Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.033600
Article

RMCARTAM For DDoS Attack Mitigation in SDN Using Machine Learning

M. Revathi, V. V. Ramalingam* and B. Amutha

Department of Computing Technologies, SRM Institute of Science and Technology, Kattankulathur, Chennai, 603203, Tamil Nadu, India
*Corresponding Author: V. V. Ramalingam. Email: vramalin@outlook.com
Received: 21 June 2022; Accepted: 22 August 2022

Abstract: The impact of a Distributed Denial of Service (DDoS) attack on Software Defined Networks (SDN) is briefly analyzed. Many approaches to detecting DDoS attacks exist, varying on the feature being considered and the method used. Still, the methods have a deficiency in the performance of detecting DDoS attacks and mitigating them. To improve the performance of SDN, an efficient Real-time Multi-Constrained Adaptive Replication and Traffic Approximation Model (RMCARTAM) is sketched in this article. The RMCARTAM considers different parameters or constraints in running different controllers responsible for handling incoming packets. The model is designed with multiple controllers to handle network traffic but can turn the controllers according to requirements. The multi-constraint adaptive replication model monitors different features of network traffic like rate of packet reception, class-based packet reception and target-specific reception. According to these features, the method estimates the Replication Turning Weight (RTW) based on which triggering controllers are performed. Similarly, the method applies Traffic Approximation (TA) in the detection of DDoS attacks. The detection of a DDoS attack is performed by approximating the incoming traffic to any service and using various features like hop count, payload, service frequency, and malformed frequency to compute various support measures on bandwidth access, data support, frequency support, malformed support, route support, and so on. Using all these support measures, the method computes the value of legitimate weight to conclude the behavior of any source in identifying the malicious node. Identified node details are used in the mitigation of DDoS attacks. The method stimulates the network performance by reducing the power factor by switching the controller according to different factors, which also reduces the cost. In the same way, the proposed model improves the accuracy of detecting DDoS attacks by estimating the features of incoming traffic in different corners.

Keywords: DDoS; SDN; traffic approximation; adaptive replication; multi-controller; support measures; RTW

1 Introduction

A Software-Defined Network (SDN) is a new network in the network technology era that paves the way for the deployment of Fifth Generation (5G) networks. As the networks are deployed to provide various

support and seamless services to the consumers and users of the network, their existing physical characteristics are not supportive of achieving the expected seamless and quality data transmission. Previously, the network was only used to transmit textual and numerical data. But the development of the environment and network has adapted the transmission of multimedia content on the network. The engagement of multimedia data in the network raises the requirement for seamless and rapid data transmission. For example, when the network has been used in video conferencing, the model should provide seamless transmission as well as higher quality. But the network nodes work together to send data packets to their final destination by sending them through many intermediate nodes.

To provide seamless and rapid transmission of data, a 5G network has emerged, which is a collection of some Multiple Input and Multiple Output (MIMO) devices, the number of small units which combine the number of MIMO in a single unit, and so on. The reason for the adaption of the 5G network is to provide high-quality data transfer to the users. Mobile users have recently accessed various network services and video files through their mobile phones. However, the users access a set of services available on a specific node in the network. There is massive data transfer and traffic in the network, and to control such traffic, there are controllers engaged which receive the traffic and divert or route it towards the destination through different nodes and devices. Such processes are done by a set of software components named SDN. Routing is the most critical task in SDN, which receives the incoming traffic and diverts the traffic through several routes according to the protocol engaged. Several routing protocols exist that route the traffic according to features like hop count, latency, congestion, energy, throughput, etc. But because there is so much traffic at the service point, it will take a more strategic approach to find the impaired node.

The presence of a malicious node in the SDN generates various threats to the service point. However, the malicious node generates different threats to the service point; the effect of a DDoS attack is greater and affects the entire network performance. Any service can handle many requests, but the malicious node, in turn, would obtain a large number of connections and hold them idle while they don't perform any data transmission. In another case, the malicious node would send many malicious packets. Both these attacks would degrade the service performance as well as the network. The presence of malicious or DDoS attacks can be identified using several features, and there are many methods available in the literature, but they suffer from achieving the expected performance.

