Exploring LDoS Attack Detection in SDNs using Machine Learning Techniques

Ali Osman Mohammed Salih
Department of Information Systems, College of Computing and Information Technology, University of Bisha, P.O Box: 551, Bisha, Saudi Arabia
aomohammed@ub.edu.sa (corresponding author)

Received: 27 October 2024 | Revised: 19 November 2024 | Accepted: 27 November 2024

 DOI: https://doi.org/10.48084/etasr.9424

ABSTRACT

This study investigates the application of machine learning algorithms for detecting Low-Rate Denial-of-Service (LDoS) attacks within Software-Defined Networks (SDNs). LDoS attacks are challenging to detect due to their similarity to normal network behavior. This study evaluates the performance of algorithms such as Logistic Regression (LR), K-Nearest Neighbors (KNN), and BIRCH clustering in this challenge. The results show that the LR and BIRCH algorithms outperformed other approaches, achieving a detection accuracy of 99.96% with minimal false positive and negative rates. The models demonstrated a fast detection time of 0.03 seconds, highlighting the potential of machine learning to improve SDN security. The study recommends future work to validate these findings in real-world environments to strengthen security systems.

Keywords-LDoS attacks; DoS detection; SDN; logistic regression; BIRCH algorithm; k-nearest neighbors

I. INTRODUCTION

The proliferation of Internet services has greatly increased cyber security challenges. This has made it necessary to implement strong measures to protect systems, devices, networks, and electronic data from unauthorized access. Machine learning offers a transformative approach to enhance cyber security, providing interactive and cost-effective solutions. In particular, machine learning plays a key role in Intrusion Detection Systems (IDS), monitoring network traffic for malicious activities and alerting systems to potential intrusions. In Intrusion Prevention Systems (IPS), machine learning helps detect and mitigate attacks by identifying malicious packets, resetting connections, and blocking harmful traffic. However, the effectiveness of machine learning in these systems depends on having accurate and comprehensive data that reflects the entire network environment.

Low-Rate Denial of Service (LDoS) attacks are more harmful and difficult to detect. Traditional DoS attacks involve many data packets, which can cause anomalies in the statistical characteristics of network traffic to be detected [1-3]. In contrast, LDoS attacks have reduced average network traffic, as attackers do not need to maintain a high attack rate. Instead, they periodically send short-burst traffic to the victim [4, 5]. Therefore, only a few attack packets, which reduce victims' throughput, can cause large-scale and long-term network paralysis. Additionally, a single LDoS attack flow disguised as a legally-formed pulse flow exhibits the same basic characteristics as normal traffic. Its average packet rate is low, 10-20% of normal data traffic, and it often submerges in normal traffic, making it difficult to detect [6].

An LDoS attack is a type of Denial of Service (DoS) attack. A DoS attack is an older form of network attack that usually uses a botnet to flood the network with data packets. This paralyzes the network and uses up a large amount of its resources. However, with advances in network security, many devices can now quickly detect DoS attacks and then take action to defend against them. New DoS attack types have been developed to avoid detection and increase attack effectiveness, such as LDoS attacks. Unlike DoS attacks, an LDoS attack is a more targeted and precise attack. It exploits vulnerabilities in network protocols to launch attacks and often achieves superior impact at a lower attack cost [7]. The pulses emitted by an LDoS attack are periodic rather than persistent. The average rate of an LDoS attack is very low. Therefore, an LDoS attack has superior concealment and a variable attack approach, which hinders the detection by current network security mechanisms.

Despite the significant threat posed by LDoS attacks, current studies and detection methods focus primarily on high-rate DoS attacks, leaving a notable gap in strategies for identifying and mitigating them in modern Software-Defined Networks (SDNs). The limited studies addressing LDoS attacks in SDNs often lack comprehensive machine-learning approaches tailored to this purpose. This study aimed to bridge this gap by evaluating various machine learning models to improve LDoS detection in SDNs, providing a detailed comparison with existing methods to showcase advances in accuracy and efficiency. Different scenarios are explored to assess the performance of machine learning models, such as Logistic Regression (LR), K-Nearest Neighbors (KNN), and BIRCH clustering, for LDoS attack detection, aiming to contribute valuable insights into network security in SDNs.

