Performance Analysis of Gradient Boosting and Decision Tree on Distributed Denial of Service Attacks in Software-Defined Networks

Lilis Nurhayati [1]*, A.M Iqra Rezky Hatta [2], Herdianti[3]
Departemen of Informatic Engineering, Faculty Of Computer Science[1], [2], [3]
Unvieristy Muslim Indonesia
Makassar, Indonesia
13020210005@umi.ac.id[1], lilis.nurhayati@umi.ac.id[2], herdianti.darwis@umi.ac.id [3]

Abstract— **Distributed Denial of Service (DDoS) attacks remain a prominent threat to modern network infrastructures, particularly in Software Defined Networks (SDNs), which operate under a centralized control architecture. This study aims to assess the effectiveness of Gradient Boosting and Decision Tree algorithms for identifying DDoS attacks in SDN environments. To improve model performance, we applied preprocessing and feature selection to a publicly available SDN-based DDoS dataset. The feature selection process successfully reduced the number of attributes from 23 to the 10 most influential features for classification. The models were trained and evaluated using multiple data splitting ratios: 60:40, 70:30, 80:20, and 90:10. Their performance was measured through accuracy, precision, recall, F1-score, and confusion matrix analysis. Experimental results showed that Gradient Boosting achieved the highest accuracy of 95.53% on a 90:10 split, with relatively low computation time. In comparison, the Decision Tree achieved a maximum accuracy of 94.26% but required more processing time. The confusion matrix for the best-performing model showed high true-positive and true-negative rates, with a low false-negative rate, indicating reliable detection capabilities. This study contributes to the ongoing research in DDoS detection by highlighting the effectiveness of machine learning algorithms in SDN environments.**

Keywords— DDoS, SDN, Machine Learning, Gradient Boosting, Decision Tree

I. INTRODUCTION

The rapid evolution of technology has increased our dependence on computer systems across various sectors such as finance, healthcare, and manufacturing. These systems are interconnected through diverse computer networks to enable seamless communication and data exchange. However, this interconnectivity has also introduced new vulnerabilities, as evidenced by the growing number of major global cyberattacks in recent years. Consequently, enhancing computer networks security has become essential, particularly through the implementation of network intrusion detection systems, to detect and mitigate potential threats proactively [1]. However, the rapid advancement of network infrastructure has significantly increased the risk of cyberattacks. Among the most severe threats are Distributed Denial-of-Service (DDoS) attacks, which aim to flood target systems with excessive illegitimate traffic. Such attacks can lead to substantial degradation in system performance, service disruptions, or even the complete shutdown of critical operations. These attacks often exploit system vulnerabilities and require minimal resources from the attacker, making them relatively easy to launch but difficult to mitigate. As digital services become more integral to daily operations across industries, the potential impact of a successful DDoS attack becomes increasingly severe. Therefore, developing effective detection and mitigation strategies has become a key priority in ensuring the resilience and security of modern network infrastructures [2][3][4].

Software-Defined Networking (SDN) is a modern approach to network management that separates control functions from data forwarding, enabling more centralized, flexible, and efficient network management. With this architecture, SDN can dynamically respond to changes in network requirements. However, centralizing control in SDN also creates significant security vulnerabilities, particularly against DDoS attacks. Such attacks can disrupt the primary control pathways, potentially crippling the entire network systems and hindering critical services. [5][6][7]. In implementing an SDN architecture, the main challenge is detecting DDoS attacks efficiently, balancing accuracy and speed. The huge volume of data and the dynamics of network traffic patterns create high complexity, so that conventional detection mechanisms such as firewalls and intrusion detection systems (IDSs) are often unable to identify threats with sufficient accuracy [8][9][10]. Fig. 1 illustrates the fundamental architecture of SDN, consisting of three main layers: the application layer, the control layer, and the data layer. The separation of the control and data planes in SDN introduces new opportunities for centralized traffic monitoring and dynamic policy enforcement. This architectural feature enables more flexible, programmable network management, which can be leveraged to deliver more effective security solutions. Despite this, the lack of built-in security features in SDN makes it vulnerable to various types of attacks, including DDoS.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i4.2482, Copyright ©2025
Submitted: July 22, 2025, Revised: August 25, 2025, Accepted: September 9, 2025, Published: October 15, 2025

