Indonesian Journal of Electrical Engineering and Computer Science
Vol. 32, No. 3, December 2023, pp. 1503~1511
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v32.i3.pp1503-1511

Machine learning based detection of DDoS attacks in software defined network

Charulatha Kannan[1], Rajendiran Muthusamy[2], Vimala Srinivasan[1], Vivek Chidambaram[1], Kiruthika Karunakaran[3]

[1]Department of Artificial Intelligence and Data Science, Panimalar Engineering College, Chennai, India
[2]Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, India
[3]Department of Computer Science and Business Systems, Panimalar Engineering College, Chennai, India

Article Info

Article history:

Received Mar 4, 2023
Revised Aug 4, 2023
Accepted Sep 4, 2023

Keywords:

Decision tree
DoS attack
Mininet
Ryu controller
Support vector machine

ABSTRACT

Nowadays, software defined networking (SDN) offers benefits in the area so fautomation, elasticity, and resource consumption. However, evidenceis there that SDN controller may undergo certain defeat for the network structure, particularly as the yare targeted by attacks like denial of service (DoS). Due to this network traffic has increased tremendously and attacked the server severely. To handle this issue, weused the Ryu controller and Mininet tool to identify and all eviate the DoS attack by the machine learning (ML) algorithm. Since ML is deemed as themain method for detecting peculiarities, the detection of DoS attacks was done through ML based classification. In this paper, several ML techniques were used to identify the DoS attack, and the traffic which is causing the attack has been dropped immediately to avoid congestion. The proposed work hasbeen simulated in Mininet and the results show that the proposed work detects DoS attacks well and achieves good accuracy.

Corresponding Author:

Rajendiran Muthusamy
Department of Computer Science and Engineering, Panimalar Engineering College
Poonamallee, Chennai, Tamilnadu, India
Email: mrajendiran@panimalar.ac.in

1. INTRODUCTION

The current trends are based on networking technologies which seem to be in high demand and issues. The increase in demand also leads to a rise in the number of users of the network for various purposes in various fields [1]. The networking concepts are not only applied and made by the software technical people but also by the civilians of the world. Even a commoner uses a website or a web application by requesting and receiving services over the network approximately more than 4,000 times.

As this is the case, networking traffic occurs, which seems not to be solvable sometimes. Network congestion is a result of massive network traffic. As a result, network and traffic management problems developed as well. The traffic is forwarded across a number of switches and routers in a traditional network by integrating hardware and software [2], [3]. The traditional was purely based on the hardware networking components for networking before the rise of the Software defined networking (SDN) concept. The traditional network is basically of static nature which makes use of fixed and dedicated hardware devices to control network traffic [4].

SDN is a networking architecture that defeats the problems in the state of art network by distinguishing the network controlling plane from the forwarding plane in order to simplify and improve the network control. SDN architecture has a centralized controller which monitors and collects all network statistics for the effective organization of resources and packets routing. SDN has the separation of control plane functions from the forwarding functions, which makes of greater for automation and programmability. The SDN networking will only do the forwarding without any interpretation. The comparison of traditional and software-defined networking is shown in Figure 1. The current trends are based on networking technologies which seem to be in high demand and issues. The increase in demand also leads to a rise in the number of users of the network for various purposes in various fields [5], [6]. The networking concepts are not only applied and made by the software technical people but also by the civilians of the world. Even a commoner uses a website or a web application by requesting and receiving services over the network approximately more than 4,000 times.