The sample SDN topology with the Internet of Things (IoT) devices considered for the problem is presented in Fig. 1. The malicious node would generate different packets and threats to degrade the service performance. In SDN, there will be a controller who monitors the incoming traffic. But in the presence of large users and when there are enormous packets received at the controller, it would blindly drop and allow malicious packets into the system. So, this struck the service performance and the network performance. To override this, an adaptive controller replication model is presented in this article. The method can replicate the controller to handle incoming traffic when there is enormous traffic. Similarly, the model can replicate the controller based on various features. Also, the issue of DDoS attacks has been handled by performing TA, which considers the features like payload, hop count, latency, class-based reception, etc. The features considered are used in measuring different support measures for various features. According to various constraints and features, the DDoS attack has been detected and mitigated.

2 Related Works

OpenFlow vulnerabilities in SDN have been analyzed to detect DDoS attacks in [1], which monitors the vulnerabilities in controllers and switches to perform the attack detection. A cooperative scheme on Multiple SDN for DDOS attack detection is presented in [2], which applies Machine Learning (ML) to make the controllers share information about attacks to perform mitigation. To protect SDN from DDoS attacks, an

extreme gradient boosting with a bandwidth control approach is presented in [3], which has been evaluated using the Cooperative Association for Internet Data Analysis (CAIDA) data set. An entropy-based DDoS detection approach is presented in [4], which measures the difference between normal and abnormal traffic to compute the entropy value to perform the detection. Similarly, an integrated model is presented in [5], which uses SDN features and produces higher accuracy and lower cost.

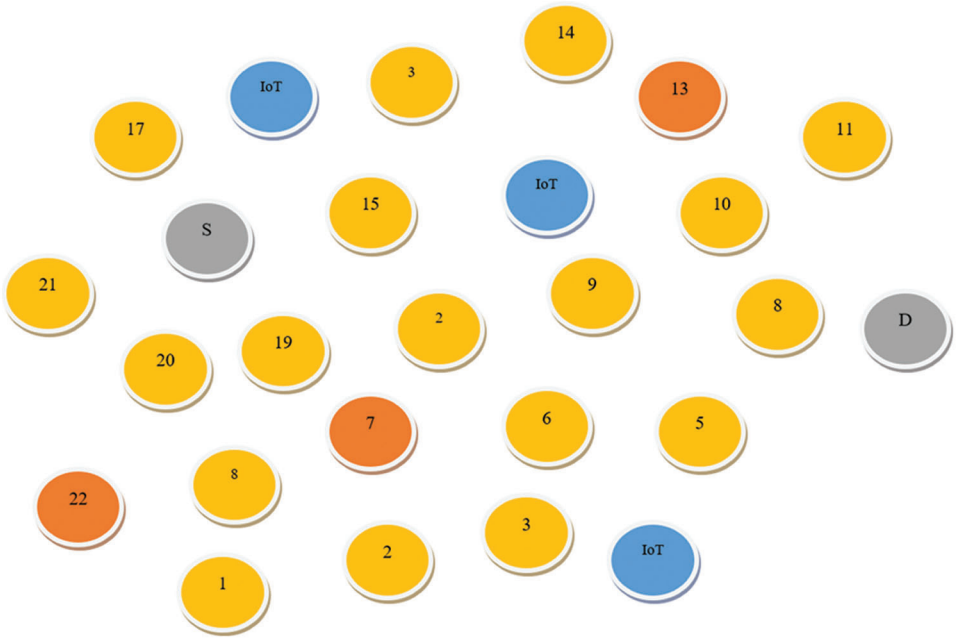

Figure 1: Sample network topology

A decision tree-based low latency system for DDoS attack detection DEcision Tree Pro (DETPro) is presented in [6], where the Pox controller and agents collect network traffic information. Based on the information collected, decision trees are used for detection. Similarly, a Random Forest (RF) and K means+ based approach is presented in [7], where the ensemble learning algorithm is used in classification. An entropy with a deep learning algorithm is presented in [8], which measures the entropy value on the traffic features and, based on the value, the classification is performed. A Joint Entropy-based Security Scheme (JEES) is presented in [9], which estimates the joint entropy in detecting DDoS attacks.

