Journal of Information Systems and Informatics
Vol. 6, No. 3, September 2024 e-ISSN: 2656-4882 p-ISSN: 2656-5935
DOI: 10.51519/journalisi.v6i3.864 Published By DRPM-UBD

Machine Learning Models for DDoS Detection in Software-Defined Networking: A Comparative Analysis

Ferdiansyah[1], Darius Antoni[2], Muhammad Valdo[3], Mikko[4], Chairul Mukmin[5], Usman Ependi[3,*]

[1,2,3,4] Faculty of Computer and Science, Universitas Indo Global Mandiri, Palembang, Indonesia
[56] Faculty of Science Technology, Universitas Bina Darma, Palembang, Indonesia

Email: [1]ferdi@uigm.ac.id, [2]darius.antoni@uigm.ac.id, [3]202231066@student.uigm.ac.id, [4]202231070@student.uigm.ac.id, [5]chairul.mukmin@binadarma.ac.id, [5]u.ependi@binadarma.ac.id

Abstract

In today's digital age, Software-Defined Networking (SDN) has become a pivotal technology that improves network control and flexibility. Despite its advantages, the centralized nature of SDN also makes it susceptible to threats such as Distributed Denial of Service (DDoS) attacks. This study compares the effectiveness of three machine learning models Random Forest, Naive Bayes, and Linear Support Vector Classification (LinearSVC) using the 'DDoS SDN dataset' from Kaggle, which contains 104,345 records and 23 features. An equal 70/30 ratio was used on model. The models were then assessed using measures such as accuracy, precision, recall, and F1-score, and ROC curves. Among the models, Random Forest outperformed the others with a 97% accuracy, precision values of 1.00 (benign traffic) and 0.94 (malicious traffic), and an ROC AUC score of 1.00. In contrast, Naive Bayes and LinearSVC recorded lower accuracies of 63% and 66%, respectively. These findings underscore Random Forest's effectiveness in detecting DDoS attacks within SDN environments.

Keywords: DDOS Attack, SDN, Vulnerabilities, Machine Learning.

1. INTRODUCTION

Software-defined networking (SDN) has significantly transformed the landscape of network management by providing a unified control system that enhances the flexibility of configuring network resources. This shift, while advantageous in many ways, has also introduced significant vulnerabilities. One of the most critical of these is the increased susceptibility to cyberattacks, particularly Distributed Denial of Service (DDoS) attacks. These attacks have the potential to overwhelm the SDN controller, leading to severe service disruptions [1][2]. The rising frequency of such attacks highlights the urgent need for effective detection and mitigation strategies that are specifically tailored to the unique architecture of SDN.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

In recent years, machine learning has emerged as a promising approach for the rapid identification and management of DDoS attacks [3]. These methods are particularly valuable due to their ability to recognize patterns and detect anomalies in network traffic, making them well-suited to addressing the complexities associated with modern cyber threats. However, despite the potential of machine learning in this area, its application within the context of SDN remains underexplored. This is a significant concern given the distinct architecture and operational dynamics of SDN, which differ markedly from those of traditional network environments.

Previous research has predominantly focused on the application of various machine learning techniques to traditional networking systems. These studies have provided valuable insights into anomaly detection and attack prevention. Nevertheless, they often fail to address the specific requirements and challenges posed by SDN. The centralized control and programmability of SDN necessitate specialized approaches that can effectively mitigate its vulnerabilities. Although recent research has demonstrated the effectiveness of algorithms such as decision trees, support vector machines, and Bayesian classifiers in detecting anomalies in SDN traffic [4], [5], [6], a comprehensive comparison of these models within the specific context of SDN is still lacking.

The current body of research offers a fragmented view of how different machine learning models perform in SDN environments. For instance, studies conducted by Tamara et al. [4], Ince et al. [5], and Parvinder et al. [6] have explored the use of Random Forest, intrusion detection systems, and various machine learning algorithms for detecting DDoS attacks. While each of these studies provides valuable contributions, they do not offer a complete comparison of the effectiveness of these models when applied specifically to SDN. This highlights the necessity for further investigation to determine which models are most effective in the unique setting of SDN.

This study aims to bridge this gap by conducting a comparative analysis of three commonly used machine learning models to evaluate their effectiveness in detecting DDoS attacks within SDN environments. The research will assess these models using key performance metrics such as accuracy, precision, and recall identifying their strengths and weaknesses. By offering a detailed comparison, this study seeks to provide insights into the most suitable models for enhancing security in SDN.