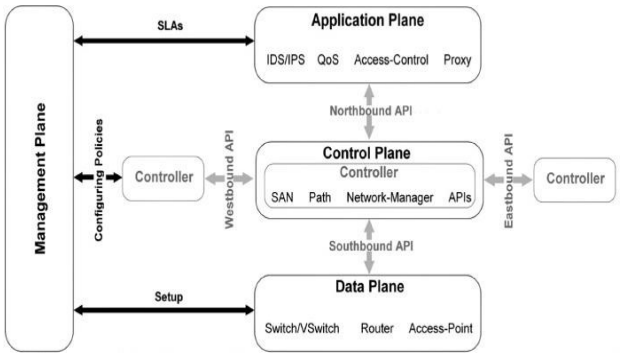

Fig. 1. SDN Architecture

Therefore, machine learning-based approaches are increasingly adopted in recent studies, given their ability to adapt to changing network environments and automatically and independently recognize attack patterns. The use of machine learning-based predictive models has become a practical approach for classifying incidents, including in network security. One of its most relevant applications is in identifying and classifying types of DDoS attacks [11][12]. This classification process not only helps recognize the characteristics of each kind of attack but also provides insight into the techniques used by cybercriminals. With a deeper understanding of these threat patterns, companies can evaluate the potential risks that could cause reputational and financial losses. This enables strategic decisions to allocate resources to strengthen network security systems optimally and sustainably.

This study integrates an Intrusion Detection System (IDS) with a DDoS attack detection mechanism specifically for SDN environments. This system consists of two main components. The first component processes incoming requests and identifies host behavior, whether it shows normal or deviant patterns. If abnormal behavior is detected, the host is forwarded to the second component for further analysis of the data packets it sends, to improve the accuracy of anomaly detection. The implementation of these two components effectively reduces processing time, as demonstrated by previous research [13].

In related studies, the integration of sFlow and OpenFlow was used to detect DDoS attacks in SDN networks and implement effective mitigation mechanisms. sFlow functions as a network traffic monitoring system based on sampling, providing comprehensive visibility into network activity by collecting data from devices such as switches and routers. On the other hand, OpenFlow serves as a communication protocol between controllers and network devices, enabling flexible, centralized configuration of data flows. The combination of these two technologies allows early detection of traffic anomalies and efficient adaptive responses to potential cyberattacks [14].

This study aims to develop and analyze the performance of Gradient Boosting and Decision Trees for classifying DDoS attacks in SDN networks. The study examines the extent to which the two models can identify various attack types with higher accuracy than several other machine learning algorithms, using a DDoS attack dataset designed explicitly for the SDN environment. The results obtained are expected to contribute to the development of more effective approaches in detecting DDoS attacks and enhancing the resilience of SDN networks against increasingly complex cyber threats. The primary focus of this research is to conduct a comparative analysis of the effectiveness and accuracy of attack classification to assess the advantages of each method in the context of SDN-based network security.

II. METHODOLOGY

Every scientific study requires a systematic methodology to achieve its research objectives. The conceptual framework of this study was formulated through a series of structured stages, each described in the sub-sections of the research process. These stages are arranged in sequence and visualized as a flowchart in Fig. 2 to provide a clearer picture of the overall research process.

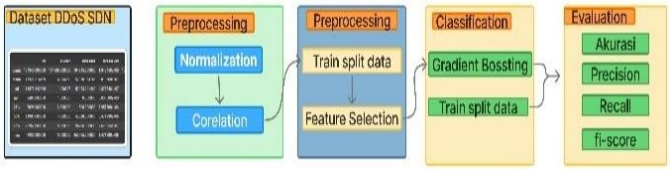

Fig. 2. Research Method

III. DATASET

This study uses a publicly available dataset of DDoS attacks on SDN networks via the Kaggle platform, titled “DDoS SDN Dataset.” This dataset was selected because it is highly relevant to the literature and provides comprehensive information on the characteristics of DDoS attacks in SDN networks. The information provided includes various attack types and the resulting network traffic patterns. The primary purpose of using this dataset is to evaluate its effectiveness in identifying and classifying DDoS attacks by applying various classification methods. In addition, this dataset is used as the primary data source for the machine learning model's training and testing in this study. Specifically, the dataset consists of 104,345 rows and 23 columns, including a target variable, the label, which identifies the type of network traffic: 0 denotes normal (benign) traffic and 1 denotes malicious traffic. The distribution of data counts across label categories is presented in Table I.

TABLE I. DATASET INFORMATION

Column	Definition	Data type
dt	Date timestamp	int64
switch	Network switch ID	int64
src	Source address	object
dst	Destination address	object
pktcount	Packet count	int64
bytecount	Byte count	int64
dur	Flow duration	int64
dur_nsec	Duration in nanoseconds	float64
tot_dur	Total flow duration	int64

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i4.2482, Copyright ©2025
Submitted: July 22, 2025, Revised: August 25, 2025, Accepted: September 9, 2025, Published: October 15, 2025

flows	Total flows	int64
packetins	Packet-in events	int64
pktperflow	Packets per flow	int64
byteperflow	Bytes per flow	int64
pktrate	Packet transmission rate	int64
Pairflow	Paired flows	int64
Protocol	Network protocol used	object
port_no	Port number	int64
tx_bytes	Transmitted bytes	int64
rx_bytes	Received bytes	int64
tx_kbps	Transmitted rate (kbps)	int64
rx_kbps	Received rate (kbps)	float64
tot_kbps	Total data rate (kbps)	float64
label	Flow label	int64

To evaluate the optimal performance of the developed model, several data-splitting scenarios were applied with ratios of 60:40, 70:30, 80:20, and 90:10 for training and testing data. Each of these scenarios was analyzed to assess the impact of the training data proportion on the ability of the Gradient Boosting and Decision Tree models to identify and classify DDoS attacks in the SDN dataset accurately. The evaluation used all data-splitting scenarios as a reference to measure model performance, both in terms of classification accuracy and efficiency in completing the attack pattern categorization task.