A volume-specific application-oriented DDoS detection approach is presented in [10,11], which classifies Transmission Control Protocol (TCP), Ping, and Hypertext Transfer Protocol (HTTP) packets to perform the detection with mininet and Pox controller. An OpenFlow statistics-based approach estimates the packet rate and bandwidth constraints in detecting DDoS attacks. A factorization machine-based low rate attack detection model is presented in [12], which is a multi-feature model and uses flow rules in the detection. A comparative study in the detection of DDoS attacks is presented in [13], where an entropy-based improved model is presented in [14], which uses Bidirectional Long Short-Term Memory-Recurrent Neural Networks (BiLSTM-RNN) to analyze the traffic and to perform classification.

An ML-based intelligent model is presented in [15], which uses the length of service and rules in the classification [16]. A network-oriented defensive approach for SDN is presented in [17], which detects HTTP attacks. An entropy-based Particle Swarm Optimization-Back-Propagation (PSO-BP) has been presented in [18], which uses the characteristics of SDN and entropy values. A time-oriented DDoS

detection is presented in [19], which uses a single-point controller. Similarly, to support third-party applications in SDN, a traffic monitoring scheme is presented, which views the traffic at a specific time. Based on the traffic and bandwidth of incoming packets, the method handles the detection. A low-rate DDoS attack detection model in SDN is presented in [20,21], which uses different ML models to evaluate.

A real-time mitigation approach toward DDoS attacks is presented in [22], which uses the Flow mitigation technique by analyzing the network traffic and rules. A Convolution Neural Network with Long Short-Term Memory (CNN-LSTM) based approach is presented in [23] to support SDN networks. An RF algorithm-based early detection scheme is presented in [24], which uses flow features and rules to support the detection of DDoS attacks. An SDN feature-based approach is presented in [25] to detect an attack, and a ML classifier is presented in [26], which uses polynomial SVM in classification.

An offline training and online recognition algorithm to detect the threat in Conditional Generative Adversarial Network (CGAN) [27] uses various detection features. A k-Nearest Neighbours (KNN) based model is presented for improving threat detection by incorporating ML techniques. An early detection approach is presented, which uses timeout and idle flow values to identify the presence of a threat [28–30]. We show an HTTP-based threat detection model that uses the number of requests and a threshold to find threats.

3 The Proposed Real-Time-Multi Constrained Adaptive Replication and Traffic Approximation Model

The proposed RMCARTAM monitors incoming network traffic and finds traffic in different network services. Accordingly, to that, the method estimates the Replication Turning Weight (RTW) to decide on the process of replicating the controllers. Also, the method monitors the traffic and identifies various features to perform DDoS attack detection by computing different support measures on various features. The detailed approach is presented in this section. In this section, we talked in detail about Fig. 2's functional architecture of the proposed RMCARTAM and the method's functional stages.