II. RELATED WORKS

Research on DoS attack detection covers a wide range, mainly focusing on traditional rather than LDoS attacks. Despite the potential of LDoS attacks to severely disrupt critical network links that handle command, control, and data simultaneously, limited exploration has been made in this area. Most previous studies have focused on identifying LDoS attacks in conventional networking environments, with minimal investigation of their detection in SDNs using advanced artificial intelligence techniques. Therefore, there is an urgent need for research to evaluate the applicability of machine learning methods specifically designed to detect LDoS attacks in SDN contexts. Various studies have proposed different approaches in this field. This summary presents the accuracy, precision, recall, and F1-score metrics of previous studies on different machine learning algorithms used to detect LDoS attacks in SDNs. In [8], various machine learning algorithms were compared, including backpropagation neural networks, achieving high-performance metrics. In [9], the focus was on QS minimization attacks in SDNs using Multilayer Perceptron (MLP). In [10], the AdaBoost algorithm was applied to traditional networks. In [11], the unsupervised BIRCH algorithm was used to detect LDoS attacks, and in [12], LR was used for the same purpose. In [13], K-Nearest Neighbors (KNN) was applied for LDoS attack detection. In [14], Convolutional Neural Networks (CNNs) were used for LDoS attack detection. In [15], the role of various artificial intelligence and machine learning algorithms in wireless network security was explored. Table I presents a comparative analysis of each method. Feature-based methods for detecting LDoS attacks can effectively distinguish between normal and attack traffic through machine learning and data mining. However, selecting features and training models requires significant computational resources and time, necessitating simpler methods or models to detect various types of LDoS attacks. Detection methods based on the time-frequency domain are limited by the Fourier transform's inability to handle non-periodic signals or those with time constraints, and selecting an appropriate wavelet basis function is crucial for improving detection accuracy.

TABLE I. SUMMARY OF RESEARCH STATUS ABOUT DETECTION METHODS.

Algorithm	Accuracy	Precision	Recall	F1-Score
BP	98.9	98.1	97.94	98.12
LR	99.94	99.94	99.94	99.94
MLP	99.92	99.94	99.91	99.93
KNN	99.87	99.92	99.86	99.89
BIRCH	99.82	99.74	99.94	99.84

Table I provides a performance comparison of several machine learning algorithms based on key metrics. The comparison highlights the effectiveness of each algorithm in detecting LDoS attacks. Previous studies have employed a variety of techniques, including LR, KNN, and BIRCH clustering, with varying levels of success. The results of this study demonstrate that the LR and BIRCH algorithms consistently outperform other methods in terms of accuracy and recall, making them highly effective in mitigating LDoS attacks.

III. MACHINE LEARNING TECHNIQUES

Machine learning techniques are pivotal in modern data analysis, being an appropriate choice to identify network intrusions by acquiring traffic characteristics [16].

A. Supervised Learning

Supervised learning involves training a model on labeled data to make predictions or decisions.

- SVMs are effective for classification tasks, finding the optimal hyperplane that separates classes into high-dimensional spaces.
- RF is a versatile ensemble learning method that constructs multiple decision trees and merges them to improve accuracy and robustness.
- Neural networks mimic the human brain's interconnected neurons and are used for complex pattern recognition tasks, ranging from image and speech recognition to natural language processing.

B. Unsupervised Learning

Unsupervised learning deals with unlabeled data, identifying patterns and structures without prior knowledge.

- K-means clustering partitions data into clusters based on similarity, optimizing centroids to minimize intracluster variance.
- Isolation Forest is an anomaly detection method that isolates anomalies by randomly partitioning data into subsets, efficiently identifying outliers.
- Principal Component Analysis (PCA) reduces data dimensionality while preserving variance, aiding in visualization and feature extraction.

C. Reinforcement Learning (RL)

RL focuses on decision-making and learning through interaction with an environment. RL techniques can dynamically learn the best strategies to detect and respond to anomalies in SDNs, where network conditions frequently change. These machine-learning techniques provide advanced analytical and decision-making capabilities in complex SDN environments.