IV. PREPROCESSING

In the data preprocessing stage, selecting relevant features is a crucial step in improving the efficiency and effectiveness of the classification model. The feature selection process aims to remove data dimensions that do not contribute significantly to classification results, while identifying the variables most influential in detecting DDoS attacks [15][16]. This approach not only improves model accuracy but also reduces the risk of overfitting, a condition in which the model is overly complex and tends to capture noise rather than true patterns. By using only features that have a significant impact on classification, the model becomes simpler and more reliable, producing consistent predictions. Additionally, feature selection accelerates model training time and improves the readability and interpretability of analysis results. This enables researchers to better understand the role of each feature at every stage of attack detection and highlight the key elements that need to be considered in network security systems. Evaluations of feature relevance and redundancy are conducted using a range of approaches, including statistical methods, machine learning algorithms, and heuristic techniques. Therefore, in designing an optimal DDoS attack detection model in an SDN environment, feature selection methods are a fundamental aspect that cannot be overlooked [17][18].

This process begins with data normalization, followed by correlation analysis to evaluate feature relationships, then data partitioning, feature selection to identify relevant attributes, and finally the application of the Gradient Boosting and Decision Tree models. The primary objective of correlation analysis is to identify relationships between features and targets and ensure that the features used are truly significant for DDoS attack patterns in SDN networks [19]. Next, the dataset was split into training and test sets to facilitate objective evaluation of model performance. The feature selection process was then applied to filter out attributes that did not significantly contribute to classification results, thereby improving the model's efficiency. After the essential features are identified, the model is trained using two classification algorithms: Gradient Boosting and Decision Tree, chosen for their ability to handle nonlinear data and provide interpretable results. The visualization of the correlation results with the features considered relevant in detecting DDoS attacks on SDN networks is presented in Fig. 3 as the basis for the feature selection stage.

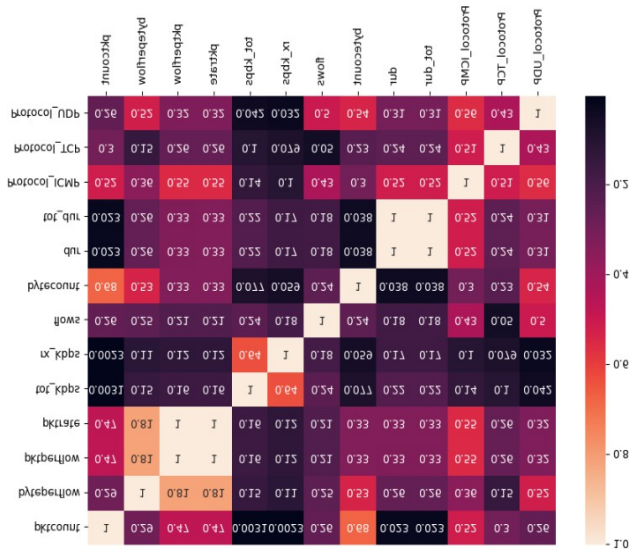

Fig. 3. Heatmap Correlation

Fig. 3 presents a correlation heatmap that illustrates the relationships among the features in the dataset. Light colors indicate high correlation values, indicating a strong relationship between two features, while dark colors indicate low correlation values, indicating a weak relationship. This correlation analysis assists the feature selection process by identifying attributes with a significant relationship with the target feature. This information serves as the basis for determining which features are relevant to retain and which should be eliminated because they are redundant or do not significantly improve classification performance. Table II contains the correlation results considered appropriate for this process. By prioritizing features with high correlations to the target, the feature selection process becomes more focused. Removing duplicative attributes can improve model efficiency, reduce computational complexity, and enhance prediction accuracy. This strategy ultimately yields a more optimal and effective model for classifying DDoS attacks in SDN networks.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i4.2482, Copyright ©2025
Submitted: July 22, 2025, Revised: August 25, 2025, Accepted: September 9, 2025, Published: October 15, 2025

TABLE II. FEATURE SELECTION

Column	Definition	Data type
pktcount	Packet count	int64
byteperflow	Bytes per flow	int64
tot_kbps	Total data rate (kbps)	float64
rx_kbps	Received rate (kbps)	float64
flows	Total flows	int64
bytecount	Byte count	int64
tot_dur	Total flow duration	int64
Protocol_ICMP	Protocol ICMP	object
Protocol_TCP	Protocol TCP	object
Protocol_UDP	Protocol UDP	object

After the feature selection based on relevance level, the dataset was reduced from 23 to 10 columns, yielding 103,839 data entries. This dataset is divided into two categories: class 0, comprising 63,335 data points representing regular traffic, and class 1, comprising 40,504 data points indicating attack activity. The ten selected features serve as key indicators reflecting the characteristics of DDoS attacks on SDN networks. These features include increases in the number of packets, data flow rate, and the volume of data transmitted. Additionally, to support a comprehensive understanding of attack patterns, essential parameters such as the number of data flows, flow duration, and the protocol type used in network communication are included.

