Chapter 26
DDoS Attack Detection and Prevention with Local Monitoring Attack Detection (Lm-Ad) Scheme Under SDN Context in Cloud Computing

K. Radha and R. Parameswari

26.1 Introduction

Distributed Denial of Service (DDoS) attack is the most dangerous. It allows coordinated attacks on a network environment. The primary goal of this attack is to malfunction the entire system and make it unable to provide any service to authorized users. These infected machines will be under the attackers' control and implement DDoS attacks [1]. 1999, the first denial-of-service assaults (DDoS) were identified and considered significant cyber-attacks [2]. According to a Redware poll, these DDoS attacks have become a major security concern for internet-related firms [3]. In SDN, DDOS detection falls into three categories: machine learning-based methods, deep learning-based methods and information statistics-based methods [4]. Information statistics-based approaches commonly use information entropy. This method attempts to identify network data anomalies by predicting packet property modifications like source or target IP addresses [5]. The detection accuracy level is determined by the threshold value set for entropy. Expert knowledge and subjective decisions are required to determine the threshold value, which affects detection effectiveness.

The information statistics mechanism could be more efficient due to its small feature set analysis and tendency, which results in false positive results. Scientists have used machine learning techniques such as support vector machines, decision trees, and random forests to detect DDoS attacks and identify alternatives [6]. These techniques do not help identify assaults in large-sample, high-dimensional data,

K. Radha (✉) · R. Parameswari
Departmentof Computer Science, School of Computing Sciences, VISTAS, Chennai, India
e-mail: radhavarshini2012@gmail.com

R. Parameswari
e-mail: dr.r.parameswari16@gmail.com

S. Nagini et al. (eds.), *Proceedings of Fifth International Conference on Advances in Computer Engineering and Communication Systems*, Smart Innovation, Systems and Technologies 117, https://doi.org/10.1007/978-981-96-4410-0_26

even though they perform better than information statistics-based techniques. They typically produce sufficient results when working with small-sample data and low-dimensional characteristics. Deep learning techniques have become a practical way to get around these restrictions. Deep learning techniques may learn complicated characteristics and turn high-dimensional data into abstract representations. Examples of these techniques are convolutional neural networks (CNNs), recurrent neural networks (RNNs), and graph neural networks (GNNs). By overcoming the drawbacks of conventional machine learning techniques, these methods provide promising gains in accurately identifying DDoS attacks.

This capability enables deep learning-based methods to handle high-dimensional and large-sample data with speed effectively. However, existing deep learning-based anomaly detection techniques primarily rely on a single model, which may need help to accurately and promptly identify abnormal traffic based on its features in real time. Regarding DDOS attack mitigation methodologies, the current focus is on attack identification rather than implementing effective reduction techniques. Some researchers have proposed transferring extra anomalous traffic to other controllers for handling [7]. This approach maximizes the processing time, which can harm DDoS attack defence even while relieving the stress on controllers in different domains. Other researchers categorize regular traffic into a self-built allow list [8]. They then remove any abnormal traffic from the database that does not conform to this allowlist. However, neither of these mitigation techniques considers the source of the attack or specifically targets eliminating abnormal traffic at its source.

26.2 Related Work

DDOS attacks are a significant threat to network security. Timely detection and mitigation measures are necessary to safeguard network infrastructures from the destructive power of DDOS attacks. Maninder et al. and Eliyan et al. (2021) proposed a classification system for DDOS detection methods in the SDN context. They have categorized these methods into two main groups: statistics-based and machine learning-based (Singh and Bhandari, 2020; Eliyan and Di Pietro, 2021) [9, 10]. These approaches include packet-based, flow-based, and packet-in-flow methods to examine network traffic.