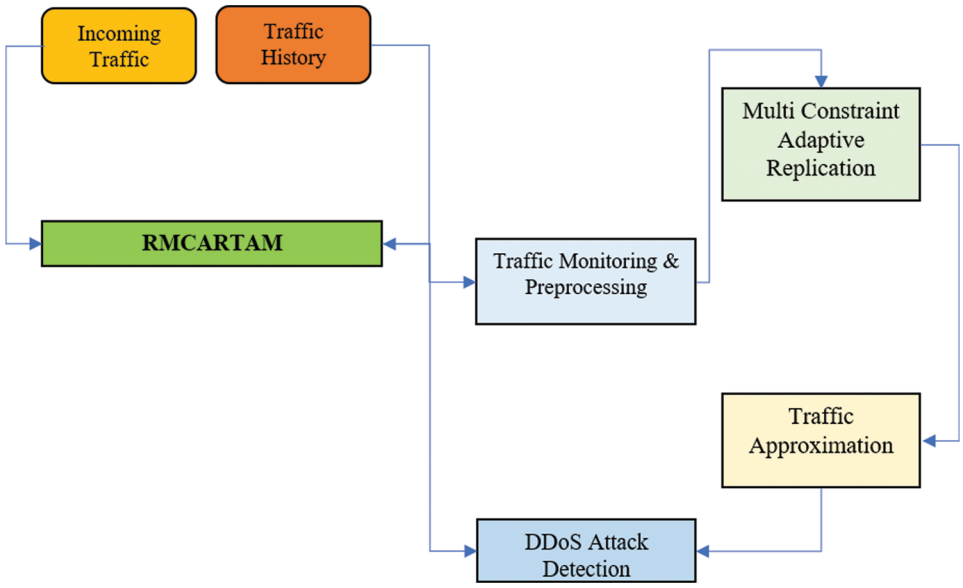

Figure 2: Architecture of proposed RMCARTAM

3.1 Traffic Monitoring and Preprocessing

The proposed model monitors the incoming traffic to the controller and is deployed. From the traffic received, the method extracts features like rate of packet reception, class-based packet reception, and target-specific reception. The packet reception rate represents the number of packets received at any given time. The class-based packet reception represents the type of packet and the rate at which it is received. For example, Internet Control Message Protocol (ICMP) packets are received at a constant rate at any timestamp; similarly, the packets are classified as TCP, User Datagram Protocol (UDP), HTTP, File Transfer Protocol (FTP), and so on. In each class, the rate of packet reception is measured. Such features extracted are forwarded to the adaptive replication model to perform replication. Consider that the model maintains the traffic trace T_{rt}, which has several pieces of information related to different sources from where the packets are received. So, whenever a packet has been received at a controller, it receives it and extracts the features mentioned earlier. Once these features are taken out, the method uses Eqs. (1) and (2) to estimate the number of packets that were received at any given time stamp T_s:

$$\text{Tpr}\ (\text{Ts}) \ = \ \sum \textit{Packets Received at Time}\ (\textit{TS}) \tag{1}$$

$$\text{CbPr}\ (\text{Ts}) \ = \ \sum_{\text{i=1}}^{\text{Size(Tpr)}} \text{Tpr.Type} == \text{Class} \tag{2}$$

Similarly, the service-based packet reception is measured as follows, Eq. (3):

$$\text{SbPr} \ = \ \sum_{\text{i=1}}^{\text{size(Tpr)}} \text{Tpr(i).Service} == \text{S} \tag{3}$$

Step 1. Given: Traffic Trace TrT

Step 2. Obtain Class Set Cs, Service Set SeS, Tpr

Step 3. Start

Step 4. While true

Step 5. Receive packet p

Step 6. Fetch Source Address Saddr = *Saddr* $\in$

Step 7. Fetch Source Port Sport = $Source_{port} \in P$

Step 8. Fetch Type of packet PType = *Ptype* $\in P$

Step 9. Fetch Service Type SType = *Stype* $\in$

Step 10. Compute Total packet reception

Step 11. For Each class C

Step 12. Compute CbPr

Step 13. Compute SbPr

Step 14. Add Cbpr to Cs = $\sum(\text{Cbpr} \in \text{Cs}) \cup \text{CbPr}$

Step 15. Add SbPr to Ses = $\sum(\text{Sbpr} \in \text{Cs}) \cup \text{SbPr}$

Step 16. End

Step 17. Stop

3.2 Multi Constraint Adaptive Replication

The proposed multi-constraint adaptive replication model works based on the various constraints considered. To perform this, the features extracted in the previous stage are utilized. The method continuously monitors the conditions of the environment within a time stamp. At each time stamp, the method applies the adaptive replication process to stabilize the workload as well as to perform DDoS attack detection. To handle the replication, the method first receives the rate of packet reception, class-based packet reception, and target-specific reception sets generated in the preprocessing.