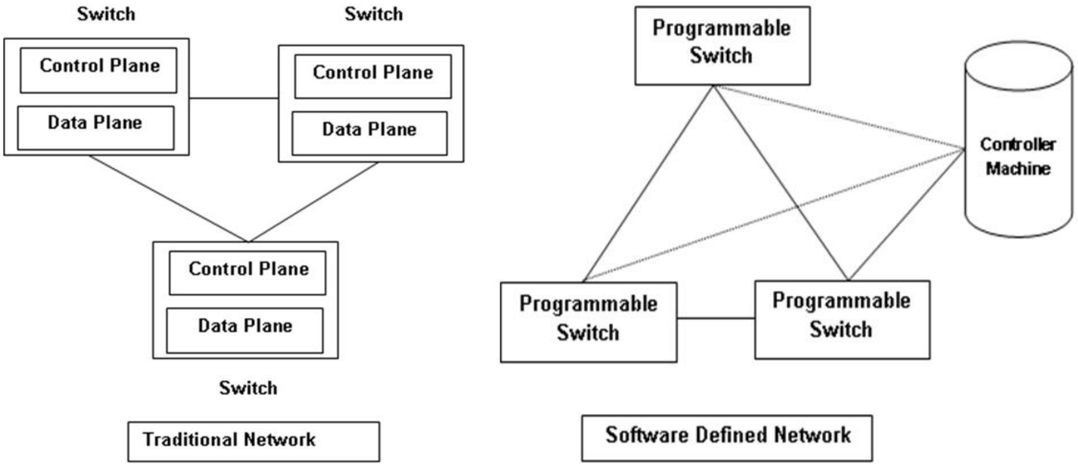

Figure 1. Traditional network vs SDN

Rules and actions to take will be present in the SDN switches and these actions are coordinated by the programmable controller in the controller plane [7]. The open flow protocol is an open interface through which communication between the planes occurs. In SDN, traffic engineering, security, and dynamic management of resources are promoted by the network programmability of SDN [8]. SDN traffic prediction system is capable of predicting the future traffic that is expected to occur and also the congestion in the network using offline historical and also online real-time data. This paper shows the proposed work which detects the denial of service (DoS) attacks in SDN using the Ryu controller and mininet tool [9].

2. METHOD

2.1. Ryu controller

Ryu is an SDN controller framework that can be leveraged to build custom SDN applications, including those focused on DDoS detection and mitigation. The Ryu controller can act as the brain of the SDN network, where it receives network traffic information from switches and makes intelligent decisions based on predefined rules or algorithms to identify potential DDoS attacks [10], [11]. SDN is a network architecture that separates the control plane from the data plane in network devices, allowing for more flexible and programmable network management. It serves as an SDN controller, which means it manages network devices and their traffic flows in an SDN environment. SDN controllers like Ryu provide a centralized point of control for SDN networks. Ryu is often used with OpenFlow, a standard communication protocol between the SDN controller and the network switches [12]. OpenFlow allows the controller to instruct switches on how to handle traffic flows.

2.2. Roles of Ryu in DDoS detection and mitigation

Traffic monitoring: Ryu can collect flow statistics and packet information from the network switches to monitor traffic patterns in real-time [13]. Anomaly detection: Using machine learning (ML) algorithms or predefined thresholds, the Ryu controller can detect unusual traffic patterns that might indicate a DDoS attack:

- Flow classification: Ryu can classify network flows based on their characteristics and prioritize important flows over suspicious ones during a DDoS attack.
- Traffic engineering: The controller can dynamically reconfigure network paths and allocate resources to mitigate the impact of the DDoS attack.

– Blackhole routing: Ryu can instruct switches to forward DDoS traffic to a blackhole (a null route) to prevent it from reaching the target.

2.3. Mininet

Mininet provides a platform for emulating an SDN network on a single physical machine or a cluster of machines. It allows users to create custom network topologies and emulate traffic scenarios, including DDoS attacks, in a controlled environment [14]. Mininet is a network emulator that creates virtualized network environments for testing and simulating computer networks within a single physical machine. Users can interact with the emulated network in real-time, enabling monitoring, troubleshooting, and the evaluation of network application behavior [15], [16].

2.4. Roles of mininet in DDoS detection and mitigation

Testing DDoS scenarios: Mininet allows researchers and developers to simulate various DDoS attack scenarios by generating high volumes of traffic to mimic attack patterns. Evaluating Ryu applications: Mininet can be used to evaluate the effectiveness of Ryu-based DDoS detection and mitigation applications in a simulated network environment before deploying them in a production network [17]. Network administrators and security experts can use Mininet to develop and fine-tune DDoS mitigation strategies, including traffic filtering, rate limiting, and redirection. They can experiment with different mitigation techniques to see how they perform under attack conditions.

2.5. Combining Ryu and Mininet

By combining Ryu and Mininet, users can create a controlled testbed to develop, validate, and optimize DDoS detection and mitigation strategies using SDN. The Ryu controller, with its programmability, real-time traffic analysis, and flow control capabilities, can work in tandem with Mininet’s emulation capabilities to effectively identify and respond to DDoS attacks in a simulated SDN environment. This integrated approach can lead to more robust and efficient DDoS mitigation solutions in actual SDN deployments [18].

2.6. Real-time DDoS detection

Ryu's ability to gather real-time network traffic information from switches allows it to perform continuous monitoring of network flows. By analyzing the flow statistics, packet headers, and payload data, Ryu can implement sophisticated DDoS detection mechanisms [19]. These mechanisms can range from simple threshold-based approaches to more advanced ML ing algorithms, such as anomaly detection and behavioral analysis. For example, Ryu can use ML models to learn the normal behavior of the network and identify deviations from this behavior that may indicate DDoS attacks. When suspicious patterns are detected, Ryu can trigger the appropriate response to mitigate the attack. The technique can include flow-based mitigation, blackhole routing, cooperative mitigation and mininet for scalable testing [20], [21].