Numerous experts and researchers have recently introduced various detection methods to combat DDOS attacks in SDN. Among these methods, statistical analysis-based anomaly detection has emerged as one of the most common approaches. Statistical analysis and detection methods benefit from regular network traffic exhibiting specific statistical patterns or laws across certain characteristics [11, 12]. By utilizing these statistical laws, these methods effectively differentiate traffic that deviates from these patterns and classify it as potential attack traffic. Some commonly used methods include information entropy, principal component analysis, and cardinality statistics. Mousavi et al. [13] proposed an intrusion detection system involving the calculation of entropy values to assess network traffic and identify any abnormalities indicative of

DDOS attacks, designed to identify DDOS attacks within an SDN controller. Salaria et al. [14] introduced an upgraded principal component analysis (PCA) technique for identifying abnormal traffic in distinct classified regions. Their experiments achieved a detection accuracy of 95.24%, which was 2.94% better compared to the enhanced method [15]. However, one limitation of the statistical analysis method is its reliance on a single fixed threshold. This characteristic makes it a false positive when dealing with DDOS attacks. Furthermore, the threshold for statistical analysis methods can vary depending on the environment. Threshold modification requires extensive expertise. Otherwise, it will directly impact the detection's accuracy. Considering these factors, it becomes evident that relying solely on statistical analysis methods may not be reliable for accurately judging abnormal traffic in real-world network scenarios. Despite this, flow sampling techniques mainly result in a high false-positive rate. Examining the packet-in message is another suggestion. For instance, in existing works they gathered three different forms of entropy from the packet-in data and used confidence intervals to identify attacks in real time. However, packet-in messages only contain information on new-arrival packets, which is insufficient to describe the state of the entire traffic. Overall, this research aims to introduce a methodology that effectively monitors network performance, identifies misbehaving nodes, and guarantees the reliable transmission of packets in a randomly deployed network with varying packet sizes.

26.3 Proposed Methodology

In our proposed architecture, as depicted in Fig. 26.1, a network of users (nodes) is connected to switches and routers. We establish a local server, $\boldsymbol{S_s(m)}$, within this setup on the source side. This server is responsible for monitoring the performance of network links and node behaviors, and it is equipped with high bandwidth allocation and storage capacity, denoted as $\boldsymbol{D_s(m)}$. The local server collects and analyzes comprehensive network information, which the SDN controller subsequently updates The SDN controller is a central repository, storing packet information and network data for further analysis and management. SDN, or Software-Defined Networking, is a highly effective architectural approach that enables network operators to manage their networks uniformly and comprehensively, regardless of the underlying network technology. By leveraging software applications and APIs (Application Programming Interfaces), SDN allows network behaviors to be programmed and centrally managed. Integrating a control layer with traditional network platforms empowers operators to operate and control their entire network efficiently, accommodating standard and complex network technologies.