V. CLASSIFICATION

At this stage, classification is performed, which is the grouping of data into a set of classes or categories based on the dataset's characteristics [20]. This classification process is a crucial component of data analysis, as it aims to recognize and reveal hidden patterns in varied data [21][22][23]. In this study, two classification approaches were used: Gradient Boosting and Decision Tree. Figure 4 shows the architecture of the Gradient Boosting model.

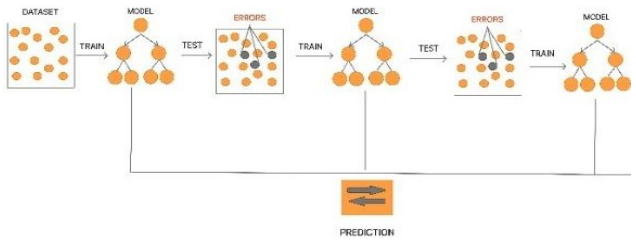

Fig. 4. Model Gradient Boosting

Gradient Boosting is an ensemble learning method used for predictive modeling in both regression and classification. This technique builds models gradually, with each new model correcting the errors of the previous one [24][25]. Each iteration adds a new tree trained to minimize the residual error of the previous model. This results in an accurate model with high predictive performance.

A decision tree is a machine learning algorithm used to solve classification and regression problems. This algorithm works by recursively dividing a dataset based on specific features to form a tree structure that represents the decision-making process [26][27]. However, Decision Trees tend to overfit to training data, especially if the tree structure is not simplified. Fig 5. shows the architecture of a Decision Tree model.

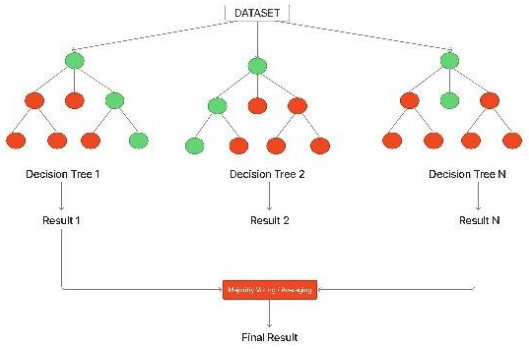

Fig. 5. Model *Decision Tree*

Each node in the tree represents a test of a specific feature; branches represent the results of those tests, while leaf nodes show the final results as classes or predicted values. One advantage of decision trees is their ability to produce models that are easy for humans to understand and do not require specific assumptions about data distribution. This makes the algorithm very flexible for various types of datasets.

VI. PERFORMANCE EVALUATION AND ANALYSIS

During the testing phase, performance in classifying DDoS attacks was evaluated using models trained with Gradient Boosting and Decision Tree architectures. We evaluated the model using validation data not used during training to ensure an objective assessment of its generalization capabilities.

To assess the model's ability to recognize and classify DDoS attack types, various data partitioning schemes were applied and compared. The purpose of this evaluation is to identify data partitioning strategies that yield high performance. The parameters used are accuracy, precision, recall, and F1-Score. Validation accuracy is calculated by comparing the number of correct predictions to the total number of predictions generated by the model. The accuracy formula is presented in equation (1).

$$\text{Accuracy} = \frac{\square + \square\square}{\square + \square + \square + \square\square} \quad (1)$$

Equation (1) is used to calculate accuracy, the proportion of correct predictions among the model's total predictions. In this case, True Positive (TP) and True Negative (TN) indicate the number of cases in which the model successfully identified the data correctly as positive or negative. Conversely, False Positive (FP) and False Negative (FN) represent prediction errors, i.e., when the model incorrectly classifies data into the wrong class, either labeling negative data as positive or vice versa [28].