Step 1. Class Set Cs, Service Set SeS, T_{pr}, Controller Set Cons, Malicious Trace (MT)

Step 2. Obtain Cons

Step 3. Start

Step 4. Read class set Cs, Service Set S_{es}, and Controller set Cons

Step 5. For Each controller running Cr

Step 6. For Each service S

Step 7. For Each class C

Step 8. Compute $\text{RTW} = \frac{\text{Cs(C).Cbpr} \times \text{Ses(S).Sbpr}}{\text{Tpr}} \times \frac{\sum_{i=1}^{\text{size(MT)}} \text{MT(i).Service} == \text{S} \&\& \textit{Controller} == \textit{Cr}}{\text{size(MT)}}$

Step 9. End

Step 10. Compute cumulative $\text{CRTW} = \frac{\sum \text{RTW}}{\text{Size(Cs)}}$

Step 11. If CRTW>Th Then

Step 12. Initialize new controller Nc

Step 13. Replicate the controller on new port

Step 14. Add to controller set Cons = $\sum(\text{Controllers} \in \text{Cons}) \cup \text{NC}$

Step 15. End

Step 16. End

Step 17. End

Step 18. Stop

The adaptive replication algorithm shows how the controller replication is performed and on what basis the controller is deployed. The method deploys a new controller for a dedicated service only based on CRTW when it is greater than a threshold that represents the occurrence of a higher attack and the occurrence of higher traffic in a specific controller. To make the controller work dedicated to the service, it deploys a new controller to handle the service packets.

3.3 Traffic Approximation

The proposed model performs DDoS attacks by approximating the traffic present in the network which is received at any of the controllers. It has been performed independently by all the controllers deployed. The controller receives the incoming traffic, and at each packet received, the following features are extracted: hop count, payload, Time To Live (TTL) value, and joint hop. Using these features, the method computes service frequency and malformed frequency. Further, the method estimates support on bandwidth access, data support, access frequency support, malformed support, route support, and so on. The TA algorithm uses the access trace and malicious trace to compute various support measures on data, route, time, malformed access frequency support, and access frequency support. All these values are given to the model to perform attack detection.

Step 1. MT, Access Trace (AT), Packet P

Step 2. BS, DS, AFS, MS, RS, TS

Step 3. Start

Step 4. Read MT and AT, P

Step 5. Source address S_{addr} = source_address $\in$ P

Step 6. Hop count Hp = $\sum$ Hops $\in$ P

Step 7. Payload Pl = $\sum$ Bytes $\in$ P

Step 8. Extract TTL = $TTL \in P$

Step 9. Compute service frequency $Sf_{req} = \frac{\sum_{i=1}^{Size(AT)} AT(i).Service == S \;\&\&\; AT(i).Host == Saddr}{size(AT)}$

Step 10. Compute malicious frequency MF_{req}

Step 11. $MF_{req} = \frac{\sum_{i=1}^{Size(AT)} AT(i).Service == S \;\&\&\; AT(i).Host == Saddr \;\&\&\; AT(i).Type == Malicious}{size(AT)}$

Step 12. Compute Bandwidth Support (BS) = $\frac{\sum_{i=1}^{size(AT)} Payload(AT(i)) \&\& AT(i).Host == Saddr}{\sum_{i=1}^{size(R)} Bandwidth(R(i))}$

Step 13. Compute Data Support (DS) = $\frac{Pl}{\sum_{i=1}^{size(AT)} Payload(AT(i)) \&\& AT(i).Host == Saddr}$