IV. PROBLEM STATEMENT

The number of studies that focus on detecting DoS attacks is significantly greater than those targeting LDoS attacks, despite the substantial damage they can cause, particularly when directed at a shared link that handles both command and control data along with regular network traffic. In addition, previous studies predominantly addressed LDoS attacks within traditional networking environments, with very few studies extending to SDN networks and employing sophisticated intelligence algorithms, particularly artificial intelligence ones. This study aims to explore the feasibility of utilizing machine learning techniques to detect LDoS attacks in SDN networks.

A. Significance of Research

The importance of this research is underscored by several key factors: Most studies predominantly focus on detecting high-rate DoS attacks, with relatively few addressing LDoS ones. Despite being less prominent, LDoS attacks pose significant dangers. From a technological standpoint, SDNs play a pivotal role in modern infrastructure, fostering rapid growth and enhancement of existing services and applications. In this context, machine learning techniques are indispensable for analyzing traffic patterns and detecting anomalies that may signal attacks. By harnessing the collective features within network data, these techniques enable precise identification of potential threats.

B. Research Hypotheses

The Transmission Control Protocol (TCP) is fundamental to the Internet, as it serves as the primary protocol for data transmission across a vast array of Internet services and applications. Figure 1 shows the network topology used to assess the efficiency of data exchange between devices H1 and H5 through TCP. In this network topology, the communication channel is fully utilized to monitor packet delivery. Simultaneously, the iperf tool calculates productivity metrics in one-second intervals. This provides essential network performance metrics under normal operating conditions and during an attack scenario using UDP. Machine learning models are then trained to distinguish between these two network states. Considering scenarios where legitimate devices use UDP, they could inadvertently mimic malicious traffic patterns. Thus, additional scenarios are examined where devices H2 or H3 communicate with H4 using UDP. The features extracted from these communications are incorporated into the dataset to train the machine learning model to differentiate between natural network operations involving TCP or UDP and instances where attacks exploit both protocols simultaneously.

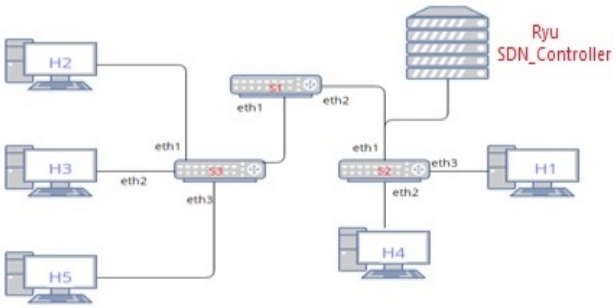

Fig. 1. The network topology used.

C. Summary of the Two Scenarios Described

- Scenario 1 involves H1 and H5 for TCP data exchange, with potential attackers being H2, H3, or both.
- Scenario 2 includes five users, where H1 and H5 manage TCP data exchange. Additionally, one of H2 or H3, along with H4 use UDP for communication and data exchange. Attackers in this scenario are represented by the other of H2 H3. Machine learning algorithms will be used to evaluate the outcomes in both scenarios.

V. RESEARCH METHOD

Mininet was used to evaluate the performance of machine learning algorithms in LDoS attack detection in an SDN [17]. In this setting, a network of virtual machines, switches, controllers, and connections can be created. Mininet provides basic Linux network programs, and its replacement supports the Open Flow protocol, which is a standard in SDN that determines the connection between the controller and the network switches. To evaluate the models, several performance metrics were used, namely:

- False Positive (FP) is called a Type 1 error that indicates a false detection of the event when it did not actually happen. For example, the model predicted that the attack happened, but in fact it did not.
- False Negative (FN) is called a Type 2 error that detects that the event did not happen but it actually did. For example, the model predicts that the attack did not happen, but in fact it happened.
- False Positive Ratio (FPR) indicates the ratio of FP to the actual negatives, given by the following relation:

$$\text{FPR} = \frac{FP}{FP+TN} \quad (1)$$

- False Negative Ratio (FNR) indicates the ratio of FN to actual positives, given by the following relation:

$$\text{FNR} = \frac{FN}{TP+FN} \quad (2)$$

- Accuracy indicates the ratio of the number of correct detections versus the number of total detection is given by:

$$\text{Accuracy} = \frac{Number\ of\ Correct\ predictions}{Total\ number\ of\ predictions\ made} \quad (3)$$