Algorithm

1. Algorithm Local monitoring base attack detection (LM-AD)
2. Begin
3. {

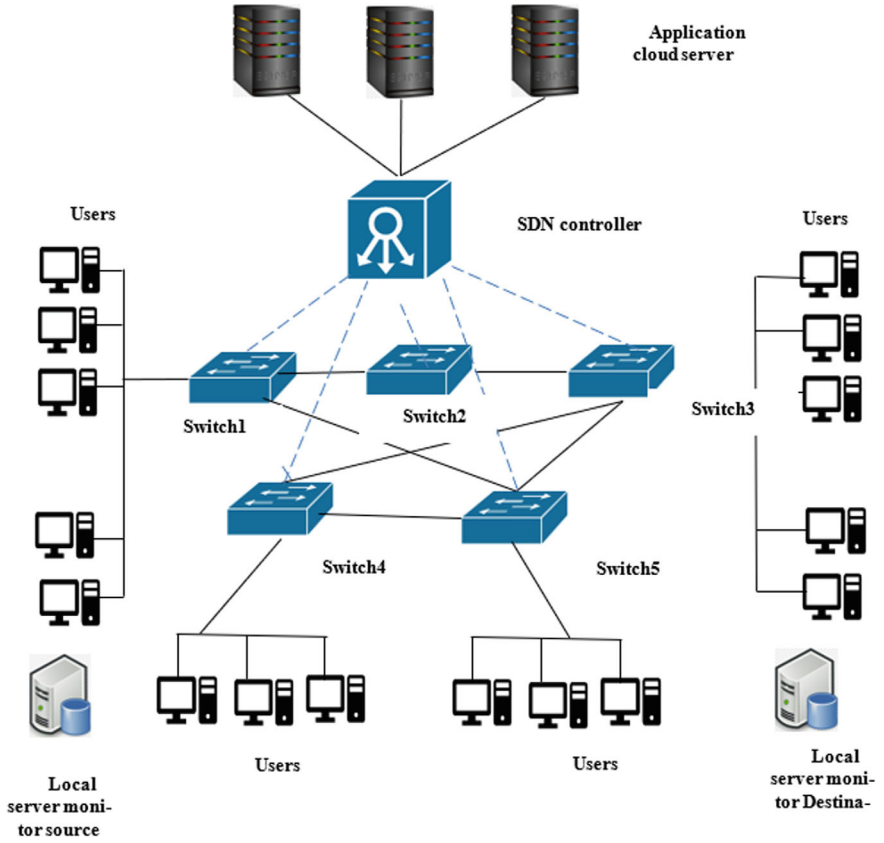

Fig. 26.1 Proposed architecture

4. Create $S_s(m), D_s(m)$ // source and destination local server with high storage and Bw
5. Allocate high bandwidth and storage;
6. To register all nodes and routers $S_s(m), D_s(m)$;
7. Number of nodes created (N1,N2,N3….Nn);
8. $S_s(m), D_s(m) \rightarrow high(bw)$; $(storage) = 5Tbramsize32to64gb$;
9. $(bandwidth) = \sum_{i=0}^{N} bw\left(\frac{D_t}{T_t}\right)$ // D_t Data transferred and T_t total time;
10. Packet transmission starts ();
11. {
12. $P_{size} \rightarrow (DataSize + HeaderSize + TrailerSize)$

    Case A
13. $D_s(m) \rightarrow registerallnodesbysenderside$;
14. Route ID: A, Nodes ID N1,N3,N8,13,N18,N20;
15. Check the routes and packet flow;

16. If (good reach destination) else
17. Check the nods behaviors if any node misbehaves remove or hold the Network;
18. Check the $D_s(m) \rightarrow SDN(storeapplicationserver)$; // check the nodes and path link updated to the SDN to cloud storage (application)

Case B

$$D_s(m) \rightarrow registerallnodesbysenderside;$$

19. Route ID: B, Nodes ID N1,N4,N7,N12,N14,N20;
20. If (the packet is good) else packet is;
21. $P_{size} \leq 30\%$ the traffic is high;
22. Check the path status and packet flow node 12 is misbehave; // may be remove and hold the node;
23. Check the $D_s(m) \rightarrow SDN(storeapplicationserver)$; // check the nodes and path link updated to the SDN to cloud storage (application);

Case C

24. $D_s(m) \rightarrow registerallnodesbysenderside$;
25. Route ID: B, Nodes ID N1,N4,N5,N11,N15,N19,N20
26. If (the packet is good) else packet is;
27. $P_{size} \leq 30\%$ the traffic is high;
28. Check the path status and packet flow node 19 is misbehave; // may be remove and hold the node;
29. Check the $D_s(m) \rightarrow SDN(storeapplicationserver)$; // check the nodes and path link updated to the SDN to cloud storage (application);

Case D

30. $D_s(m) \rightarrow registerallnodesbysenderside$;
31. Route ID: A Node ID N1,N24,N6,N9,N16,N20;
32. Check the routes and packet flow;
33. If (good reach destination) else
34. Check the nods behaviors if any node misbehaves remove or hold the Network;
35. Check the $D_s(m) \rightarrow SDN(storeapplicationserver)$; // check the nodes and path link updated to the SDN to cloud storage (application)
36. }

Figure 26.2 shows the number of nodes (N1–N21). The nodes were randomly deployed, and each node has a different packet size and is not static. In this proposed methodology, the source-side local monitor server and destination-side local monitor server are created with high bandwidth and storage, as shown in algorithm 1. These two local monitor servers store the entire network performance and update the SDN to the application server (cloud storage). This Fig. 26.2 shows the number of routes for routes a, b, c, and d. The packet transmission process starts with many packets flowing in different directions. This checks and monitors the network's performance and node behaviors. In routing Table 26.1, we clearly explain the network performance and misbehaving node structure. and packet behaviors of N12 behave abnormally;