In the case of flow-based mitigation, SDN, with Ryu as the controller, provides fine-grained control over network flows. When a DDoS attack is detected, Ryu can dynamically adjust flow rules in switches to divert, rate-limit, or drop malicious traffic. For instance, Ryu can steer DDoS traffic away from critical network resources, ensuring that legitimate traffic still reaches its intended destinations. In the case of severe DDoS attacks, where immediate action is required, Ryu can instruct switches to implement blackhole routing. This involves directing all the traffic destined for the attacked resource to a null route or a blackhole, effectively discarding the malicious traffic and preventing it from reaching the target [22].

In larger SDN deployments, multiple Ryu controllers can work together, each responsible for a specific domain of the network. When a DDoS attack is detected in one domain, the affected controller can signal other controllers to implement cooperative mitigation strategies. This collaboration helps to contain the DDoS attack’s impact and improve the overall network’s resilience [23]. Mininet’s ability to emulate various network topologies and traffic scenarios allows users to test Ryu-based DDoS detection and mitigation applications at scale [24], [25]. Researchers and developers can simulate different attack scenarios and validate the effectiveness of their strategies before deploying them in a production environment. Mininet’s flexibility and scalability make it an essential tool for testing DDoS resilience in SDN networks.

3. PROPOSED SYSTEM

In the developed networking architecture, the threat to the safety of the network system is a splendid task. The major threat is found to be the DDoS attack. The proposed work is to identify the DDoS attack in the SDN using the techniques of ML. Also, to mitigate the attack by specifically identifying the port which causes the attack and blocking the port. The ML algorithm is trained with the data collected in training mode to detect and differentiate between the attack and normal traffic [26]. In training mode one of the ports is made to ping one of the servers and by generating normal traffic and the collected data is saved in a comma separated values (CSV) file format. The attack traffic is made by creating a syn attack on the server by pinging from any of the ports and the collected data is stored in the same CSV file. The stored file is given as input to the ML algorithm and the Ryu controller is run in detection mode. The ML algorithm classifies normal and attack traffic and displays the result as either 0 or 1 meaning normal or attack respectively. Also, this port that attacks the network is identified and blocked by dropping the packets from that port for a specified time interval [13], [27]. Once the time interval expires the port is set free to ping over again in normal conditions. This method not only detects the attack but also mitigates the attack and ensures the security of the network.

The network topology mentioned in Figure 2, consists of three switches and a pair of servers for sending and receiving requests and responses. The entire network topology is connected with a Ryu controller, a type of SDN controller which helps in efficient management and control of the network traffic. The network is of single-layered topology with four ports in which each port is run using some transport layer protocols. The used protocols are ICMP, TCP/IP, and UDP [28]. Each of the port run in all the protocols and the network traffic is formed through one of the ports and the data in testing mode is collected. The attack traffic is generated via the xterm terminal on one of the servers from one of the hosts. The type of attack caused is a syn-attack which is a type of DoS attack which makes use of the internet communication protocol TCP/IP and bombards the server with multiple requests to make a large queue for requests ofservices and thus making the host unresponsive due to the heavily loaded traffic [29].