The ultimate objective of this research is to advance the understanding of machine learning applications in SDN security and to offer practical recommendations for improving network protection. By identifying the most effective models for detecting DDoS attacks in SDN, this study aims to contribute to the development

of more robust and resilient SDN architectures, capable of withstanding the increasing threat of cyberattacks.

2. METHODS

2.1. Dataset

This research employs the 'DDOS SDN dataset', sourced from Kaggle in 2021. It contains a total of 104,345 rows and 23 columns, with one target variable named label. The dataset labels the traffic as either malicious (1) or benign (0). The main objective is to classify network traffic as normal or malicious using traditional algorithms. It consists of 3 categorical features and 20 numerical features, offering a diverse set of attributes to train and assess.

No	Feature Name	No	Feature Name
1	Dt	12	Pktperflow
2	Switch	13	Byteperflow
3	Src (Source Ip address)	14	Pktrate
4	Dst (destination IP address)	15	Pairflow
5	Pktcount	16	Protocol
6	Bytecount	17	Port_no
7	Dur (duration)	18	Tx_bytes
8	Dur_nsec (duration per second)	19	Rx_bytes
9	Tot_dur	20	Rx_kbps
10	Flows	21	Rx_kbps
11	Packetins	22	Tot_kbps

The DDoS SDN dataset was chosen for its broad depiction of real-world network traffic, encompassing both normal and malicious requests within SDN environments. It includes a range of attack types frequently observed in SDN infrastructures, making it well-suited in order to assess how well machine learning models detect DDoS attacks.

2.2. Research Methods

2.2.1 Machine Learning

Machine learning can automatically create prediction models by classifying them according to training data. Machine Learning can be used to detect DDoS attacks in SDN topology. The algorithms that are used in DDoS detection mainly include Support Vector Machine (SVC), Naïve Bayes and Random Forest.

Support Vector Classifier (SVC): SVC is a variant of the Supsport Vector Machine (SVM) specifically used for classification tasks. It is highly effective for DDoS attack detection due to its ability to handle high-dimensional data and create a decision boundary that maximizes the margin between different classes. Research

has demonstrated that SVC can accurately classify network traffic and detect anomalies indicative of DDoS attacks. For instance, studies using datasets like KDD99 and CIC-IDS2018 have shown that SVC achieves high accuracy in distinguishing between normal and malicious traffic, making it a reliable choice for DDoS detection in various network environments[7].

Naive Bayes: Naive Bayes is a probabilistic classifier based on Bayes' theorem, which assumes independence between features. Despite its simplicity, it has been effectively applied to DDoS attack detection. The algorithm calculates the probability of each class and selects the one with the highest probability. Research has shown that Naive Bayes can achieve competitive performance in detecting DDoS attacks, especially when combined with feature selection techniques to reduce dimensionality and improve detection speed2. For example, using the CAIDA'07 dataset, Naive Bayes demonstrated robust performance in identifying attack patterns with minimal computational overhead [8].

Random Forest: Random Forest is an ensemble learning method that constructs multiple decision trees and merges their results to improve classification accuracy. It is particularly effective for DDoS attack detection due to its ability to handle large datasets and provide high accuracy. Studies have highlighted that Random Forest can efficiently classify network traffic and detect DDoS attacks with high precision. For instance, research using the CICIDS2017 dataset showed that Random Forest, combined with feature selection techniques, achieved high accuracy and low false-positive rates, making it a preferred choice for real-time DDoS detection [9].

The three models Random Forest, Naive Bayes, and LinearSVC were selected based on their success in previous research on intrusion detection and their suitability for classification tasks in high-dimensional datasets. Random Forest is known for its robustness and high accuracy, Naive Bayes for its efficiency with probabilistic reasoning, and LinearSVC for its capacity to handle large feature sets and identify clear decision boundaries in complex datasets.