Step 14. Compute Malformed Support (MS) $\sum_{i=1}^{size(AT)} AT(i).Host == Saddr$

Step 15. $MS = \frac{\sum_{i=1}^{size(AT)} (AT(i).Service == S)) \&\& AT(i).Host == Saddr \;\&\&\; AT(i).Type == Malicious}{\sum_{i=1}^{size(AT)} AT(i).Host == Saddr}$

Step 16. Compute Route Support (RS) = $\frac{Hp}{\sum_{i=1}^{size(AT)} AT(i).HopCount/Size(AT)}$

Step 17. Compute Time support (TS) = $\frac{TTL}{\sum_{i=1}^{size(AT)} AT(i).TTL/Size(AT)}$

Step 18. Stop

3.4 Distributed Denial of Service Attack Detection and Mitigation

The method estimates the source's Transport Security Model (TSM) using all these support measures. Once the value of TSM is higher than a threshold, the packets are traced at malicious tracing, and the source has been blocked for traffic. The incoming traffic is monitored and preprocessed with the traffic features to perform adaptive replication according to various constraints.

Furthermore, the incoming packets are received, and their features are extracted to perform TA. The result of TA has been used to measure legitimate weight. Based on the value of the legitimate weight, the method performed DDoS attack detection and identified malicious packets that were dropped. The DDoS attack detection algorithm shows how the presence of a DDoS attack is performed. The method estimates the legitimate weight according to various support measures obtained by approximating the traffic features. According to the features, the method computes the value of the legitimate weight and, based on that, and the method performs DDoS attack detection.

Step 1. MT, AT, Malicious Node Set Ms

Step 2. Obtain: Null

Step 3. Start

Step 4. Read MT and AT

Step 5. While true

Step 6. Receive incoming packet P

Step 7. Features {Cset, Sset, Tpr} = Traffic Monitoring and Preprocessing (P)

Step 8. If Time is up, then

Step 9. Perform multi-constraint adaptive replication

Step 10. End

Step 11. [BS, DS, AFS, MS, RS, TS] = Perform TA

Step 12. Compute legitimate weight $Lw = \frac{MS}{AFS} \times \frac{BS \times DS}{RS} \times \frac{TS}{RS}$

Step 13. If $Lw < Th$

Step 14. Then DDOS attack

Step 15. Generate trace in MT

Step 16. Add to malicious set $Ms = \sum(Nodes \in Ms) \cup NodeId$

Step 17. Perform malicious geospatial clustering

Step 18. End

Step 19. End

Step 20. Stop

3.5 Geo-Spatial Malicious Clustering

The malicious nodes identified in the malicious node detection phase have been grouped according to geographic and spatial properties. The method reads the set of clusters available, and for the newly identified node, the method identifies the location details. With the location details, for each node in each cluster, the method computes the average distance value. Finally, the cluster with the most negligible distance value has been selected, and the malicious node has been added to the cluster. Once the clustering is performed, the method invokes adaptive replication, which decides on the implication of a dedicated controller for the clusters generated. The clustering approach represents how the geospatial clustering of malicious nodes is performed according to the distance measures. Also, the method replicates the controller accordingly.

Step 1. Cluster set Cs, Node N

Step 2. Obtain Cluster Set Cs

Step 3. Start

Step 4. Read CS, N

Step 5. Location l = Extract location of node N

Step 6. For each cluster C

Step 7. For each node, CN

Step 8. Compute distance CNdist = Dist(CN.location,N.location)

Step 9. End

Step 10. Compute mean distance $Amd = \frac{\sum_{i=1}^{size©} C(i).CNdist}{size©}$

Step 11. End

Step 12. C = Choose, the cluster with the least mean distance C

Step 13. If Ld > Th, then

Step 14. Initialize new cluster NC

Step 15. Add node to the cluster NC

Step 16. Add a cluster to the cluster set

Step 17. Else

Step 18. Add node to cluster C

Step 19. Perform adaptive controller replication

Step 20. End

Step 21. Stop

4 Results and Discussion

The proposed RMCARTAM has been implemented using the Mininet network simulator. The method has been evaluated for its performance using different stimulation parameters and values. The evaluation details used for the performance measurement of different methods are presented in Tab. 1. The proposed RMCARTAM has been measured for its performance under different conditions, such as in several nodes as 50, 100, and 200 node simulation conditions. The method's performance has been measured and compared with other methods in each test condition.