- Precision indicates the number of correct positive to the total positive predictions, given by:

$$\text{Precision} = \frac{TP\ (True\ Positive)}{TP(True\ Positive) + FP(False\ Positive)} \quad (4)$$

- Recall, also called sensitivity, denotes the number of positive predictions correctly made by the model to the total positive instances, given by:

$$\text{Recall} = \frac{TP(True\ Positive)}{TP(True\ Positive) + FN(False\ Negative)} \quad (5)$$

- F1-score is a performance metric that provides the harmonic mean of Precision and Recall, defined by:

$$\text{F1} - \text{score} = 2 * \frac{Precision.Recall}{Precision + Recall} \quad (6)$$

VI. MACHINE LEARNING ALGORITHMS USED

A. Logistic Regression (LR)

LR is a supervised machine-learning algorithm that is used primarily for classification tasks. Its key advantage lies in providing a probabilistic output value between 0 and 1, making it suitable for binary classification problems and adaptable to multiple categories [18].

B. K-Nearest Neighbors (KNN)

KNN is a simple supervised learning algorithm that is used for both prediction and classification. It operates by calculating the distance between the point to be classified and its *K* nearest neighbors, typically using a Euclidean distance metric. The point is then assigned to the group with the highest number of closest neighbors [19, 20]. KNN can perform regression and classification processes [21].

$$\text{Distance} = \sqrt{(x1 - x2)^2 + (y_2 - y_1)^2} \quad (7)$$

C. Neural Networks (NNs)

NNs simulate the human brain's behavior and enable computers to perform pattern recognition and problem-solving tasks common to artificial intelligence, machine learning, and deep learning. They consist of layers comprising nodes connected with weighted edges and thresholds. When a node's output exceeds its threshold, it activates and transmits data to the next layer. NNs are fundamental deep learning algorithms [22, 23].

D. Backpropagation Algorithm

The backpropagation algorithm is a key method for training artificial neural networks, consisting of two main phases: forward propagation and backward propagation. In forward propagation, input data are processed layer by layer to generate predictions, while backward propagation calculates the error between predicted and actual output, adjusting the weights of the neurons to minimize the error. By using gradient calculations and optimization techniques such as Stochastic Gradient Descent (SGD), backpropagation effectively enhances the performance of neural networks in various tasks, including classification and anomaly detection.

VII. METHODOLOGY

A. Preprocessing and Feature Engineering

Discrepancies in the feature values can introduce bias during training. To mitigate this, the dataset was standardized to a unified measurement domain where the standard deviation and mean were normalized to 1. This standardization adjusted the critical parameter values for the machine learning algorithms used. Algorithm performance was evaluated using Grid Search CV [24], to determine the optimal parameters that produce the best evaluation results. Each machine learning algorithm was tested with various parameters, and the final parameters selected were those that achieved the highest accuracy values.

B. Machine Learning Algorithms Used

Given the binary nature of the classification task (attack or no attack), five machine-learning algorithms were employed: LR, NN, KNN, BIRCH (two-step clustering), and a backpropagation. These algorithms were implemented using the Scikit-learn library, each with specific parameters influencing the final model evaluation.

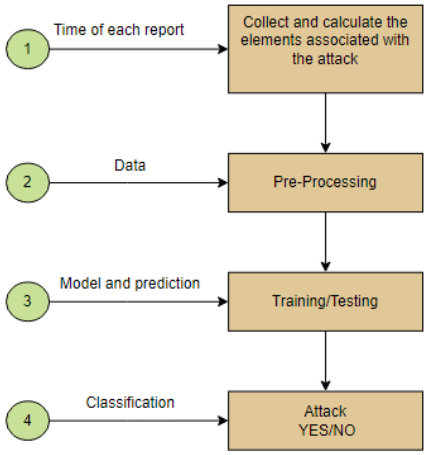

Fig. 2. Research method.