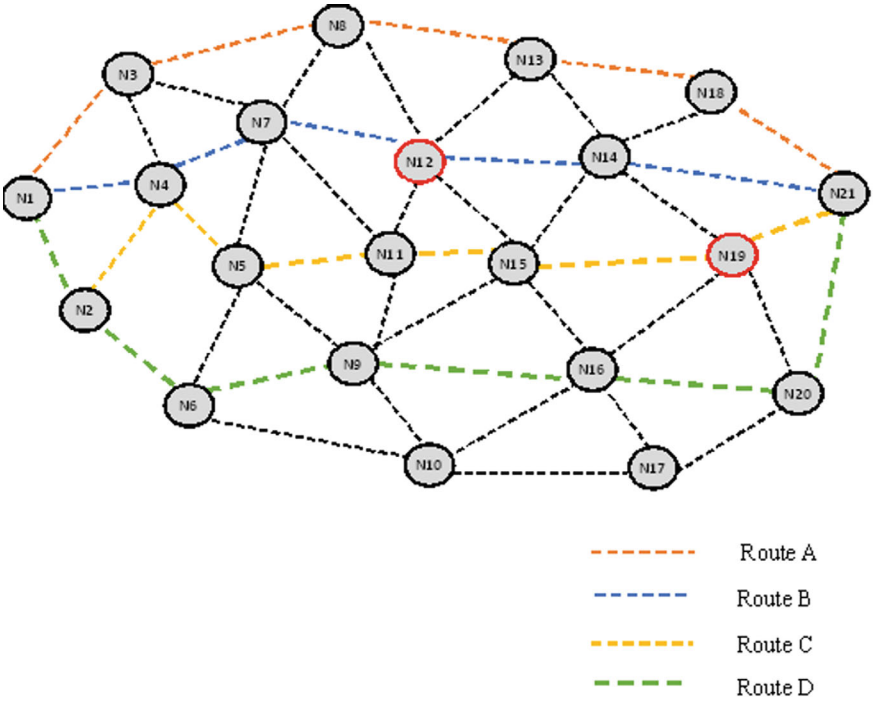

Fig. 26.2 Node connection

Table 26.1 Routing table

Route ID	Node no	Packet flow	Path weight	Misbehaving node
A	N1, N3, N8, N13, N18, N20	Good	Good	Nil
B	N1, N4, N7, N17, N14, N20	Up normal	Average	N12
C	N1, N4, N5, N11, N15, N19, N20	Up normal	Average	N19
D	N1, N24, N6, N9, N16, N20	Good	Good	Nil

checking the packet size, which is less than normal size, is charted in algorithm 1, steps 12 and 23. N12 is removed from the network or holds N12. In the case of C, path C has N1, N4, N5, N11, N15, N19, and N20. Here N19 is abnormal, checks if the packet size is less than the normal size. As shown in algorithm 1, steps 12 and 30, N19 is removed from the network or held. So in Case D, path D has N1, N24, N6, N9, N16, and N20 By checking the nodes and packets activity, this process proceeds after some time when there is no normal activity. It reaches the destination where the current packet transmission has no error, checks, and updates the SDN to cloud storage.