Table 1: Evaluation details

Parameter	Value
Tool used	Mininet
Number of nodes	200
Number of controllers	5
Number of services	10
Simulation time	10 min

4.1 Attack Detection Performance

The attack detection performance is measured based on the number of attacks produced and the number of them detected and stopped. This paper presented a RMCARTAM for the detection presented in Tab. 2. The proposed RMCARTAM performs better than JESS, BiLSTM, and DETPro in all the test conditions.

Table 2: Analysis of attack detection performance

Attack detection performance *vs*. no of nodes in %			
	50 Nodes	100 Nodes	200 Nodes
JESS	66	71	74
BiLSTM	71	76	77
DETPro	75	79	84
RMCARTAM	87	91	97

The performance in detecting the threat is measured in the presence of the different numbers of nodes in the network and plotted in Fig. 3. The proposed RMCARTAM method achieved higher performance in all the test cases than the JESS, BiLSTM, and DETPro methods.

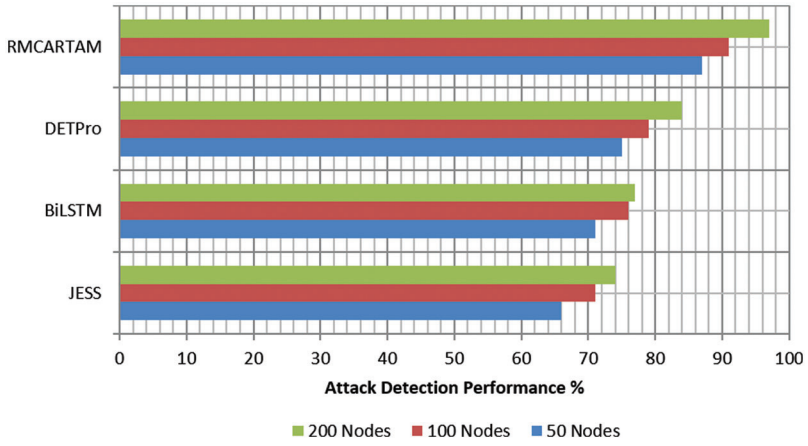

Figure 3: Analysis of attack detection

4.2 Throughput Performance

The throughput performance of the algorithm is measured based on the amount of data generated by the genuine source node and the number of bytes of data delivered successfully (Eq. (4)). The throughput achievement performance in the number of nodes condition is measured and presented in Tab. 3. In all of the test conditions, the RMCARTAM did better than the other JESS, BiLSTM, and DETPro methods.

Table 3: Analysis of throughput performance

Models	50 Nodes	100 Nodes	200 Nodes
JESS	65	68	72
BiLSTM	68	73	76
DETPro	71	78	82
RMCARTAM	86	90	96

$$\text{Throughput} = \frac{\text{Total Bytes Delivered}}{\text{Total Bytes Generated}} \times 100 \tag{4}$$

The analysis of throughput performance is measured and presented in Fig. 4, which shows that the proposed RMCARTAM method produced higher throughput performance than the JESS, BiLSTM, and DETPro methods.