C. Experiments

Experiments were carried out in an Ubuntu environment using an HP laptop with the following specifications: 8th generation Intel Core i7 hexacore processor, Integrated Intel UHD Graphics 630, 16 GB DDR4-2666 SDRAM, and a 512 GB PCIe® NVMe™ M.2 SSD. Each trained model detects the presence of attacks or normal network conditions. The workflow includes launching the Ryu console, starting the Mininet emulator, and executing the IDS.py code. This program trains the device on a pre-generated .csv file by collecting the traffic, counting the pre-generated elements, and then predicting the output in each reporting period. The experiments involved the following scenarios:

- Scenario 1 involves an attacker and a legitimate user using only the TCP. The terminal windows display the TCP communication between H1 and H5 and initially detect normal operation without the presence of an attacker.
- Scenario 2 involves an attacker and a legitimate user using TCP and another using UDP to exchange data, along with a TCP connection between H1 and H5, without any attacker detected according to the network status report.

D. Process of Selecting Items Most Associated with the Attack and Its Plans

The following diagrams illustrate the elements most related to LDoS attacks in both scenarios. LDoS attacks rely primarily on UDP packets, affecting the TCP congestion control mechanisms. Therefore, studying the values and changes in TCP and UDP packets is crucial. This study aimed to identify a unique pattern of network flows in each scenario that can automatically classify the presence of an attack.

The experiments showed variability among several elements to distinguish a clear pattern between the presence or absence of an attack in both scenarios. The elements that provide a clear pattern, resulting in better evaluation results and less complexity, are: the number of TCP packets, the standard deviation of UDP packets, the deviation of TCP and UDP packets, and the TCP/UDP ratio. These elements were used to train machine learning models to classify the presence or absence of an attack.

1) Number of TCP Packets

This represents the count of packets within each time interval (report time) for TCP packets. From Figure 3, the possibility of distinguishing whether an attack exists can be inferred in the two scenarios. The number of TCP packets above a specific threshold (1500 packets) indicates the normal state, while a count below 250 suggests an attack (as the attack reduces the rate of TCP packets). This significant change can be used as an indicator to train the machine learning model to differentiate between the presence or absence of an attack.

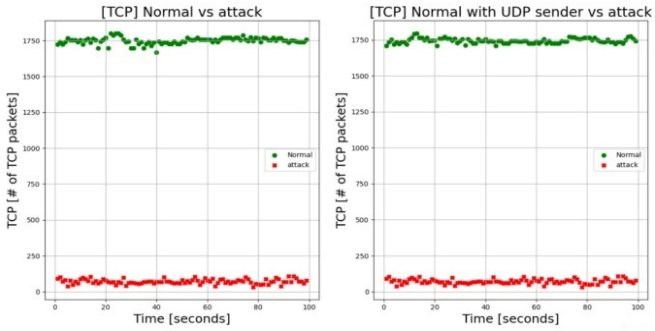

Fig. 3. TCP packets vs. time for each scenario.

2) Standard Deviation of UDP Packets

The standard deviation indicates the extent of dispersion within a dataset, measured mathematically by taking the square root of the variance. This reflects the spread and deviation from the arithmetic mean. The formula for standard deviation (STD) is:

$$STD = \sqrt{Variancea} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

Examining the standard deviation of the UDP packets, as shown in Figure 3, discrepancies between the presence and absence of an attack can be observed in both scenarios. This measure provides a clear indication of whether an attack has occurred, aiding in training the machine learning models.

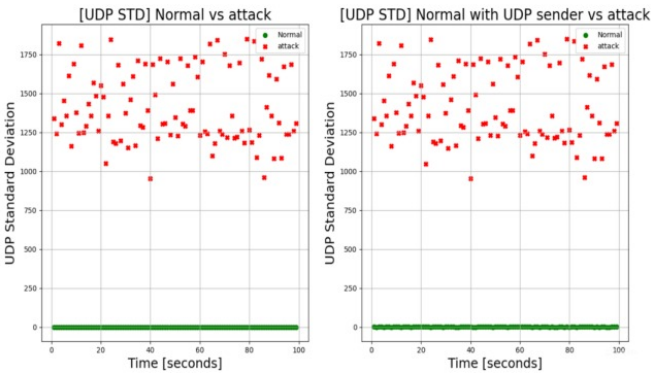

Fig. 4. Plot of UDP standard deviation vs. time for each scenario.

VIII. RESULTS AND DISCUSSION

The evaluation was performed using k = 10 for each algorithm, and the result was taken as the arithmetic mean of all attempts. Figure 5 shows that the results were close. The comparison of machine learning algorithms used all performance measures.

