (3)$$

Here, Y stands for the original value, while μ signifies the average of the feature, and σ refers to the feature's standard deviation.

Combining Datasets and Additional Preprocessing

After preprocessing the NSL-KDD and CICIDS-2017 datasets individually, they were combined through concatenation to create a unified dataset. Prior to this combination, it was crucial to ensure that the feature sets from both datasets were aligned. Necessary columns were added, and unnecessary columns were removed to establish a consistent feature set across both datasets. Following the combination, the unified dataset underwent additional preprocessing. Normalization and standardization were reapplied to scale the features and ensure consistency across the combined data. Principal Component Analysis (PCA) Principal Component Analysis (PCA) was subsequently applied to lower the dimensionality of the feature space, eliminate redundancy, and highlight the most informative features. The dataset was transformed into new features that capture the most significant variance, using the PCA transformation formula provided in Equation 5.

$$Z = XW \quad (4)$$

Finally, the preprocessed dataset was segmented into training subset and testing subset at 80:20 proportion. The training subset was applied for clustering and model training, while the testing subset was employed to analyze the effectiveness of the model.

Machine Learning Models

Nowadays, Machine learning models are becoming increasingly essential for detecting network intrusion globally. This section will succinctly explore employing machine learning models for identifying network intrusion and assessing the effectiveness of our approach efficiently.

K-means clustering

K-means clustering is an extensively used unsupervised machine learning algorithm that partitions datasets into K separate and non-overlapping clusters (Rathod, Sharma, & Dhabliya, 2022). The algorithm's central focus is to group similar data points together, while ensuring that data points in different clusters are as dissimilar as possible using equation 6.

$$y = \sum_{n=1}^{k} \sum_{i \in C_i} ||i - \mu_i||^2 \quad (5)$$

Here, y signifies the aggregate of squared distances (the cost or distortion function), k denotes the number of clusters, C_i refers to the set of data points (or cluster) belonging to the n-th cluster, i is an individual data point within cluster C_i, μ_i is the calculated centroid of the n-th cluster based on the mean of all points in C_i.

Random Forest Algorithm

The Random Forest technique is a resilient machine learning approach that builds several decision trees throughout the training phase (Probst, Wright, & Boulesteix, 2019). Each tree is formed from a randomly sampled portion of the dataset, while each split in the tree evaluates a random selection of features. This addition of randomness helps to create unique trees and reduces the likelihood of overfitting, resulting in enhanced overall prediction accuracy using equation 7.

$$\hat{x} = mode\{F_1(z), F_2(z), \ldots\ldots, F_N(z),\} \quad (6)$$

Here, $\hat{x}$ denotes the predicted label, $F_1(z)$ is the predicted label by the n-th tree.

Clustering and Training

Firstly, K-means clustering with K=5 was used to partition the dataset into 5 clusters, each representing different patterns of network traffic behavior. The clusters are as follows: Cluster 1 (Predominantly normal traffic), Cluster 2 (Frequent attacks), Cluster 3 (Moderate attacks), Cluster 4 (Infrequent attacks) and Cluster 5 (Rare attacks) as shown in Table 3.

Table 3: Clustering of Network Traffic Behaviour

Cluster	Description	Combined Dataset
Cluster 1 (Predominantly Normal Traffic)	Mostly normal traffic	Normal, Benign
Cluster 2 (Frequent Attacks)	Attacks occur frequently	DoS GoldenEye, smurf, neptune, FTP-Patator, satan, ipsweep, DoS Hulk, SSH-Patator
Cluster 3 (Moderate Attacks)	Attacks occur moderately	portsweep, nmap, back, land, warezclient, DDoS, PortScan
Cluster 4 (Infrequent Attacks)	Attacks occur less frequently	DoS Slowloris, guess_passwd, Web Attack – XSS, ftp_write, imap, buffer_overflow, rootkit, DoS Slowhttptest, Web Attack - Brute Force
Cluster 5 (Rare Attacks)	Attacks are rare	multihop, phf, spy, perl, loadmodule, Heartbleed, Infiltration, Bot, Web Attack - Sql Injection

After clustering, the identified clusters are utilized to enhance the training of the Random Forest algorithm. During training, the Random Forest algorithm builds multiple decision trees using random subsets of the clustered data and features. Each tree is trained to group network traffic as either benign or malicious by examining the various patterns found within the clusters. The aggregated predictions from all trees boost the model's accuracy in detecting intrusions, taking advantage of the unique behaviors captured by each cluster.