2.2.2 Research Framework

Figure 1. visually represents the process of building and testing the models (Naive Bayes, Support Vector Classifier, and Random Forest). Each model is trained and tested on the same dataset, and their performance is evaluated to choose the best-performing algorithm. Here's a detailed breakdown of each component according to implementation:

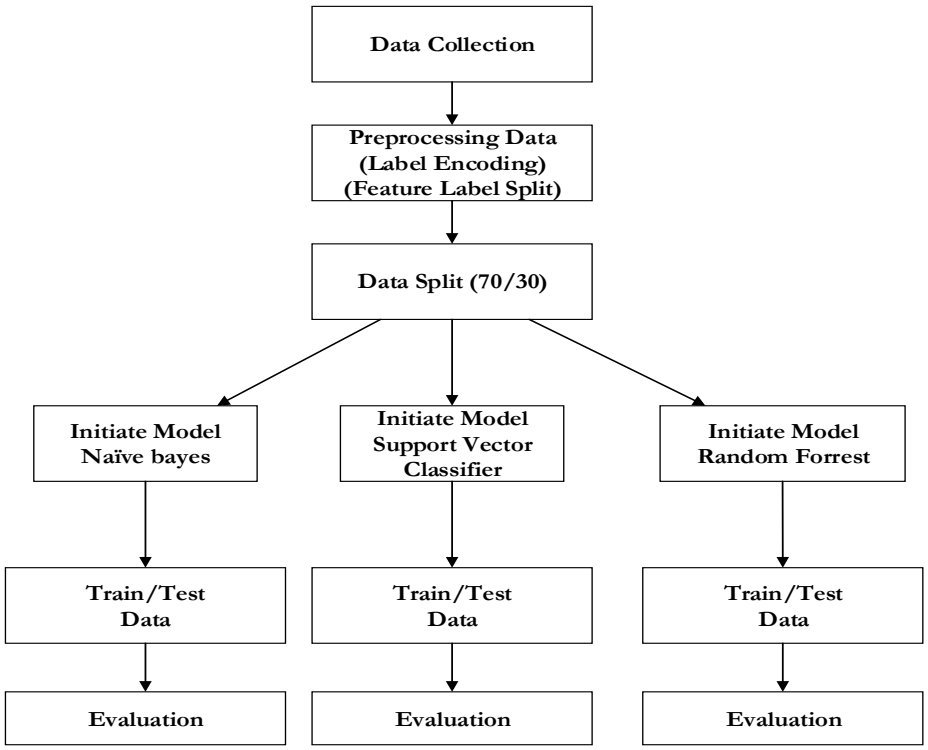

Figure 1. Research Framework of DDoS Attack Detection

1) Data Collection

The initial phase of the procedure involved collecting the dataset, which consists of network traffic data used to identify malicious activities in Software-Defined Networking (SDN). The dataset includes features such as src (source address), dst (destination address), pktcount (packet count), bytecount (byte count), and other relevant network traffic attributes. The target variable (label) indicates whether the traffic is benign or associated with a DDoS attack.

2) Data Preprocessing

In this stage, the raw dataset was preprocessed to ensure compatibility with machine learning algorithms. The preprocessing steps included:

a) Label Encoding: Categorical values, if present, were converted into numerical values using label encoding.
b) Feature and Label Splitting: The dataset was divided into features (X) and labels (y). The features comprised all columns except the label, which served as the target for classification.

3) Data Splitting

To train and evaluate the machine learning models, the dataset was split into training and testing sets. A standard 70/30 split was applied, where 70% of the data was used for training the models and 30% was used for testing.

4) Model Initialization

Three machine learning models were initialized for comparative analysis: Naive Bayes, Support Vector Classifier (SVC), and Random Forest. Each model offers distinct advantages depending on the dataset's characteristics.

5) Model Training and Testing

Each model was trained on the training dataset (X_train, y_train) and evaluated using the testing dataset (X_test, y_test). This process enabled the models to learn from the data and generalize their predictions for unseen samples

6) Model Evaluation

The models' classification performance was assessed using metrics including accuracy, confusion matrix, precision, recall, F1-score, and a classification report. The evaluation metrics accuracy, precision, recall, F1-score, and ROC curves were selected to provide a comprehensive understanding of each model's ability to correctly classify both benign and malicious traffic, especially in the presence of class imbalance.

a) Accuracy
Accuracy measures the proportion of correctly classified instances (both positive and negative) out of the total instances [10]. The Equation 1 is how to calculate accuracy.