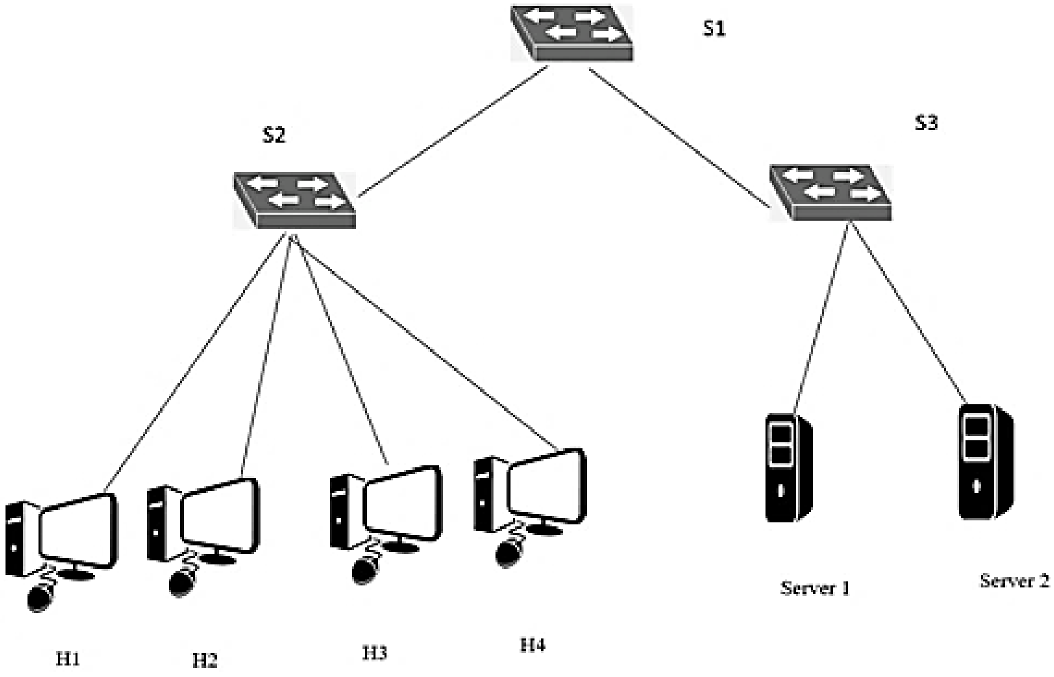

Figure 2. Network topology

The ML algorithm is trained with the data collected in training mode to detect and differentiate between traffic that could be either normal or attack. The workflow of data collection is shown in Figure 3. In training mode one of the ports is made to ping one of the servers and by generating normal traffic and the collected data is stored in a CSV file format [30], [31]. The attack traffic is made by creating a syn attack on the server by pinging from any of the ports and the collected data is stored in the same CSV file. The data collected in normal traffic mode D1 and attack traffic mode D2 is trained using ML algorithms and it is depicted in Figure 4.

3.1. Proposed algorithm

Algorithm 1 the proposed algorithm defines how exactly the working flow goes on. It starts from setting up the project and ends with detection and mitigation of DDOS attack in SDN architecture. On succefull end of the algorithm is once the attack is detected it will be mitigated and the packets will be sent immediately.

The flow of the project starts with the data collection for training the system with the normal traffic and attack traffic. The datas are stored in a CSV file format. The detection of the attack is made by using the ML classifier to differentiate between attack and normal, and then the attack is mitigated which is referred to in Figure 5.