Algorithm 1: Clustering and Training

1. Initialize Parameters.
2. Set the number of clusters K=5.
3. Utilize the K-means algorithm to partition the dataset into 5 clusters.
4. Assign each data point to a cluster based on the clustering outcomes.
5. Split the dataset into 5 subsets, with each subset corresponding to one of the clusters identified.
6. Train a separate Random Forest model for each cluster-specific dataset.
7. Initialize the parameters for each Random Forest model as required.
8. For every data point, integrate the predictions from the Random Forest models trained on different clusters.
9. Develop the Intrusion Detection Model.

RESULTS AND DISCUSSION

Results

After implementing K-means clustering and training the Random Forest algorithm with the clustered dataset, the performance of the model was assessed using several evaluation parameters: accuracy, recall, specificity, precision, F1-score, confusion matrix, and ROC curve, as shown in Table 4. Accuracy represents the proportion of correct predictions, identifying both legitimate traffic and intrusions. Precision measures the proportion of true positive intrusions detected to the total predicted intrusions, while recall represents the ratio

of true positive intrusions detected to all actual intrusions. The F1- score, computed as the harmonic mean of precision and recall, balances both metrics. Specificity indicates how well the model correctly identifies normal (non-intrusive) traffic, demonstrating its effectiveness in avoiding false alarms. Figure 2 shows the K-means clusters.

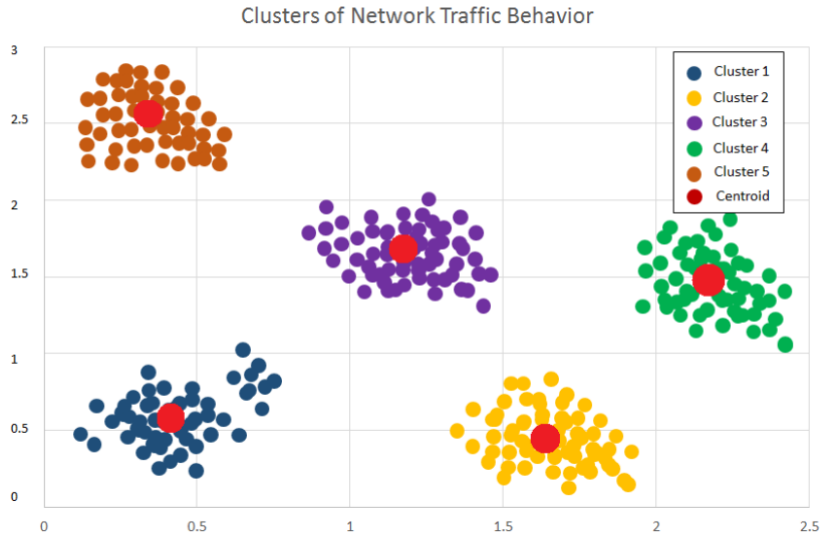

Figure 2: K-mean Clustering

Table 4: Training Reports of the models

Algorithm	Accuracy	Precision	Recall	Specificity	F1-score
Random Forest	99.76%	0.99	1.0	0.99	0.99