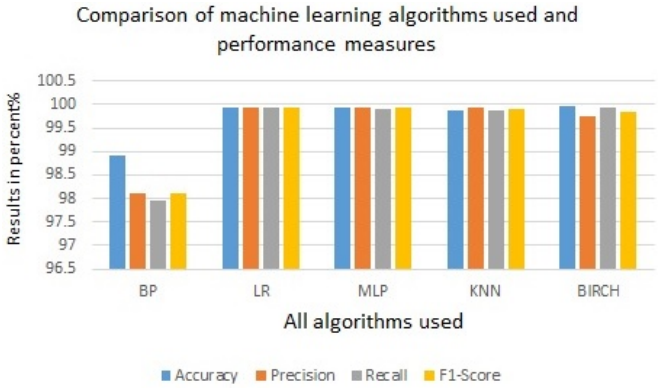

Fig. 5. Comparison of machine learning algorithms.

Since the objective is to design a cyber attack detection system, it is crucial to minimize FN. The cost of an attack that occurs when the model predicts normality is higher than when the model falsely predicts an attack. Therefore, additional tests are required to confirm and address the attack. In detection, it is essential to focus on high Recall values, so the algorithms were compared using Recall, as shown in Figure 6:

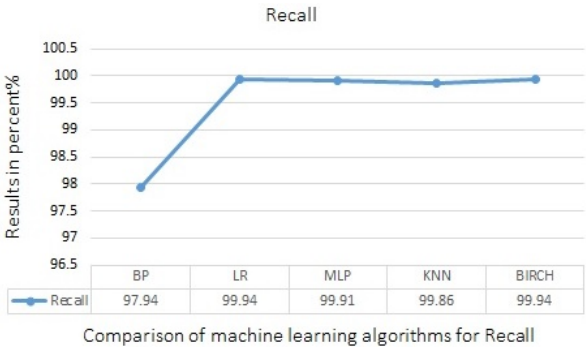

Fig. 6. Comparison of machine learning algorithms based on Recall.

LR and BIRCH algorithms give similar high Recall values, followed by MLP and then KNN.

TABLE II. PERFORMANCE METRICS OF MACHINE LEARNING ALGORITHMS

Algorithms	Accuracy	Precision	Recall	F1-Score
Backpropagation	98.9	98.1	97.94	**98.12**
LR	99.94	99.94	99.94	99.94
MLP	99.92	99.94	99.91	99.93
KNN	99.87	99.92	99.86	99.89
BIRCH	99.96	99.74	99.94	99.84

Table II provides a comprehensive overview of the performance of several machine learning algorithms in

detecting LDoS attacks. The high accuracy, precision, recall, and F1 scores of these algorithms indicate that machine-learning techniques are effective tools for enhancing cybersecurity in SDNs. LR was the top performer, achieving the highest accuracy of 99.94 and excelling in precision, recall, and F1-score, indicating its effectiveness in identifying both attack and non-attack instances. The BIRCH and MLP algorithms also showed high performance with similar metrics. KNN, although slightly less accurate than LR and BIRCH, still demonstrates robustness in LDoS detection. The backpropagation algorithm, although the lowest performer with an accuracy of 98.9, still achieved respectable scores, showcasing its applicability despite reduced effectiveness. Table III shows the performance results of previous studies in LDoS attack detection, along with detection times and whether they can be used in SDNs.

TABLE III. COMPARISON RESULTS OF MODELS FROM THIS AND PREVIOUS STUDIES

Study	Accuracy	FNR %	FPR %	SDN	Machine learning algorithm	Detection time (s)
[10]	97.06	2.94	0.33	NO	Adaboost	~3
[11]	98.06	0.61	1.33	NO	BIRCH	~3
[13]	97.00	3	4.5	NO	Isolation Forest	~5
[14]	99.22	0.78	0.33	NO	KNN	~0.05
[15]	97.1	2.9	0	NO	Convolutional Neural Network	~5
[17]	98.41	1.59	6.21	NO	SVM	~1
[12]	99.58	0.42	0.38	NO	Improved Logistic Regression	~0.05
[25]	99.94	0.05	0.07	YES	Logistic Regression	~1
This study	99.96	0.03	0.06	YES	Logistic Regression	~0.03