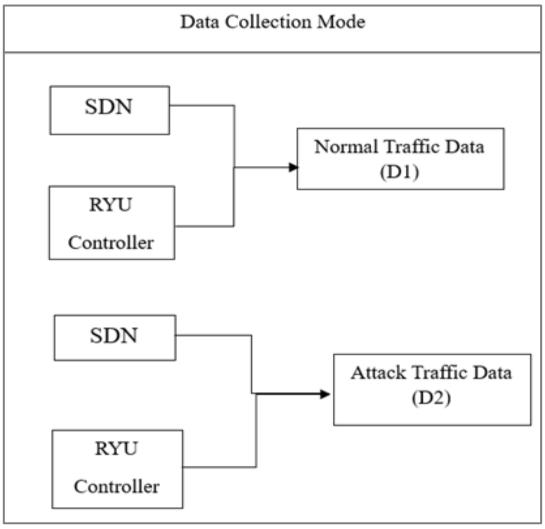

Figure 3. Work flow of data collection

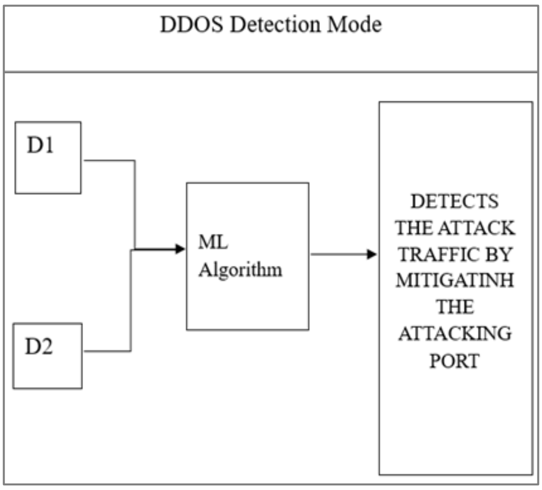

Figure 4. Work flow of DDOS detection

Algorithm 1. The proposed algorithm defines how exactly the working flow goes on

```
Step1: Open the terminal
Step2: Set up the project in data collection mode (Appmode=0, Trafficmode=0) for normal
traffic generation. Step3: Start the Ryu controller and the topology in the collection mode.
Step4: Ping any of the servers from one of the ports using all three protocols.
Step5: Save the collected data in a .csv file.
Step6: Stop the controller and topology after the data collection in normal traffic.
Step7: Again start the Ryu controller and the topology in attack traffic mode (App mode=0,
Traffic mode=1)
Step8: Ping any of the servers from one of the ports with the syn attack traffic and store
the collected data in the same CSV file created in Step 5
Step9: Repeat step 6 and open a new terminal.
Step10: Edit the algorithm to DoS detection mode (App mode=1) and repeat step 1 for dos
attack detection mode.
Step11: Testing the attack by generating normal traffic and attack traffic by pinging the
server from the ports.
```

```
Step12: While an attack is detected it will be mitigated simultaneously and the flow can be
viewed in the dump flow of the packets.
Step13: Once the attack is identified, the particular port will be suspended for a specific
time and will be again active and eligible to send the packets again.
Step 14: Thus the Dos attack is successfully detected and mitigated in the SDN
architecture.
```

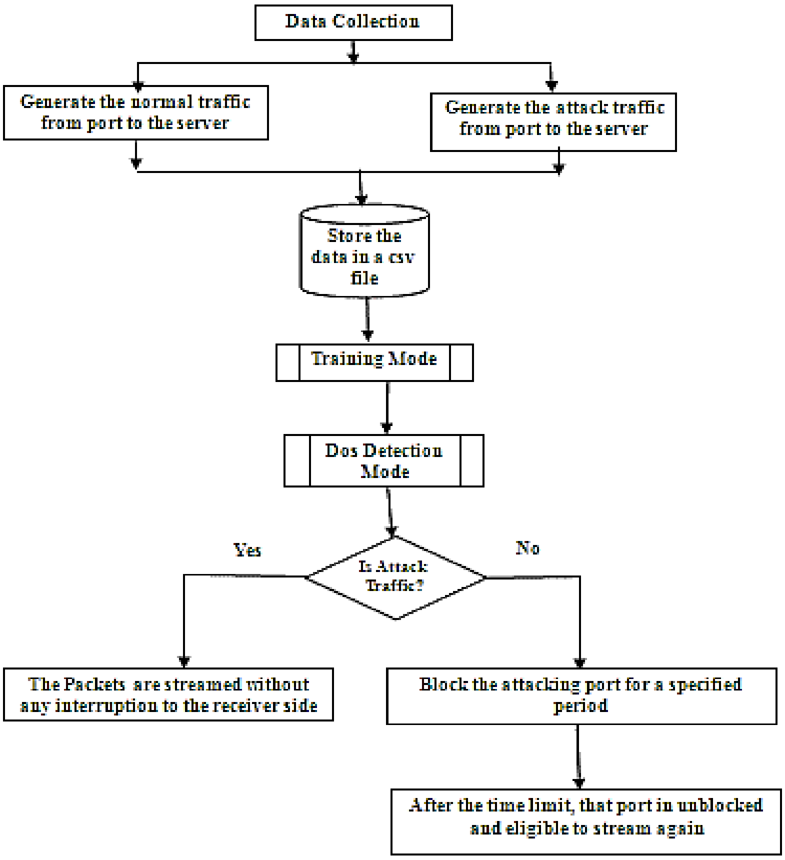
Data Collection
Generate the normal traffic from port to the server
Generate the attack traffic from port to the server
Store the data in a csv file
Training Mode
Dos Detection Mode
Is Attack Traffic?
Yes
No
The Packets are streamed without any interruption to the receiver side
Block the attacking port for a specified period
After the time limit, that port in unblocked and eligible to stream again

Figure 5. Flow diagram of proposed work

4. EXPERIMENTAL RESULTS AND DISCUSSION

The experiments are especially conducted on Mininet, which has been the testing environment for many experimenters for the last few years. Researchers conduct experiments using hardware testbeds with the result of experimental platforms. Here we estimated our method's performance in this section by comparing it to other methods and proving its effectiveness through test-bed experiments. The testbed contains three switches, four hosts, and two servers in it. The network metrics are collected as training data and then we train the ML algorithm with the training data. In live traffic, the attack is made in the network and using the ML algorithm the traffic is categorized based on the collected training data as an attack or normal traffic. ML algorithm then classified as attack traffic or normal traffic. Based on the response generated, if it is an attack, SDN controller performs the prevention (block the attacked port). We then compare and scrutinize the performance metric values such as accuracy score, detection rate, confusion matrix, precision, recall, and F1-score. Subsequently, we trained and tested the six classifiers support vector machine (SVM), decision tree, gaussian Naïve Bayes, random forest, extra tree classifier, and neural network classifier). Furthermore, we compared variable values for the identical classifier parameter across different classifiers to optimize each classifier. From the result fetched we have classified the obtained results and tabulated them. The comparisons are illustrated in graphs the metrics used for classification.