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i4.2482, Copyright ©2025
Submitted: July 22, 2025, Revised: August 25, 2025, Accepted: September 9, 2025, Published: October 15, 2025

$$\text{Presisi} = \frac{\square\square}{\square + \square\square} \quad (2)$$

Precision is a quantitative measure that expresses the proportion of true positives to the total number of optimistic predictions (true positives + false positives). Conceptually, precision describes a model's accuracy in classifying entities as positive. Precision values range from [0, 1], where values close to 1 indicate that the majority of the model's optimistic predictions are valid or consistent with actual conditions.

$$\text{Recall} = \frac{\square\square}{\square + \square\square} \quad (3)$$

Recall measures the proportion of actual positive cases correctly identified by the model out of the total number of actual positive cases. The recall value ranges from 0 to 1. The higher the recall value, the greater the model's ability to identify all instances of the actual positive class. Thus, recall is crucial in applications that require comprehensive detection of positive cases, such as disease diagnosis, fraud detection, or security systems.

$$\square_1 = \frac{2\square\square}{2\square + \square + \square\square} \quad (4)$$

The F1-Score provides a single value that accounts for both the accuracy of optimistic predictions (precision) and the coverage of detected positive classes (recall). The F1-score ranges from 0 to 1, where a value close to 1 indicates that the model performs well on both precision and recall. Therefore, this metric is highly relevant in situations where minimizing classification errors in both directions is essential.

During the testing phase, the confusion matrix is used as an evaluation tool to analyze the performance of classification models that achieve the highest validation accuracy in each data-sharing scenario. The confusion matrix provides a detailed numerical representation of classification results, enabling the identification of misclassifications and the data patterns that are misclassified.

Using a confusion matrix, this study aims to identify the optimal model architecture for detecting and classifying DDoS attacks. In addition, this analysis also seeks to evaluate the extent to which variations in the proportion of training and validation data affect model performance. Thus, the results of this evaluation are expected to provide a comprehensive understanding of the model's sensitivity to changes in data distribution during training and validation, and its implications for generalization to previously unseen data.

VII. Results and Discussions

In the evaluation phase, we trained two classification algorithms — Gradient Boosting and Decision Tree — on identical datasets to classify different types of DDoS attacks. To reduce the possibility of overfitting or bias, the training and validation data were divided into scenarios with various proportions, namely 60:40, 70:30, 80:20, and 90:10. This division aims to assess the generalization capacity of each model on data that has not been used during training, resulting in a more objective and representative performance evaluation. In addition, applying various data split ratios is intended to assess the model's resilience to fluctuations in the amount of available training data. This ensures the model's performance is not dependent on a particular data distribution. Through a comparative analysis of performance metrics across scenarios, this study seeks to identify the model that is most reliable at producing accurate predictions, even under unstable, variable real-world conditions.

A. Gradient Boosting Model Accuracy

The first experiment was conducted on training data using ratios of 60:40, 70:30, 80:20, and 90:10, and compared the Gradient Boosting model, as shown in Table III.

TABLE I. GRADIENT BOOSTING ACCURACY RESULT

No.	Data Separation	High Validation Accuracy	Computing Time
1	60:40	95.45%	11,85 second
2	70:30	95.44%	13,52 second
3	80:20	95.46%	15,65 second
4	90:10	**95.53%**	**17,18 second**

The Gradient Boosting model showed consistent classification performance, with validation accuracies ranging from 95.45% for a 60:40 data split to 95.53% for a 90:10 data split, indicating good generalization. Increasing the proportion of training data led to a slight increase in accuracy but also increased computation time from 11.85 seconds to 17.18 seconds. Overall, the model is effective at achieving a balance between accuracy and time efficiency, making it suitable for network traffic classification.

B. Accuracy results of the Decision Tree Model

The accuracy results from the Decision Tree model are shown in Table IV.

TABLE IV. DECISION TREE ACCURACY RESULT

No.	Data Separation	High Validation Accuracy	Computing Time
1	60:40	**94.26%**	**1 minute 29,52 seconds**
2	70:30	94.19%	1 minute 30,18 seconds
3	80:20	94.15%	1 minute 51,36 seconds
4	90:10	94.04%	2 minutes 2,99 seconds

Table IV shows that the Decision Tree model exhibits stable classification performance, with validation accuracies ranging from 94.04% at a 90:10 data split to 90.26% at a 60:40 data split. Although there is a tendency for accuracy to decrease as the proportion of training data increases, the difference is relatively small (<0.3%), suggesting a still pretty good generalization ability. In terms of efficiency, computation time increased as the training data increased, from 1 minute 29.52 seconds at a 60:40 split to 2 minutes 2.99 seconds at a 90:10 split. This finding indicates that using the Decision Tree model

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i4.2482, Copyright ©2025
Submitted: July 22, 2025, Revised: August 25, 2025, Accepted: September 9, 2025, Published: October 15, 2025

at a larger scale requires consideration of the higher computational time due to the increased complexity of the training process.

C. Overall Model Accuracy Results

A comparison of the validation accuracy results and the computation time of all models is shown in Table V.