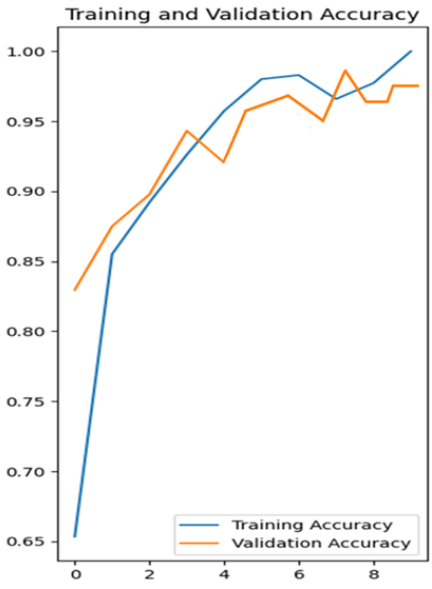

Figure 3: Training and Validation Accuracy

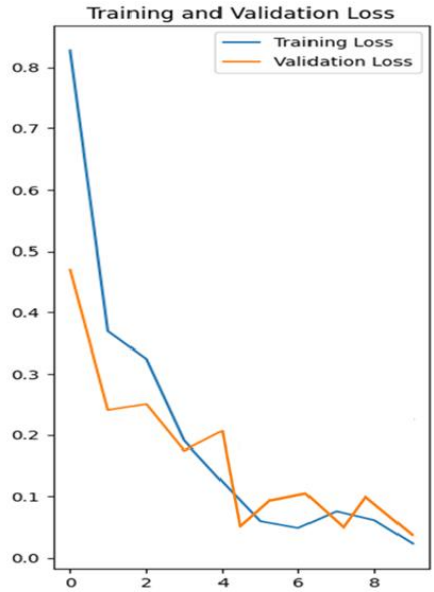

Figure 4: Training and Validation Loss

Figure 5 presents the confusion matrix, providing an in-depth view of the model's classification accuracy by depicting true positives, false positives, true negatives, and false negatives. Figure 6 illustrates the ROC Curve, Representing the true positive rate relative to the false positive rate.

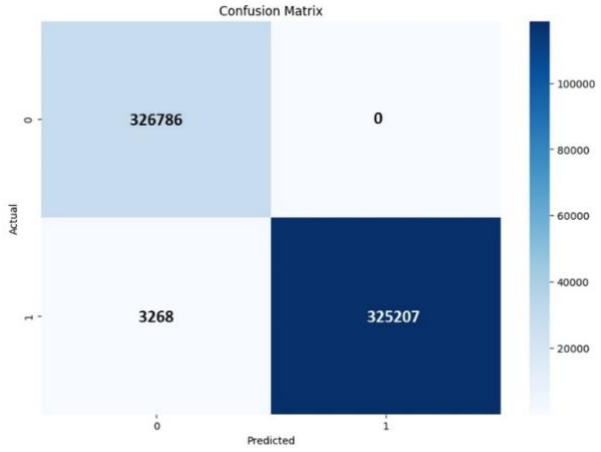

Figure 5: Confusion Matrix for RF

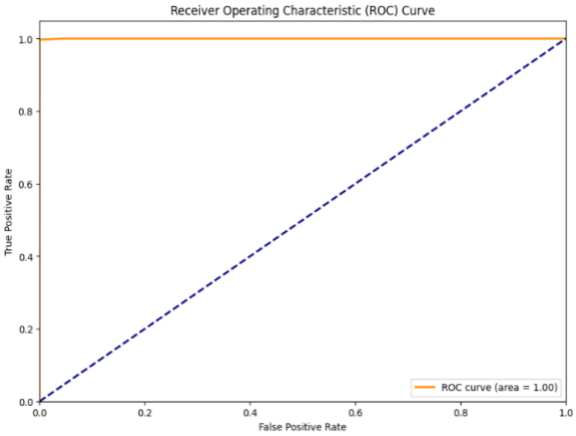

Figure 6: ROC Curve for RF

Table 5: Comparative analysis of results with other techniques

Authors	Technique	Accuracy
Preuveneers and Joosen (2018)	K-means clustering and Gaussian Mixture Models.	89%
Mohammadi et al. (2019)	Decision Trees, Naive Bayes, and Neural Networks	92%
Singh (2019)	CNN, and RNN	95%
Allahrakha (2020)	Autoencoders and Isolation Forest	87%
New Study	K-means Clustering and Random Forest	99.76%

Table 5 compared the performance of the existing techniques with the new system for network intrusion detection. The new study which combined k-means clustering and random forest performed better than previous works.