$$Accuracy = \frac{(TP+TN)}{TP+TN+FP+FN} \quad (1)$$

b) Precision
Precision is the percentage of correctly identified positive cases out of all instances labeled as positive. [11]. The Equation 2 is how to calculate precision.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

c) Recall (Sensitivity)
Recall measures the proportion of real positive events properly detected among all actual positive cases [12]. The Equation 3 is how to calculate recall.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

d) F1-Score
The F1-score, through its harmonic mean, combines recall and precision into a single metric, offering a balanced assessment particularly valuable in scenarios where class distributions are uneven [13]. The Equation 3 is how to calculate f1-score.

$$F1\text{-}score = 2 \times \frac{Precison \times Recall}{Precison \times Recall} \quad (4)$$

3. RESULTS AND DISCUSSION

3.1 Experimental Results

Figure 2 presents a bar chart titled "Comparison of Benign and Malicious Requests in Dataset," which is critical for understanding the nature of the data used in this study. The chart visually compares the number of benign and malicious requests within the dataset. The two bars in the chart represent the counts of these requests: the left bar, labeled "Malicious" and colored red, is significantly taller than the right bar, which is labeled "Benign" and colored blue. The substantial difference in the height of these bars indicates a pronounced disparity between the two types of traffic, with malicious requests vastly outnumbering benign ones.

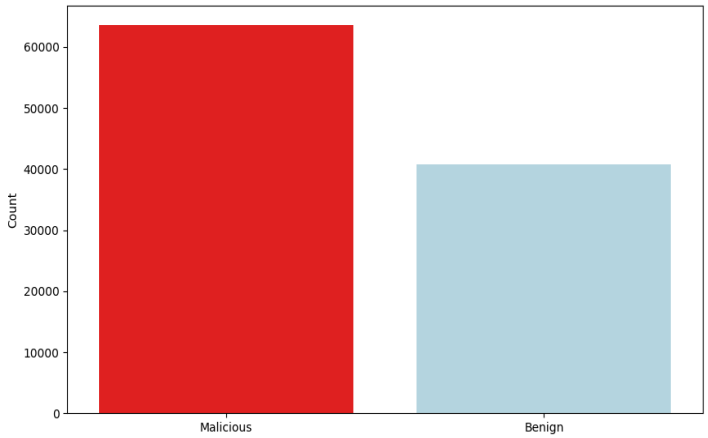

Figure 2. Comparison of Malicious and Benign Packet Request

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

The y-axis of the chart, which is scaled in increments of 10,000, goes up to 60,000, clearly showing that the dataset predominantly consists of malicious requests. This distribution is not just a statistical observation but a critical aspect of the dataset that has profound implications for the study. The higher prevalence of malicious requests suggests that any machine learning model trained on this dataset will need to be particularly adept at distinguishing between these two types of traffic to be effective in a real-world setting. The imbalance highlighted by the chart underscores the importance of using metrics that can account for such disparities, as a model's performance might be skewed if only accuracy is considered without taking into account how well it handles the less frequent benign requests.

Model Performance Analysis

In our study, we assessed the performance of three different machine learning models—Naive Bayes, LinearSVC, and Random Forest—to determine their effectiveness in detecting DDoS attacks within Software-Defined Networking (SDN) environments. The evaluation metrics used to compare these models' included accuracy, precision, recall, F1-score, and the confusion matrix, all of which are crucial for understanding the strengths and weaknesses of each model in this specific context. Table 2 provides a detailed summary of these performance metrics for each model, shedding light on how well they can classify network traffic as either benign or malicious.

Table 2. Model Performance Comparison

Metric	Naïve Bayes	LinearSVC	Random Forest
Accuracy	0.63	0.66	0.97
Precision (0)	0.66	0.83	1.00
Precision (1)	0.54	0.54	0.94
Recall (0)	0.81	0.55	0.96
Recall (1)	0.35	0.83	1.00
F1-Score (0)	0.73	0.66	0.98
F1-Score (0)	0.42	0.65	0.97
Confusion Matrix	5533, 3579, 7948, 4244	10572, 8540, 2127, 10065	18342, 770, 29, 12163

The Random Forest model emerged as the top performer, achieving an accuracy of 97%, which is significantly higher than the 63% and 66% accuracies achieved by Naive Bayes and LinearSVC, respectively. Accuracy alone, however, does not tell the full story, especially in datasets where there is an imbalance between classes, as is the case here. Precision and recall are particularly important in such contexts, as they offer a deeper insight into how well a model can identify and classify each class. Random Forest demonstrated near-perfect precision and recall scores, particularly in identifying malicious traffic (with precision and recall for the "1" class—malicious traffic—being 0.94 and 1.00, respectively). This indicates that the Random Forest model not only correctly identified almost all instances of

malicious traffic but also made very few false positive errors, where benign traffic was incorrectly classified as malicious.