TABLE V. ACCURACY VALIDATION RESULT ACCORDING TO ALL DATA SHARING

Architecture	60:40	70:30	80:20	90:10	Computing Time
Gradient Boosting	95.45%	95.45%	95.46%	**95.53%**	**17,18 second**
Decision Tree	**94.26%**	94.19%	94.15%	94.04%	**1 minute 29,52 second**

Based on the table results, the Gradient Boosting algorithm shows more consistent and superior classification performance than the Decision Tree model across all training and test splits. The validation accuracy of Gradient Boosting ranges from 95.45% to 95.53%, indicating the model's stability and strong generalization across different test datasets. In contrast, the Decision Tree shows slightly lower accuracy, ranging from 94.04% to 94.26%, with a decreasing trend as the proportion of training data increases. In terms of time efficiency, Gradient Boosting showed superiority, with a training time of about 17.18 seconds, much shorter than Decision Tree's 1 minute 29.52 seconds. This indicates that although Gradient Boosting theoretically has a more complex structure, in terms of computation, it is more efficient.

Considering accuracy and time efficiency, it can be concluded that Gradient Boosting is a more optimal approach for network traffic classification in this scenario. The confusion matrix for the best-performing model, namely Gradient Boosting with a 90:10 split, is shown in Fig. 5. This model achieves a high actual positive rate while maintaining a low false positive rate. The balanced trade-off between precision and recall indicates its robustness in handling diverse traffic patterns. Moreover, Gradient Boosting's ability to handle high-dimensional data and its resilience to overfitting contribute to its effectiveness in complex network environments. Compared to other models tested in this study, such as Random Forest and Support Vector Machine, Gradient Boosting consistently produced more stable results across different validation splits.

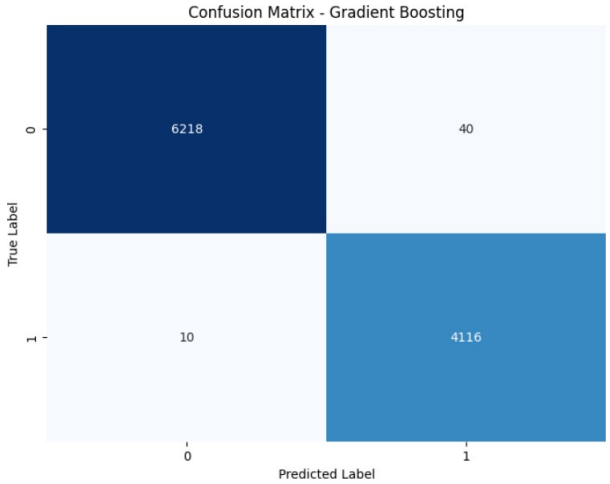

Fig. 5. Confusion Matrix Gradient Boosting 90:10

The confusion matrix above shows the performance of the Gradient Boosting model with a 90:10 training-to-testing split in detecting regular benign network traffic and malicious DDoS attacks. Based on these results, the model correctly classified 6,218 benign data points as true negatives and 4,116 malicious data points as true positives. Meanwhile, 40 benign data points were incorrectly classified as malicious, leading to false positives, while only 10 malicious data points were incorrectly classified as benign, resulting in false negatives.

The high true-positive and true-negative values indicate that the model has strong generalization ability and high sensitivity and specificity. The low false-negative rate is crucial in cybersecurity, as it ensures that most attacks are detected on time. Thus, the Gradient Boosting model has been proven effective and efficient for use in an intrusion detection system to identify DDoS threats accurately.

D. Accuracy results of previous research comparisons

A comparison of the performance model's accuracy in this study with that in previous studies is presented in Table VI.

TABLE VI. ACCURACY COMPARISON OF SDN DDOS DATASET

Research	Algorithm	Accuracy
Dahlan, I A [2]	KNN	98.00%
Jia [8]	SVM	99.86
Model yang diusulkan	Gradient Boosting	95.53%
	Decision Tree	94.26%

Table VI compares the accuracy levels of the model proposed in this study with several machine learning algorithms used in previous studies. In the study by Dahlan, I. A. [2], the KNN algorithm achieved an accuracy of 98.00%. Meanwhile, research conducted by Jia [8] showed that the Support Vector Machine (SVM) algorithm achieved the highest accuracy of 99.86%.

In this study, two algorithmic approaches were applied: Gradient Boosting and Decision Trees. The test results showed that the Gradient Boosting model achieved the highest validation accuracy of 95.53%, while the Decision Tree

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i4.2482, Copyright ©2025
Submitted: July 22, 2025, Revised: August 25, 2025, Accepted: September 9, 2025, Published: October 15, 2025

achieved 94.26%. Although the accuracy of the two models has not surpassed the performance of models from previous research, especially SVM, they still show quite good and consistent classification performance.

VIII. CONCLUSION

This study aims to evaluate the performance of two classification algorithms — Gradient Boosting and Decision Tree — in detecting DDoS attacks in SDN environments. We developed and tested the model using a machine learning approach and a systematic feature selection process on SDN-based DDoS datasets with various training and testing scenarios.