4.1. Metrics used

The accuracy is determined based on the number of data sets in which the model analysts are accurate. A precision (P) measure shows how many packets got attacked actually in the data packet considered as attack type by the model. Recall rate (R) represents a percentage of all packets the model identifies as attack types [32]. Using the F1-score (F1) makes it possible to determine the accuracy of model performance by using the harmonic average of precision and recall the various metrics mentioned can be calculated using the following relations:

$$Accuracy\ =\frac{TN+TP}{FN+TN+TP+FP} \tag{1}$$

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

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

$$F1-score\ =\ 2*\frac{PR}{P+R} \tag{4}$$

Table 1 revealed the comparing results of various ML algorithms applied in the proposed DoS detection algorithm. Considering the accuracy rate, the decision tree, forest of random trees, and extra tree classifier have achieved 100%. Next to that, gaussian Naïve Bayes has achieved an accuracy rate of 96.05%. The SVM classifier has achieved the lowest accuracy rate of 94.35%.

When considering the cross-validation score, the extra tree classifier has achieved a score of 1 whereas the decision tree and forests of random tree have achieved 0.99. The gaussian Naïve Bayes and SVM has cross-validation score of 0.97 and 0.96 respectively. Similarly, when the detection rate is considered, all algorithms except gaussian Naïve Bayes have achieved a rate of 1. When algorithms were analyzed in terms of false alarm rate, the decision tree, forests of random trees, and extra tree classifier has zero false alarm rates whereas gaussian Naïve Bayes has 0.42 and SVM has the highest false alarm rate of 0.65. The precision, recall and F1-score of all the algorithms were almost 1 which means that the DoS attack classification was done correctly. The extra tree classifier outperforms all the algorithms in considering all the metrics like accuracy rate, detection rate, false alarm rate, precision, recall, and F1-score.

Table 1. Comparison of ML algorithms in the detection of DoS

Algorithm	SVM	Decision tree	Gaussian Naïve Bayes	Forestsof random trees	Extra tree classifier
Accuracy score	94.35	100.0	96.05	100.0	100.0
Cross validation score	0.96	0.99	0.97	0.99	1
Detection rate	1.0	1.0	0.99	1.0	1.0
False alarm rate	0.65	0.0	0.42	0.0	0.0
Precision	0.941	1	0.961	1	1
Recall	1	1	0.996	1	1
F1-score	0.969	1	0.96	1	1

5. CONCLUSION

In this work, we have examined the summary of SDN and the outcomes of DoS and DDoS attacks in SDN. The centralization of network management functionality at the controller makes DoS attacks more likely in SDN, where they are vital in traditional networks. With the refined technique of SDN, the protection of SDN has evolved into one of the most important parts of network technology. With SDN, programmable networking has also gradually developed. We have proposed a solution using ML techniques to predict DDoS attacks against SDN networks and we have accomplished this using the tool Mininet. The proposed work has revealed high precision and efficiency under two types of attacks. we have tabulated a summary of ML methods for identifying DDoS attacks in SDN. In the future, we will focus on security challenges in DDoS attack identification and will also provide a reduction in the SDN domain. Behind that, we will examine the attack and scan the mechanisms that are currently in place in present networks and concern whether they can be deployed in environments.

REFERENCES

[1] M. Yue, H. Wang, L. Liu, and Z. Wu, “Detecting DoS attacks based on multi-features in SDN,” *IEEE Access*, vol. 8, pp. 104688–104700, 2020, doi: 10.1109/ACCESS.2020.2999668.

[2] S. Gao, Z. Peng, B. Xiao, A. Hu, Y. Song, and K. Ren, “Detection and mitigation of DoS attacks in software defined networks,” *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 1419–1433, 2020, doi: 10.1109/TNET.2020.2983976.

[3] J. E. Varghese and B. Muniyal, “An efficient IDS framework for DDoS attacks in SDN environment,” *IEEE Access*, vol. 9, pp. 69680–69699, 2021, doi: 10.1109/ACCESS.2021.3078065.

[4] 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, 2020, doi: 10.1109/ACCESS.2020.3019330.

[5] M. A. Aladaileh, M. Anbar, I. H. Hasbullah, Y. W. Chong, and Y. K. Sanjalawe, “Detection techniques of distributed denial of service attacks on software-defined networking controller-a review,” *IEEE Access*, vol. 8, pp. 143985–143995, 2020, doi: 10.1109/ACCESS.2020.3013998.