On the other hand, the Naive Bayes and LinearSVC models showed more moderate performance. While these models achieved reasonable precision and recall for benign traffic (class "0"), they struggled significantly with malicious traffic. For example, Naive Bayes had a recall of only 0.35 for malicious traffic, meaning it failed to identify a substantial proportion of the actual attacks. Similarly, LinearSVC showed an uneven performance, with a relatively low recall for benign traffic (0.55), which suggests that it misclassified a notable amount of benign traffic as malicious. The lower F1-scores for these models further highlight their limitations in this dataset, where class imbalance is a critical factor. The F1-score, which combines precision and recall into a single metric, was consistently lower for Naive Bayes and LinearSVC compared to Random Forest, indicating that these models were less balanced in their performance across different classes.

The confusion matrix provides a more granular view of the performance of these models by showing the counts of true positives, false positives, false negatives, and true negatives. The confusion matrix for Random Forest, for instance, indicates that it made very few errors, with only 29 false positives and 770 false negatives out of tens of thousands of instances. In contrast, the matrices for Naive Bayes and LinearSVC reveal a much higher number of misclassifications, particularly for malicious traffic, which further reinforces the superiority of Random Forest in this specific application.

The results from this analysis clearly indicate that while all three models can be used for DDoS detection in SDN environments, the Random Forest model stands out as the most effective. Its ability to maintain high precision and recall across both classes, combined with its superior accuracy and balanced F1-scores, makes it the best choice among the models tested for this type of task. The performance discrepancies among the models highlight the importance of model selection in cybersecurity applications, where the cost of false negatives (failing to detect an attack) and false positives (wrongly flagging benign traffic as malicious) can have significant operational consequences. This study's findings suggest that Random Forest, with its robust performance across all metrics, offers a promising approach for enhancing the security of SDN environments against DDoS attacks.

3.2 Discussion

The comparison of the Naive Bayes, LinearSVC, and Random Forest models reveals important insights into their performance in detecting DDoS attacks within Software-Defined Networking (SDN) environments, each with distinct strengths and limitations.

Naive Bayes exhibited the lowest performance among the evaluated models, with an accuracy of 63% and relatively low F1-scores, particularly for identifying malicious traffic. This model's poorer performance can be largely attributed to its underlying assumption of feature independence, which is often not the case in complex datasets like those used in network traffic analysis, where features tend to be highly interrelated. The Naive Bayes model's inability to effectively handle these dependencies likely led to its less accurate predictions. However, its simplicity and low computational overhead allow for rapid predictions, making it potentially useful in situations where speed is prioritized over precision, such as in preliminary threat detection scenarios where quick alerts are necessary.

LinearSVC offered an improvement over Naive Bayes, with an accuracy of 66%. The model achieved relatively balanced precision and recall for detecting malicious traffic, reflected in an F1-score of about 65%. Despite this, LinearSVC's reliance on linear decision boundaries likely hindered its ability to accurately classify non-linear patterns within the dataset, limiting its overall effectiveness. While LinearSVC provides a reasonable balance between speed and accuracy, it may fall short in more complex SDN attack scenarios where the data patterns are intricate and not easily separable by linear classifiers. Nevertheless, LinearSVC's moderate performance makes it a feasible option in environments where both computational efficiency and detection speed are crucial, offering a compromise between the two.

Random Forest emerged as the most effective model, achieving an impressive 97% accuracy and consistently high F1-scores across both benign and malicious classes. This model's superior performance can be attributed to its ensemble approach, which aggregates decisions from multiple trees. This technique reduces the likelihood of overfitting and enhances the model's ability to capture complex, non-linear relationships within the data. Random Forest's robustness makes it especially well-suited for identifying DDoS attacks in environments with diverse traffic patterns, where accurately distinguishing between malicious and benign activities is critical. However, the trade-off for this accuracy is that Random Forest is computationally intensive, requiring more processing power and time. This makes it less ideal for real-time DDoS mitigation but highly effective for detailed, offline analysis and network forensics, where comprehensive assessment is more important than speed.