The evaluation results show that the Gradient Boosting model consistently achieves higher validation accuracy than the Decision Tree, reaching 95.53% at a 90:10 data split. Meanwhile, the Decision Tree achieved the highest accuracy of 94.26%. In addition to excelling in accuracy, Gradient Boosting also shows better computational efficiency than Decision Trees, despite having higher structural complexity. The confusion matrix analysis of the best model, namely Gradient Boosting at a 90:10 split, shows high true-positive and negative rates, as well as a very low false-negative rate. This indicates the model's ability to accurately classify benign and malicious network traffic, which is crucial for intrusion detection systems.

Compared with previous studies using algorithms such as KNN and SVM, the proposed model's accuracy is still lower. However, this approach still delivers competitive performance and can serve as a viable alternative in SDN-based network security systems. Thus, the Gradient Boosting model can be considered as an effective and efficient solution for automatically detecting DDoS attacks in modern network environments.

For future research, it is recommended to explore the use of Deep Learning algorithms, such as Recurrent Neural Networks (RNNs), which are better suited to timing and sequential patterns in network traffic. In addition, testing against real-time datasets or more complex SDN environments may provide results that are more representative of real-world implementations.

REFERENCES

[1] M. Agus, O. Riduan, and H. Alamsyah, "Analisa Dan Implementasi Kemanan Jaringan Berbasis Firewall Raw Terhadap Serangan DDoS Pada Router Mikrotik," vol. 21, no. 1, pp. 317–328, 2025, doi: https://doi.org/10.37676/jmi.v21i1.7835.

[2] R. Satra, I. A. Dahlan, H. Darwis, Purnawansyah, S. Mujaddid, and F. Fattah, "A Comparison of Accuracy: KNN, TabNet, and Wide & Deep Learning for DDoS Attack Detection in Software Defined Network," in *2025 19th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 2025, pp. 1–8. doi: 10.1109/IMCOM64595.2025.10857511.

[3] Purnawansyah, N. A. Supriadi, A. R. Manga, R. Adawiyah, Harlinda, and T. Hasanuddin, "Application of Ensemble Machine Learning for DDoS Detection in Complex Network Environments," in *2025 19th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 2025, pp. 1–7. doi: 10.1109/IMCOM64595.2025.10857516.

[4] F. Khashab, J. Moubarak, A. Feghali, and C. Bassil, "DDoS Attack Detection and Mitigation in SDN using Machine Learning," in *2021 IEEE 7th International Conference on Network Softwarization (NetSoft)*, 2021, pp. 395–401. doi: 10.1109/NetSoft51509.2021.9492558.

[5] J. A. Perez-Diaz, I. A. Valdovinos, K. K. R. Choo, and D. Zhu, "A Flexible SDN-Based Architecture for Identifying and Mitigating Low-Rate DDoS Attacks Using Machine Learning," *IEEE Access*, vol. 8, pp. 155859–155872, 2020, doi: 10.1109/ACCESS.2020.3019330.

[6] A. Cohen *et al.*, "Bringing Network Coding into SDN: Architectural Study for Meshed Heterogeneous Communications," *IEEE Commun. Mag.*, vol. 59, no. 4, pp. 37–43, 2021, doi: 10.1109/MCOM.001.2000875.

[7] N. M. Yungaicela-Naula, C. Vargas-Rosales, and J. A. Perez-Diaz, "SDN-based architecture for transport and application layer DDoS attack detection by using machine and deep learning," *IEEE Access*, vol. 9, pp. 108495–108512, 2021, doi: 10.1109/ACCESS.2021.3101650.

[8] M. Jia, S. Yuejie, G. Qing, G. Zihe, and X. Suofei, "DDoS Attack Detection Method for DDoS Attack Detection Method for Space Space-Based Network Based Based Network Based on SDN Architecture on SDN Architecture," *ZTE Commun.*, vol. 18, no. 4, pp. 18–25, 2020, doi: 10.12142/ZTECOM.202004004.

[9] M. H. H. Khairi *et al.*, "Detection and Classification of Conflict Flows in SDN Using Machine Learning Algorithms," *IEEE Access*, vol. 9, pp. 76024–76037, 2021, doi: 10.1109/ACCESS.2021.3081629.

[10] M. Klymash, O. Shpur, N. Peleh, and O. Maksysko, "Concept of Intelligent Detection of DDoS Attacks in SDN Networks Using Machine Learning," in *2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T)*, 2020, pp. 609–612. doi: 10.1109/PICST51311.2020.9467963.

[11] U. M. Malang and D. Tree, "Deteksi Dan Mitigasi Serangan DDoS Pada Software Defined Network Menggunakan Algoritma Decision Tree," vol. 2, no. 11, pp. 1491–1502, 2020.

[12] M. R. S. Rao, D. Yadav, and V. Anbarasu, "An Improvised Machine Learning Model KNN for Malware Detection and Classification," in *2023 International Conference on Computer Communication and Informatics (ICCCI)*, 2023, pp. 1–4. doi: 10.1109/ICCCI56745.2023.10128189.