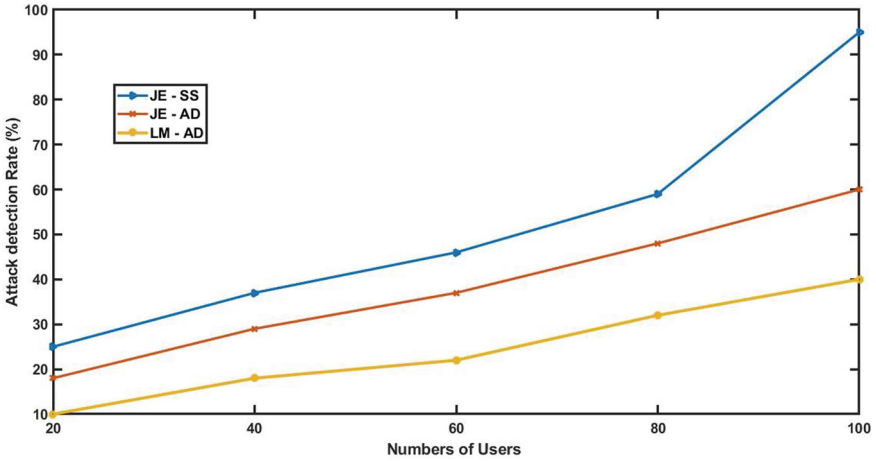

Fig. 26.3 Number of users versus Attack detection

In the case of A, path A has N1, N3, N8, N13, N18, and N20 nodes and was registered by LM-AD By checking the nodes and packets, this proceeds after some time when there is no any up normal activity, reaches the destination where the current packets or routing has no error N1.

Figure 26.3 shows a graphical representation of the number of users versus attack detection performance of the proposed LM-AD algorithm compared to existing algorithms, namely JE-SS and JE-AD. The graph shows the attack detection rate achieved by each algorithm as a function of the number of users involved in the experiment. The x-axis shows the number of users, while the y-axis shows the attack detection percentage. The attack detection gained by LM-AD for 20 users is 10%, for 40 users it is 18%, for 60 users it is 20%, for 80 users it is 30%, and for 100 users it is 40%. The ADR achieved by JE-SS and JE-AD is also illustrated. The results clearly indicate that the proposed LM-AD algorithm outperforms the other algorithms.

Figure 26.4 shows graphical representation of number of users versus packet loss ratio performance of the proposed LM-AD (Local monitoring-based attack detection) algorithm compared to existing algorithms, namely JE-SS and JE-AD. The graph shows the packet loss ratio achieved by each algorithm as a function of the number of users involved in the experiment. The x-axis shows the number of users, while the y-axis shows the packet loss ratio in percentage. The packet loss ratio gained by LM-AD for 20 users is 10%, for 40 users is 20%, for 60 users is 40%, for 80 users is 50%, and for 100 users is 60%. The packet loss ratio achieved by JE-SS (joint entropy-based security scheme) and JE-AD (joint entropy based attack detection scheme) are also illustrated. The results clearly indicate that the proposed LM-AD algorithm outperforms the other algorithms.

Figure 26.5 shows graphical representation of the number of users versus packet delivery ratio performance of the proposed LM-AD algorithm compared to existing algorithms, namely JE-SS and JE-AD. The graph shows the packet delivery ratio

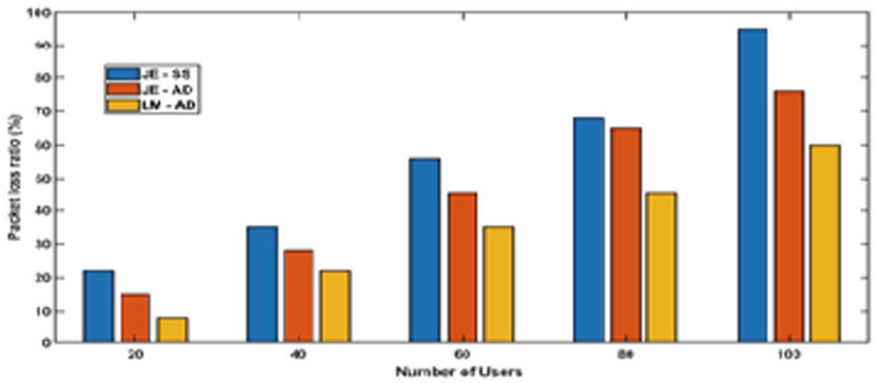

Fig. 26.4 Number of users versus packet loss ratio

achieved by each algorithm as a function of the number of users involved in the experiment. The x-axis shows the number of users, while the y-axis shows the packet delivery ratio in percentage. The packet delivery ratio gained by LM-AD for 20 users is 25%, for 40 users is 30%, for 60 users is 50%, for 80 users is 70%, and for 100 users is 95%. The packet delivery ratio achieved by JE-SS and JE-AD are also illustrated. The results clearly indicate that the proposed LM-AD algorithm outperforms the other algorithms.