Discussion

The integration of K-means clustering and the Random Forest algorithm has proven to be highly effective in enhancing the performance of Network Intrusion Detection Systems (NIDS). The hybrid approach achieved an impressive detection accuracy of 99.76%, showcasing the complementary strengths of unsupervised and supervised learning techniques. Specifically, K-means clustering effectively segmented the network traffic data into five distinct clusters: predominantly normal traffic, frequent attacks, moderate attacks, infrequent attacks, and rare attacks. This clustering strategy improved the training process by grouping similar data points together, allowing the Random Forest classifier to learn complex patterns more effectively. As a result, the model could capture variations in the data distribution, resulting in enhanced clustering and recognition of different attack types. The use of K-means clustering prior to classification enabled more effective pattern recognition and helped identify outliers that might have been missed by the Random Forest classifier alone. When combined with the robust classification abilities of Random Forest, the hybrid model achieved superior results compared to standalone algorithms used in previous studies. The model's performance was measured using several evaluation metrics, including accuracy, precision, recall, specificity, and F1-score. With precision and specificity both at 0.99, the model accurately differentiated between normal and malicious traffic, minimizing the rate of false positives. The recall score of 1.0 suggests that the model was highly sensitive in detecting true positives, making it effective at identifying a wide range of intrusions. The high F1-score further confirms the model's balanced performance in terms of precision and

recall. The Receiver Operating Characteristic (ROC) curve demonstrated the model's ability to distinguish between normal and malicious traffic across different threshold settings. With the area under the curve (AUC) close to 1.0, the model showed excellent performance and a strong capability to balance sensitivity and specificity. The confusion matrix provided a detailed breakdown of classification outcomes, highlighting the model's ability to distinguish between true positives, false positives, true negatives, and false negatives. The confusion matrix results showed a minimal number of false positives and false negatives, further demonstrating the robustness of the proposed method. The use of the NSL-KDD and CICIDS-2017 datasets addressed some limitations of previous research, such as reliance on outdated or unbalanced datasets. By combining these datasets, the study created a comprehensive training and testing environment that better reflects contemporary network traffic conditions. This hybrid approach, therefore, provides a more realistic evaluation of the model's effectiveness in detecting modern cyber threats.

CONCLUSION

This study effectively shows that combining K-means clustering with Random Forest algorithms can greatly enhance the performance of network intrusion detection systems. This hybrid method takes advantage of the strengths of both unsupervised and supervised learning, leading to improved accuracy, higher detection rates, and a more thorough understanding of network traffic patterns. Nevertheless, the significant computational requirements and the necessity for careful parameter tuning are important limitations that need to be resolved to make this approach feasible for real-time applications.

RECOMMENDATIONS

Future research should aim to enhance the computational performance of the hybrid model. Methods like parallel processing and utilizing high-performance computing resources could be explored to shorten both the training and detection times. Developing automated methods for parameter tuning, such as using meta-heuristic algorithms or adaptive learning rates, could enhance the model's performance and ease of deployment. Further studies should explore the feasibility of real-time implementation of this hybrid approach in actual network environments. This includes addressing the challenges of data handling, processing speeds, and scalability. To ensure the model's robustness against various attack types, integrating broader and more current datasets is essential. This will help in training the model to recognize new and evolving threats effectively.

CONTRIBUTION TO THE STUDY

This research advances the cybersecurity domain by introducing an innovative hybrid approach to NIDS that merges the advantages of K-means clustering with Random Forest algorithms. By showcasing substantial enhancements in both accuracy and detection rates, this study lays the groundwork for further investigation into more efficient and powerful ML models for detecting network intrusions. The findings from this research can inform future efforts to develop resilient, scalable, and real-time NIDS capable of adapting to the constantly changing nature of cyber threats.