Table III shows the significant advances achieved in this study, with the highest detection accuracy of 99.96%, the lowest FNR of 0.03%, and the lowest FPR of 0.06%, outperforming previous ones. Additionally, this approach achieved an ultrafast detection time of 0.03 seconds, setting a new benchmark for real-time LDoS attack detection in SDNs. Unlike many prior studies, the proposed method is specifically tailored for SDN environments, addressing their dynamic and programmable nature. These results demonstrate a well-balanced, efficient, and highly reliable solution for LDoS detection, combining superior accuracy, minimal error rates, and unmatched speed, making it highly applicable to modern cybersecurity challenges.

IX. CONCLUSION

This study demonstrated the feasibility of using machine learning algorithms to detect LDoS attacks in SDNs. The findings indicate that LDoS attacks can significantly affect network performance similar to high-rate DoS attacks, yet they are more challenging to detect due to their subtle traffic patterns. When comparing the results to those of previous studies, it should be noted that although algorithms such as AdaBoost and Isolation Forest reported detection accuracies of around 97% with longer detection times, this study achieved superior accuracies of 99.96% using LR and BIRCH clustering with a much shorter detection time of 0.03 seconds. This showcases not only the robustness but also the improved efficiency of this approach in contrast to prior methods.

Furthermore, although previous studies mainly focused on traditional networks or simpler machine learning models, this study specifically addressed the application within SDN environments, reinforcing its contribution to modern network security. This comparative insight underlines the novelty of this approach, which combines high accuracy and low latency and sets a new benchmark for LDoS detection. Future research should validate these findings in real world settings and expand on adaptive methods for even more resilient security frameworks.

ACKNOWLEDGMENT

The authors acknowledge the Deanship of Graduate Studies and Scientific Research at University of Bisha for supporting this work through the Fast-Track Research Support Program.

REFERENCES

[1] J. Gao, S. Chai, B. Zhang, and Y. Xia, "Research about DoS Attack against ICPS," *Sensors*, vol. 19, no. 7, Jan. 2019, Art. no. 1542, https://doi.org/10.3390/s19071542.

[2] M. P. De Almeida, R. T. De Sousa Júnior, L. J. García Villalba, and T.-H. Kim, "New DoS Defense Method Based on Strong Designated Verifier Signatures," *Sensors*, vol. 18, no. 9, Sep. 2018, Art. no. 2813, https://doi.org/10.3390/s18092813.

[3] J. David and C. Thomas, "Efficient DDoS flood attack detection using dynamic thresholding on flow-based network traffic," *Computers & Security*, vol. 82, pp. 284–295, May 2019, https://doi.org/10.1016/j.cose.2019.01.002.

[4] D. Tang, X. Wang, X. Li, P. Vijayakumar, and N. Kumar, "AKN-FGD: Adaptive Kohonen Network Based Fine-Grained Detection of LDoS Attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 1, pp. 273–287, Jan. 2023, https://doi.org/10.1109/TDSC.2021.3131531.

[5] D. Tang, C. Gao, X. Li, W. Liang, S. Xiao, and Q. Yang, "A Detection and Mitigation Scheme of LDoS Attacks via SDN Based on the FSS-RSR Algorithm," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 4, pp. 1952–1963, Jul. 2023, https://doi.org/10.1109/TNSE.2023.3236970.

[6] D. Tang, S. Wang, B. Liu, W. Jin, and J. Zhang, "GASF-IPP: Detection and Mitigation of LDoS Attack in SDN," *IEEE Transactions on Services Computing*, vol. 16, no. 5, pp. 3373–3384, Sep. 2023, https://doi.org/10.1109/TSC.2023.3266757.

[7] G. Somani, M. S. Gaur, D. Sanghi, M. Conti, and R. Buyya, "DDoS attacks in cloud computing: Issues, taxonomy, and future directions," *Computer Communications*, vol. 107, pp. 30–48, Jul. 2017, https://doi.org/10.1016/j.comcom.2017.03.010.

[8] S. Yoo *et al.*, "Highly Repeatable Rope Winch Design With Multiple Windings and Differential Gear Mechanism," *IEEE Access*, vol. 8, pp. 87291–87308, 2020, https://doi.org/10.1109/ACCESS.2020.2992674.