Figure 26.6 shows graphical representation of the number of users versus system efficiency performance of the proposed LM-AD algorithm compared to existing algorithms, namely JE-SS and JE- AD. The graph shows the system efficiency achieved by each algorithm as a function of the number of users involved in the experiment. The x-axis shows the number of users, while the y-axis shows the system efficiency

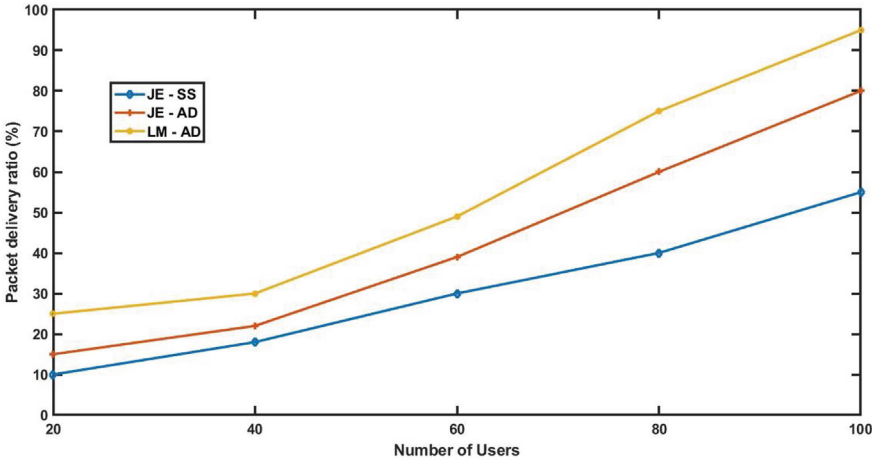

Fig. 26.5 Number of users versus Packet delivery ratio

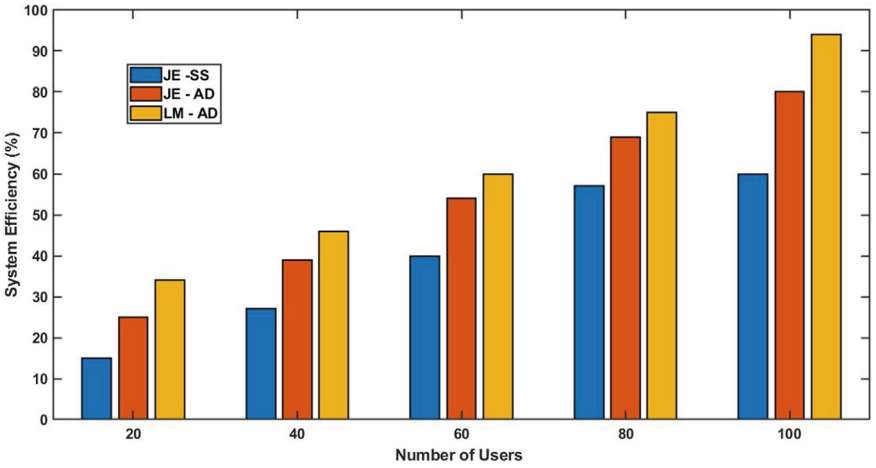

Fig. 26.6 Number of users versus system efficiency

in percentage. The system efficiency gained by LM-AD for 20 users is 35%, for 40 users is 45%, for 60 users is 60%, for 80 users is 75%, and for 100 users is 95%. The system efficiency achieved by JE-SS and JE- AD are also illustrated. The results clearly indicate that the proposed LM-AD algorithm outperforms the other algorithms.

26.4 Result and Disscussion

Achieving communication reliability and optimal DDoS attack detection is the main goal of this suggested LM-AD in dynamic systems. Nodes from Lowest packet loss ratio is very dangerous, this was analyzed using the network operations management. The suggested methodology offers a comprehensive structure for monitoring network performance, detecting intrusions, ensuring compatibility within the network, and establishing a strong foundation.