[13] L. Wang and Y. Liu, "A DDoS Attack Detection Method Based on Information Entropy and Deep Learning in SDN," in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 2020, pp. 1084–1088. doi: 10.1109/ITNEC48623.2020.9085007.

[14] L. Barki, A. Shidling, N. Meti, D. G. Narayan, and M. M. Mulla, "Detection of distributed denial of service attacks in software defined networks," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2016, pp. 2576–2581. doi: 10.1109/ICACCI.2016.7732445.

[15] R. F. Fouladi, O. Ermiş, and E. Anarim, "A DDoS attack detection and defense scheme using time-series analysis for SDN," *J. Inf. Secur. Appl.*, vol. 54, p. 102587, 2020, doi: https://doi.org/10.1016/j.jisa.2020.102587.

[16] H. Polat and O. Polat, "Detecting DDoS Attacks in Software-Defined.pdf," *Mdpi*, 2020, doi: doi:10.3390/su12031035.

[17] S. Gupta and D. Grover, "A Comprehensive Review on Detection of DDoS Attacks using ML in SDN Environment," in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, 2021, pp. 1158–1163. doi: 10.1109/ICAIS50930.2021.9395987.

[18] K. S. Sahoo *et al.*, "An Evolutionary SVM Model for DDOS Attack Detection in Software Defined Networks," *IEEE Access*, vol. 8, pp. 132502–132513, 2020, doi: 10.1109/ACCESS.2020.3009733.

[19] A. P. Wibawa and T. Widiyaningtyas, "Congestion Predictive Modelling on Network Dataset Using Ensemble Deep Learning," vol. 5, no. 4, pp. 1597–1613, 2024, doi: 10.47738/jads.v5i4.333.

[20] M. D. Salunke, V. U. Rathod, Y. K. Mali, R. S. Tambe, A. A. Dange, and S. R. Kothavle, "A Prediction and Classification Process for DDoS Attacks Using Machine Learning," in *2023 7th International Conference On Computing, Communication, Control And Automation (ICCUBEA)*, 2023, pp. 1–6. doi: 10.1109/ICCUBEA58933.2023.10392278.

[21] M. F. Banjar, I. Irawati, F. Umar, and L. N. Hayati, "Analysis of Stroke Classification Using Random Forest Method," *Ilk. J. Ilm.*, vol. 14, no. 3, pp. 186–193, 2022, doi: 10.33096/ilkom.v14i3.1252.186-193.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i4.2482, Copyright ©2025
Submitted: July 22, 2025, Revised: August 25, 2025, Accepted: September 9, 2025, Published: October 15, 2025

[22] E. Söğüt and O. A. Erdem, "A Multi-Model Proposal for Classification and Detection of DDoS Attacks on SCADA Systems," *Appl. Sci.*, vol. 13, no. 10, 2023, doi: 10.3390/app13105993.

[23] S. Dasari and R. Kaluri, "An Effective Classification of DDoS Attacks in a Distributed Network by Adopting Hierarchical Machine Learning and Hyperparameters Optimization Techniques," *IEEE Access*, vol. 12, no. January, pp. 10834–10845, 2024, doi: 10.1109/ACCESS.2024.3352281.

[24] C. A. Lesmana and L. Hakim, "Klasifikasi Serangan DDoS Menggunakan Reqursive Feature Elemination Dan Gradient Boosting," vol. 8, no. 1, pp. 60–69, 2025.

[25] Y. I. Mahendra and R. E. Putra, "Penerapan Algoritma Gradient Boosted Decision Tree (GBDT) untuk Klasifikasi Serangan DDoS," *JINACS (Journal Informatics Comput. Sci. ISSN*, vol. 06, pp. 158–166, 2024, doi: https://doi.org/10.26740/jinacs.v6n01.p158-166.

[26] Z. Azam, M. M. Islam, and M. N. Huda, "Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis Through Decision Tree," *IEEE Access*, vol. 11, no. August, pp. 80348–80391, 2023, doi: 10.1109/ACCESS.2023.3296444.

[27] A. Manimaran, R. GnanaJeyaraman, C. M. B. M. J, M. S., E. Kannan, and M. Sivaram, "An Adaptive Framework for Low-Rate DDoS Detection in Cloud Environments Using Decision Tree Machine Learning Algorithm," in *2024 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS)*, 2024, pp. 1–5. doi: 10.1109/ICBDS61829.2024.10837242.

[28] A. Ilarizky, Y. Prihantono, A. Kurniawan, and M. T. Yusuf, "Analisis Performa Pada Modifikasi VGGNet-16 Untuk Deteksi Serangan Siber : Pendekatan Deep Learning," vol. 10, no. 2, pp. 23–29, 2024.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i4.2482, Copyright ©2025
Submitted: July 22, 2025, Revised: August 25, 2025, Accepted: September 9, 2025, Published: October 15, 2025