[9] V. de M. Rios, P. R. M. Inácio, D. Magoni, and M. M. Freire, "Detection of reduction-of-quality DDoS attacks using Fuzzy Logic and machine learning algorithms," *Computer Networks*, vol. 186, Feb. 2021, Art. no. 107792, https://doi.org/10.1016/j.comnet.2020.107792.

[10] D. Tang, L. Tang, R. Dai, J. Chen, X. Li, and J. J. P. C. Rodrigues, "MF-Adaboost: LDoS attack detection based on multi-features and improved Adaboost," *Future Generation Computer Systems*, vol. 106, pp. 347–359, May 2020, https://doi.org/10.1016/j.future.2019.12.034.

[11] D. Tang, R. Dai, L. Tang, and X. Li, "Low-rate DoS attack detection based on two-step cluster analysis and UTR analysis," *Human-centric Computing and Information Sciences*, vol. 10, no. 1, Feb. 2020, Art. no. 6, https://doi.org/10.1186/s13673-020-0210-9.

[12] L. Liu, H. Wang, Z. Wu, and M. Yue, "The detection method of low-rate DoS attack based on multi-feature fusion," *Digital Communications and Networks*, vol. 6, no. 4, pp. 504–513, Nov. 2020, https://doi.org/10.1016/j.dcan.2020.04.002.

[13] D. Tang, L. Tang, W. Shi, S. Zhan, and Q. Yang, "MF-CNN: a New Approach for LDoS Attack Detection Based on Multi-feature Fusion and CNN," *Mobile Networks and Applications*, vol. 26, no. 4, pp. 1705–1722, Aug. 2021, https://doi.org/10.1007/s11036-019-01506-1.

[14] W. Zhijun, L. Wenjing, L. Liang, and Y. Meng, "Low-Rate DoS Attacks, Detection, Defense, and Challenges: A Survey," *IEEE Access*, vol. 8, pp. 43920–43943, 2020, https://doi.org/10.1109/ACCESS.2020.2976609.

[15] M. Waqas, S. Tu, Z. Halim, S. U. Rehman, G. Abbas, and Z. H. Abbas, "The role of artificial intelligence and machine learning in wireless networks security: principle, practice and challenges," *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5215–5261, Oct. 2022, https://doi.org/10.1007/s10462-022-10143-2.

[16] S. Zhan, D. Tang, J. Man, R. Dai, and X. Wang, "Low-Rate DoS Attacks Detection Based on MAF-ADM," *Sensors*, vol. 20, no. 1, Jan. 2020, Art. no. 189, https://doi.org/10.3390/s20010189.

[17] Y. Yan, D. Tang, S. Zhan, R. Dai, J. Chen, and N. Zhu, "Low-Rate DoS Attack Detection Based on Improved Logistic Regression," in *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, Zhangjiajie, China, Aug. 2019, pp. 468–476, https://doi.org/10.1109/HPCC/SmartCity/DSS.2019.00076.

[18] "Mininet: An Instant Virtual Network on Your Laptop (or Other PC) - Mininet." http://mininet.org/.

[19] "Birch," *scikit-learn*. https://scikit-learn/stable/modules/generated/sklearn.cluster.Birch.html.

[20] "Logistic Regression," *scikit-learn*. https://scikit-learn/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.

[21] W. Ismaiel, A. Alhalangy, A. O. Y. Mohamed, and A. I. A. Musa, "Deep Learning, Ensemble and Supervised Machine Learning for Arabic Speech Emotion Recognition," *Engineering, Technology & Applied Science Research*, vol. 14, no. 2, pp. 13757–13764, Apr. 2024, https://doi.org/10.48084/etasr.7134.

[22] "Confusion_matrix," *scikit-learn*. https://scikit-learn/stable/modules/generated/sklearn.metrics.confusion_matrix.html.

[23] "GridSearchCV," *scikit-learn*. https://scikit-learn/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.

[24] "KNeighborsClassifier," *scikit-learn*. https://scikit-learn/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html.

[25] D. Y. Yousef, "Evaluating the performance of machine learning techniques in detecting LDoS attacks in SDNs.," *Journal of Engineering Sciences & Information Technology*, vol. 6, no. 6, 2022.