[6] D. Tang, Y. Yan, S. Zhang, J. Chen, and Z. Qin, “Performance and Features: Mitigating the low-rate TCP-targeted DoS attack via SDN,” *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 428–444, 2022, doi: 10.1109/JSAC.2021.3126053.

[7] T. Wang, Z. Guo, H. Chen, and W. Liu, “BWManager: mitigating denial of service attacks in software-defined networks through bandwidth prediction,” *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1235–1248, 2018, doi: 10.1109/TNSM.2018.2873639.

[8] 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.

[9] L. Tan, Y. Pan, J. Wu, J. Zhou, H. Jiang, and Y. Deng, “A new framework for DDoS attack detection and defense in SDN environment,” *IEEE Access*, vol. 8, pp. 161908–161919, 2020, doi: 10.1109/ACCESS.2020.3021435.

[10] S. Dong and M. Sarem, “DDoS attack detection method based on improved KNN with the degree of DDoS attack in software-defined networks,” *IEEE Access*, vol. 8, pp. 5039–5048, 2020, doi: 10.1109/ACCESS.2019.2963077.

[11] Z. A. El-Houda, L. Khoukhi, and A. S. Hafid, “Bringing intelligence to software defined networks: mitigating DDoS attacks,” *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2523–2535, 2020, doi: 10.1109/TNSM.2020.3014870.

[12] W. Zhijun, X. Qing, W. Jingjie, Y. Meng, and L. Liang, “Low-rate DDoS attack detection based on factorization machine in software defined network,” *IEEE Access*, vol. 8, pp. 17404–17418, 2020, doi: 10.1109/ACCESS.2020.2967478.

[13] L. Yang and H. Zhao, “DDoS attack identification and defense using SDN based on machine learning method,” in *Proceedings - 2018 15th International Symposium on Pervasive Systems, Algorithms and Networks, I-SPAN 2018*, 2019, pp. 174–178, doi: 10.1109/I-SPAN.2018.00036.

[14] A. O. Sangodoyin, M. O. Akinsolu, P. Pillai, and V. Grout, “Detection and classification of DDoS flooding attacks on software-defined networks: a case study for the application of machine learning,” *IEEE Access*, vol. 9, pp. 122495–122508, 2021, doi: 10.1109/ACCESS.2021.3109490.

[15] 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.

[16] Kandoi, Rajat, and M. Antikainen., “Denial-of-service attacks in OpenFlow SDN networks,” in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015, pp. 1322–1326.

[17] P. Zhang, H. Wang, C. Hu, and C. Lin, “On denial of service attacks in software defined networks,” *IEEE Network*, vol. 30, no. 6, pp. 28–33, 2016, doi: 10.1109/MNET.2016.1600109NM.

[18] R. Durner, C. Lorenz, M. Wiedemann, and W. Kellerer, “Detecting and mitigating denial of service attacks against the data plane in software defined networks,” 2017, doi: 10.1109/NETSOFT.2017.8004229.

[19] 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*, 2016, pp. 2576–2581, doi: 10.1109/ICACCI.2016.7732445.

[20] O. Rahman, M. A. G. Quraishi, and C. H. Lung, “DDoS attacks detection and mitigation in SDN using machine learning,” in *Proceedings - 2019 IEEE World Congress on Services, SERVICES 2019*, 2019, pp. 184–189, doi: 10.1109/SERVICES.2019.00051.

[21] R. Santos, D. Souza, W. Santo, A. Ribeiro, and E. Moreno, “Machine learning algorithms to detect DDoS attacks in SDN,” *Concurrency and Computation: Practice and Experience*, vol. 32, no. 16, 2020, doi: 10.1002/cpe.5402.

[22] 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 and T 2020 - Proceedings*, 2021, pp. 609–612, doi: 10.1109/PICST51311.2020.9467963.

[23] W. Sun, Y. Li, and S. Guan, “An improved method of DDoS attack detection for controller of SDN,” in *2019 IEEE 2nd International Conference on Computer and Communication Engineering Technology, CCET 2019*, 2019, pp. 249–253, doi: 10.1109/CCET48361.2019.8989356.

[24] B. H. Lawal and A. T. Nuray, “Real-time detection and mitigation of distributed denial of service (DDoS) attacks in software defined networking (SDN),” in *26th IEEE Signal Processing and Communications Applications Conference, SIU 2018*, 2018, pp. 1–4, doi: 10.1109/SIU.2018.8404674.