The ROC curve analysis, illustrated in Figure 3, further demonstrates the Random Forest model's exceptional performance, with an Area Under the Curve (AUC) of 1.00, indicating its excellent classification ability. This high level of accuracy is essential in practical SDN deployments, where traffic volumes can surge unpredictably during an attack. However, in selecting a detection model, it is important to balance accuracy with speed and resource consumption. Although

Random Forest provides the highest accuracy, its computational demands suggest that it may be best suited for scenarios where detailed, post-attack analysis is required or where ample computational resources are available.

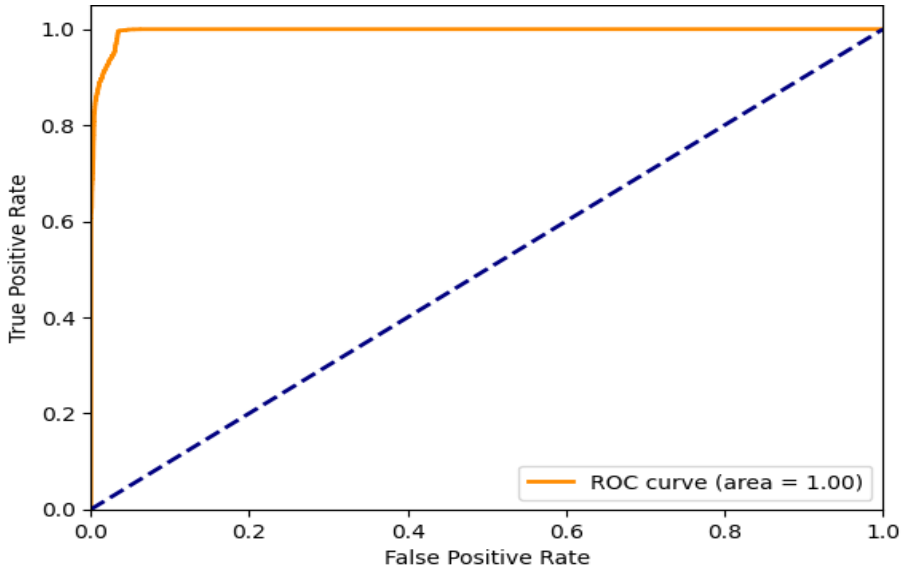

Figure 3. (ROC) Curve

In contrast, Naive Bayes may be more suitable for preliminary threat detection, where its ability to generate rapid alerts can serve as a first line of defense. These alerts can then be followed up by more computationally intensive models like Random Forest, which can provide a deeper analysis. This layered approach to security ensures that potential threats are flagged quickly, with Random Forest delivering the thorough analysis needed for confirmation and response.

LinearSVC offers a balanced middle ground, combining moderate accuracy with relatively low computational demands. This makes it a versatile option for environments where both speed and resource constraints are significant factors. However, its linear nature might limit its effectiveness in more complex attack scenarios, suggesting that its use is best suited to settings where the complexity of the traffic is moderate and the need for real-time detection is high.

These findings are consistent with previous research, which has frequently highlighted the superiority of Random Forest in detecting DDoS attacks, particularly in network environments characterized by diverse traffic patterns. For example, studies utilizing the CICIDS2017 dataset [15] have reported similarly high accuracy levels for Random Forest, reinforcing its effectiveness in anomaly detection tasks. However, other studies have noted that when Naive Bayes is used

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

in conjunction with feature selection techniques, it can achieve comparable results at a much lower computational cost. This suggests that optimizing feature selection could enhance Naive Bayes's performance in SDN environments, potentially making it a more viable option in scenarios where computational efficiency is paramount.

The Random Forest clearly stands out as the most reliable model for detecting DDoS attacks in SDN environments, the choice of model should be informed by the specific needs of the deployment scenario. Factors such as the requirement for real-time detection, available computational resources, and the complexity of the traffic patterns should guide the selection process. By carefully considering these factors, network administrators can choose a model that not only delivers high accuracy but also aligns with the operational demands of their network security strategy.