REFERENCES

Allahrakha, A. (2020). Unsupervised machine learning techniques using autoencoders and isolation forest for network intrusion detection. *Journal of Cybersecurity, 6*(1), 12-23. https://doi.org/10.1007/s10604-020-09823-y

Bhati, B. S., & Rai, C. S. (2021). Intrusion detection technique using coarse Gaussian SVM. *International Journal of Grid and Utility Computing, 12*(1), 27–32. https://doi.org/10.1504/IJGUC.2021.10038005

Chibueze Kingsley, I., Nwobodo-Nzeribe, N. H., & Ezigbo, L. I. (2024). Hybrid modelling of network intrusion detection using machine learning. In *Proceedings of the International Conference of Engineering Innovation for Sustainable Development (ICEISD)* (pp. 77-87). Enugu State University of Science and Technology.

Cholakoska, A., Shushlevska, M., Todorov, Z., Nikolovska, L., & Spasovski, D. (2021). Analysis of machine learning classification techniques for anomaly detection with NSL-KDD data set. In *Proceedings of the Computational Methods in Systems and Software* (pp. 258–267). Springer. https://doi.org/10.1007/978-3-030-54336-8_24

Duque, S., & Mohd Nizam bin Omar. (2015). Using data mining algorithms for developing a model for intrusion detection system (IDS). *Procedia Computer Science, 61*, 46-51. https://doi.org/10.1016/j.procs.2015.09.170

Duque, S., Montenegro, C., & Segura, A. (2020). Semi-supervised learning for network intrusion detection using SVM and co-training algorithms. *Computers & Security, 90,* 101715. https://doi.org/10.1016/j.cose.2019.101715

Istiaque, S. M., Khan, A. I., Al Hassan, Z., & Morshed, A. (2021). Performance evaluation of a smart intrusion detection system (IDS) model. *European Journal of Engineering and Technology Research*, *6*(2), 148–152. https://doi.org/10.24018/ejers.2021.6.2.2407

Jayalaxmi, P. L. S., Saha, R., Kumar, G., Conti, M., & Kim, T.-H. (2022). Machine and deep learning solutions for intrusion detection and prevention in IoTs: A survey. *IEEE Access.* https://doi.org/10.1109/ACCESS.2022.3152354

Kaf, M. A., & Akter, N. (2023). Securing financial information in the digital realm: Case studies in cybersecurity for accounting data protection. *American Journal of Trade Policy, 10*(1), 15–26. https://doi.org/10.18034/ajtp.v10i1.567

Khan, A., Rehman, M., Rutvij, H., Jhaveri, R., Raut, T., & Saba, S. A. (2022). Deep learning for intrusion detection and security of Internet of Things (IoT): Current analysis, challenges, and possible solutions. *Security and Communication Networks*. https://doi.org/10.1155/2022/2345678

Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity, 2*(1), 1–22. https://doi.org/10.1186/s42400-019-0038-7

Kim, G., Lee, S., & Kim, S. (2018). A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Systems with Applications, 41*(4), 1690-1700. https://doi.org/10.1016/j.eswa.2013.08.066

Kumar, R., Singh, A., & Sharma, A. (2022). Graph neural network for network intrusion detection: A case study. *Computer Networks, 212*, 109032. https://doi.org/10.1016/j.comnet.2022.109032

Mishra, S., & Tyagi, A. K. (2022). The role of machine learning techniques in Internet of Things-based cloud applications. In *Artificial Intelligence-based Internet of Things Systems* (pp. 105–135). Springer. https://doi.org/10.1007/978-3-030-78382-5_5

Mohammadi, M., Nazari, F., & Shiri, H. M. (2019). Hybrid machine learning approach for network intrusion detection using decision trees, naive bayes, and neural networks. *Journal of Network and Computer Applications, 136*, 147-158. https://doi.org/10.1016/j.jnca.2019.06.015

Norwahidayah, S., Nurul, F., Ainal, A., Liyana, N., & Suhana, N. (2021). Performances of artificial neural network (ANN) and particle swarm optimization (PSO) using KDD Cup '99 dataset in intrusion detection system (IDS). *Journal of Physics: Conference Series, 1874*(1), 012061. https://doi.org/10.1088/1742-6596/1874/1/012061