[25] M. Ruaro, L. L. Caimi, and F. G. Moraes, “A systemic and secure SDN framework for noc-based many-cores,” *IEEE Access*, vol. 8, pp. 105997–106008, 2020, doi: 10.1109/ACCESS.2020.3000457.

[26] K. M. Sudar, M. Beulah, P. Deepalakshmi, P. Nagaraj, and P. Chinnasamy, “Detection of distributed denial of service attacks in SDN using machine learning techniques,” in *2021 International Conference on Computer Communication and Informatics (ICCCI)*, 2021, doi: 10.1109/ICCCI50826.2021.9402517.

[27] M. M. Isa and L. Mhamdi, “Native SDN intrusion detection using machine learning,” in *2020 IEEE Eighth International Conference on Communications and Networking (ComNet)*, 2020, doi: 10.1109/ComNet47917.2020.9306093.

[28] V. Deepa, K. M. Sudar, and P. Deepalakshmi, “Detection of DDoS attack on SDN control plane using hybrid machine learning techniques,” in *Proceedings of the International Conference on Smart Systems and Inventive Technology, ICSSIT 2018*, 2018, pp. 299–303, doi: 10.1109/ICSSIT.2018.8748836.

[29] V. Deepa, K. M. Sudar, and P. Deepalakshmi, “Design of ensemble learning methods for DDoS detection in SDN environment,” in *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, 2019, doi: 10.1109/ViTECoN.2019.8899682.

[30] A. Ahmad, E. Harjula, M. Ylianttila, and I. Ahmad, “Evaluation of machine learning techniques for security in SDN,” in *2020 IEEE Globecom Workshops (GC Wkshps)*, 2020, doi: 10.1109/GCWkshps50303.2020.9367477.

[31] N. Meti, D. G. Narayan, and V. P. Baligar, “Detection of distributed denial of service attacks using machine learning algorithms in software defined networks,” in *2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017*, 2017, vol. 2017-Janua, pp. 1366–1371, doi: 10.1109/ICACCI.2017.8126031.

[32] T. K. Luong, T. D. Tran, and G. T. Le, “DDoS attack detection and defense in SDN based on machine learning,” in *Proceedings - 2020 7th NAFOSTED Conference on Information and Computer Science, NICS 2020*, 2020, pp. 31–35, doi: 10.1109/NICS51282.2020.9335867.

BIOGRAPHIES OF AUTHORS

Charulatha Kannan received the B.E., and M.E., degrees in computer science and engineering from Panimalar Engineering College, Anna University, Tamil Nadu, India. She is currently working as an assistant professor in the department of artificial intelligience and data science, Panimalar Engineering College, Chennai, Tamilnadu. She can be contacted at email: charulathakannan1971@gmail.com.

Rajendiran Muthusamy is a professor in the department of computer science and engineering, Panimalar Engineering College, Chennai, India. He holds a Ph.D. degree in computer science and engineering with specialization in mobile ad hoc networks. His research areas are mobile networks, machine learning, data analysis, and routing protocol. He can be contacted at email: mrajendiran@panimalar.ac.in.

Vimala Srinivasan graduated in master of computer application from Maduai Kamaraj University, Madurai, India in 1999. She has been working as assistant professor in various Engineering Colleges affiliated to Anna University, Chennai, India from 1999 to 2013. She obtained M.Tech. PG degree in Computer Science and Engineerring from Anna University, Chennai, India in 2016. She has 24 years of teaching experience and guided many UG and PG scholars. Currently working as associate professor in artificial intelligence and data science department, Panimalar Engineering College, Poonamallee, Chennai, Tamil Nadu, India. She can be contacted at email: vimalakumaran@gmail.com.

Vivek Chidambaram received B.E. degree in computer science and engineering from Anna University, India in 2010. He obtained M.Tech. in 2012 from JNTU, Anna University, India. Pursuing Ph.D. in SRM University, Chennai. He is currently working as assistant professor in the department of artificial intelligence and data science at Panimalar Engineering College, Chennai, Tamil Nadu, India. His research interests include artificial intelligence and computer vision. He can be contacted at email: vksundar7@gmail.com.

Kiruthika Karunakaran is currently pursing the Ph.D. degree at department of computer science and engineering with Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Tamil Nadu, India. She did her B.Tech. in Anna Univeristy and M.Tech. from Sathyabama University, in 2005 and 2011, respectively. She served as assistant professor for 10 years before enrolling to Ph.D. Her research interests are data structure, design and analysis of algorithms, compiler design medical imaging, and machine and deep learning. She can be contacted at email: kkiruthikamtech@gmail.com.