4. CONCLUSION

This research demonstrated that the Random Forest model significantly outperformed the other models, achieving an impressive accuracy of 97%, compared to the 63% and 66% accuracies achieved by Naive Bayes and LinearSVC, respectively. The superior performance of the Random Forest model is attributed to its ability to handle large datasets and deliver high classification accuracy. These findings underscore the potential of machine learning methods to enhance SDN security, particularly in the accurate detection of DDoS attacks. Future research should focus on exploring deep learning models to further enhance detection capabilities, as these models may offer even greater accuracy and adaptability. It is also essential to use more diverse and comprehensive datasets to validate the models' effectiveness across various network environments and attack scenarios, ensuring that the models are robust and generalizable. Additionally, developing efficient real-time DDoS detection systems is crucial to achieving minimal latency and high accuracy in live network settings. Improving feature engineering and selection techniques could further optimize model performance, potentially reducing computational overhead while maintaining high accuracy. Finally, future studies should investigate methods to enhance the models' resilience against adversarial attacks, ensuring that they remain reliable even when faced with attempts to manipulate input data. Strengthening these aspects will contribute to the development of more robust and dependable security solutions for SDN environments.

REFERENCES

[1] K. Govindarajan, K. C. Meng, and H. Ong, "A literature review on software-defined networking (SDN) research topics, challenges and solutions," in *2013 fifth International conference on advanced computing (ICoAC)*, 2013, pp. 293–299.

[2] S. Siddiqui *et al.*, "Toward software-defined networking-based IoT frameworks: A systematic literature review, taxonomy, open challenges and prospects," *IEEE Access*, vol. 10, pp. 70850–70901, 2022.

[3] A. Aljuhani, "Machine learning approaches for combating distributed denial of service attacks in modern networking environments," *IEEE Access*, vol. 9, pp. 42236–42264, 2021.

[4] T. Radivilova, L. Kirichenko, D. Ageiev, and V. Bulakh, "Classification methods of machine learning to detect DDoS attacks," in *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, 2019, pp. 207–210.

[5] U. Ince and G. Karaduman, "Classification of Distributed Denial of Service Attacks Using Machine Learning Methods," *NATURENGS*, vol. 5, no. 1, pp. 15–20, 2024.

[6] P. S. Saini, S. Behal, and S. Bhatia, "Detection of DDoS attacks using machine learning algorithms," in *2020 7th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2020, pp. 16–21.

[7] Y. Al-Hadhrami and F. K. Hussain, "DDoS attacks in IoT networks: a comprehensive systematic literature review," *World Wide Web*, vol. 24, no. 3, pp. 971–1001, 2021.

[8] A. A. Bahashwan, M. Anbar, S. Manickam, T. A. Al-Amiedy, M. A. Aladaileh, and I. H. Hasbullah, "A systematic literature review on machine learning and deep learning approaches for detecting DDoS attacks in software-defined networking," *Sensors*, vol. 23, no. 9, p. 4441, 2023.

[9] S. Kumar, N. P. Singh, and N. Kumar, "Literature Review of Distributed Denial of Service (DDoS) Attacks, its Detection Techniques and Prevention Mechanisms," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, pp. 1681–1685, 2022.

[10] G. Somani, M. S. Gaur, D. Sanghi, M. Conti, and R. Buyya, "DDoS attacks in cloud computing: Issues, taxonomy, and future directions," *Comput. Commun.*, vol. 107, pp. 30–48, 2017.

[11] S. K. Fayaz, Y. Tobioka, V. Sekar, and M. Bailey, "Bohatei: Flexible and elastic DDoS defense," in *24th USENIX security symposium (USENIX Security 15)*, 2015, pp. 817–832.

[12] T. Aytaç, M. AYDIN, and A. ZA.IM, "Detection DDOS attacks using machine learning methods," *Electrica*, vol. 20, no. 2, 2020.

[13] M. W. Nadeem, H. G. Goh, V. Ponnusamy, and Y. Aun, "DDoS Detection in SDN using Machine Learning Techniques.," *Comput. Mater. & Contin.*, vol. 71, no. 1, 2022.
[14] J. Brownlee, "Bagging and random forest ensemble algorithms for machine learning," *Mach. Learn. Algorithms*, pp. 4–22, 2016.
[15] A. A. Abdulrahman and M. K. Ibrahem, "Evaluation of DDoS attacks detection in a CICIDS2017 dataset based on classification algorithms," *Iraqi J. Inf. Commun. Technol.*, vol. 1, no. 3, 2018.