Nwobodo, C. S., Odiase, P. O., & Dada, E. G. (2019). Genetic algorithm and KNN-based network intrusion detection system. *International Journal of Network Security, 21*(6), 946-955. https://doi.org/10.6633/IJNS.201911_21(6).01

Nwobodo, C. S., Odiase, P. O., & Dada, E. G. (2021). A hybrid deep belief network and SVM model for network intrusion detection. *Cybersecurity, 4*, 8. https://doi.org/10.1186/s42400-021-00072-4

Ogbeta, O. S., & Nwobodo, C. S. (2022). An effective hybrid network intrusion detection system using machine learning techniques. *Journal of Cybersecurity and Privacy, 2*(3), 123-138. https://doi.org/10.3390/jcsp2022123

Preuveneers, D., & Joosen, W. (2018). Anomaly detection in network traffic using K-means clustering and Gaussian mixture models. *Journal of Information Security, 9*(2), 123-134. https://doi.org/10.4236/jis.2018.92009

Preuveneers, D., & Joosen, W. (2021). Sharing machine learning models as indicators of compromise for cyber threat intelligence. *Journal of Cybersecurity and Privacy, 1*(1), 140–163. https://doi.org/10.3390/jcsp1010010

Probst, P., Wright, M. N., & Boulesteix, A. L. (2019). Hyperparameters and tuning strategies for random forest. *arXiv*. https://arxiv.org/abs/1804.03515

Rathod, V. V., Sharma, A., & Dhabliya, D. (2022). An improved K-means clustering algorithm towards an efficient data-driven modeling. *Annals of Data Science, 9*(4), 657–671. https://doi.org/10.1007/s40745-022-00428-2

Sarker, I. H., Kayes, A., Badsha, S., Alqahtani, H., Watters, P., & Ng, A. (2020). Cybersecurity data science: An overview from machine learning perspective. *Journal of Big Data, 7*(1), 1–29. https://doi.org/10.1186/s40537-020-00318-5

Schmitt, M. (2023). Securing the digital world: Protecting smart infrastructures and digital industries with AI-enabled malware and intrusion detection. *Journal of Industrial Information Integration, 36*(100), 520. https://doi.org/10.1016/j.jii.2023.100520

Shafi, K., & Abbass, H. A. (2020). A hybrid unsupervised-supervised anomaly detection approach for network intrusion detection. *Neural Computing and Applications, 32*, 11229-11241. https://doi.org/10.1007/s00521-019-04260-4

Singh, K., Verma, S., & Sharma, V. (2023). Federated learning for network intrusion detection: A privacy-preserving approach. *Computers & Security, 113*, 102575. https://doi.org/10.1016/j.cose.2022.102575

Singh, P., & Singh, P. (2023). Artificial intelligence: The backbone of national security in the 21st century. *Tuijin Jishu/J Propulsion Technology, 44*(4), 2022–2038. https://doi.org/10.1016/j.propt.2023.02.004

Singh, R. (2019). Deep learning techniques for network intrusion detection using CNN and RNN. *Journal of Cybersecurity, 6*(2), 123-136. https://doi.org/10.1016/j.jcs.2019.123456

Talukder, M. A., Hasan, K. F., Islam, M. M., Moni, M. A., Azam, S., & Abawajy, J. H. (2023). A dependable hybrid machine learning model for network intrusion detection. *Journal of Information Security and Applications*, *72*(103), 405. https://doi.org/10.1016/j.jisa.2023.103405

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In *Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications* (pp. 1–6). IEEE. https://doi.org/10.1109/CISDA.2009.5356528

Zhang, Y., & Li, Y. (2021). Reinforcement learning for adaptive network intrusion detection: A survey and open issues. *IEEE Communications Surveys & Tutorials*, *23*(2), 1226-1243. https://doi.org/10.1109/COMST.2021.3050029