26.5 Conclusion

This paper proposes the Local Monitoring Attack Detection (LM-AD) method. This method is designed to effectively detect attacks and monitor overall performance even in random networks. LM-AD primary goal is to attain secure and reliable communication using local monitor servers, SDN, and cloud storage. The experimental work determines the LM-AD performance, and it is very effective in delivering the

packs, detecting the attacks and minimizing the packet loss during packet transmissions. To maintain network integrity, misbehaving nodes are identified immediately and handled appropriately. As a result, LM-AD offers a robust and efficient way to monitor network performance and ensure secure communication. Future research can focus on optimizing and implementing LM-AD in real-world networks to validate its effectiveness and applicability.

References

1. S. Bhatia, S. Behal, Distributed denial of service attacks and defense mechanisms: current landscape and future directions, in *Versatile Cybersecurity*; Springer: Berlin/Heidelberg, Germany, Vol. 72, pp. 55–97 (2018)
2. T.V. Phan, M. Park, Efficient distributed denial-of-service attack defense in SDN-based cloud. IEEE Access **7**, 18701–18714 (2019)
3. RADWARE, 2017–2018 Global Application & Network Security Report; RADWARE: Tel Aviv-Yafo, Israel (2018)
4. J. Singh, S. Behal, Detection and mitigation of DDOS attacks in SDN: a comprehensive review, research challenges and future directions. Comput. Sci. Rev. **37**, 100279 (2020)
5. R. Wang, Z.P. Jia, L. Ju, An entropy-based distributed DDOS detection mechanism in software-defined networking, in *Proceedings of the 2015 IEEE Trustcom/BigDataSE/ISPA*, Washington, DC, USA, 20–22 (2015)
6. R. Santos, D. Souza, W. Santo, Machine learning algorithm to detect DDOS attacks in SDN. Concurr. Comput. Pract. Exp. **32**, 16 (2020)
7. P. Wu, L. Yao, C. Lin, FMD: A DoS mitigation scheme based on flow migration in software-defined networking. Int. J. Commun. Syst. **31**, e3543 (2018)
8. Y. Cao, H. Jiang, Y. Deng, Detecting and mitigating DDOS attacks in SDN using spatial-temporal graph convolutional network. IEEE Trans. Dependable Secur. Comput. **19**, 1–8 (2022)
9. M.P. Singh, A. Bhandari, New-fow based DDOS attacks in SDN: taxonomy, rationales, and research challenges. ComputCommun **154**, 509–527 (2020)
10. L.F. Eliyan, R. Di Pietro, DoS and DDOS attacks in software defned networks: a survey of existing solutions and research challenges. Futur. Gener. Comput. Syst. **122**, 149–171 (2021)
11. M.E. Ahmed, S. Ullah, Statistical application fingerprinting for DDOS attack mitigation. IEEE Trans. Inf. Forensics Secur. **14**, 1471–1484 (2019)
12. R.N. Carvalho, J.L. Bordim, E.A.P. Alchieri, Entropy-based DoS attack identification in SDN, in *Proceedings of the IEEE International Parallel and Distributed Processing Symposium Workshops*, Rio de Janeiro, Brazil, 20–24 May 2019
13. S.M. Mousavi, M. St-Hilaire, Early detection of DDOS attacks against SDN controllers, in *Proceedings of the International Conference on Computing, Networking and Communications*, Anaheim, CA, USA, 16–19 February 2015
14. S. Salaria, S. Arora, N. Goyal, Implementation and analysis of an improved PCA technique for DDOS detection, in *Proceedings of the IEEE 5th International Conference on Computing Communication and Automation*, Greater Noida, India, 30–31 October 2020
15. N.-N. Dao, J. Park, M. Park, S. Cho, A feasible method to combat against DDOS attack in SDN network, in *2015 International Conference on Information Networking (ICOIN)*. IEEE, pp. 3 (2015)