4.3 Packet Delivery Ratio

The ratio of the packet delivered by different approaches is measured according to Eq. (5). The packet delivery ratios produced by various approaches are measured and presented in Tab. 4, where the RMCARTAM method produces better performance than JESS, BiLSTM, and DETPro methods.

$$\text{PDR} = \frac{\text{Number of Packets Delivered}}{\text{Total Packets Sent}} \times 100 \quad (5)$$

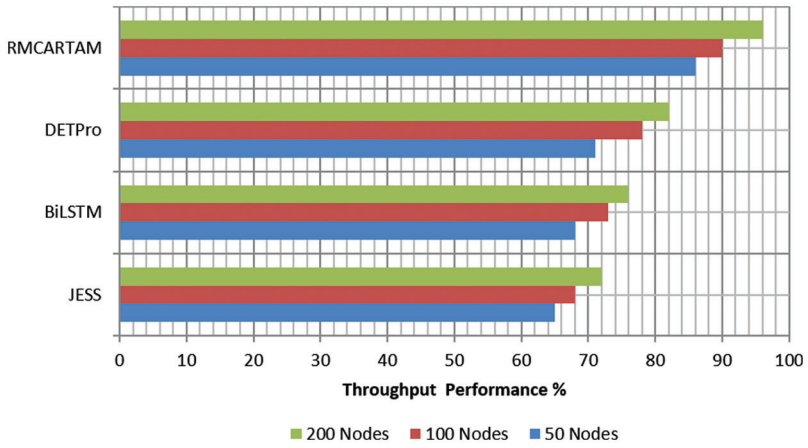

Figure 4: Performance in throughput

Table 4: Analysis of packet delivery ratio

Models	50 Nodes	100 Nodes	200 Nodes
JESS	63	65	72
BiLSTM	66	69	75
DETPro	69	74	79
RMCARTAM	81	85	96

The Packet Drop Ratio (PDR) produced by different methods is measured by the number of node conditions, like 50, 100, and 200. In each simulation condition, the PDR produced by different methods is measured and compared in Fig. 5. However, the proposed STABD scheme has introduced 18%, 14%, and 4% PDR, which is less than the existing JESS, BiLSTM, and DETPro algorithms.

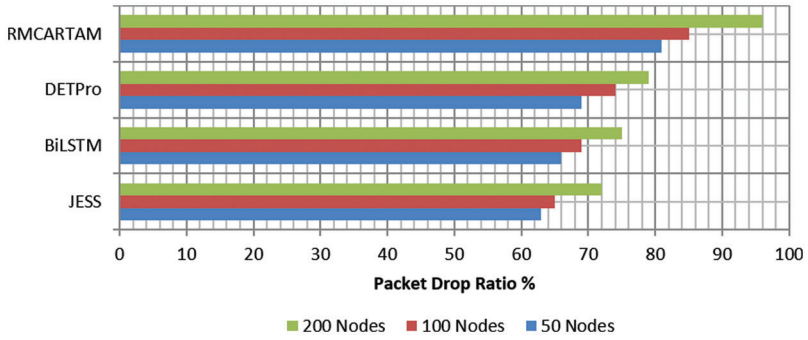

Figure 5: Analysis of PDR

4.4 Packet Drop Ratio

The PDR represents the rate of drop the algorithm makes, Eq. (6). The PDRs produced by different methods are measured on node conditions like 50, 100, and 200 nodes. The PDR produced by different methods in each simulation condition is measured and compared in Tab. 5. However, the proposed RMCARTAM has introduced 18%, 14%, and 4% PDR, which is more minor than the existing JESS, BiLSTM, and DETPro algorithms.

Table 5: Analysis of PDR

Models	50 Nodes	100 Nodes	200 Nodes
JESS	38	34	29
BiLSTM	35	32	28
DETPro	32	27	23
RMCARTAM	18	14	4

$$\text{PDR} = \frac{\text{Number of Packets Dropped}}{\text{Total Packets Sent}} \times 100 \tag{6}$$

The PDRs produced by different methods are measured on node conditions like 50, 100, and 200 nodes. In each simulation condition, the PDR produced by different methods is measured and compared in Fig. 6. However, the proposed RMCARTAM has introduced 18%, 14%, and 4% PDR, which is less than the existing JESS, BiLSTM, and DETPro algorithms.

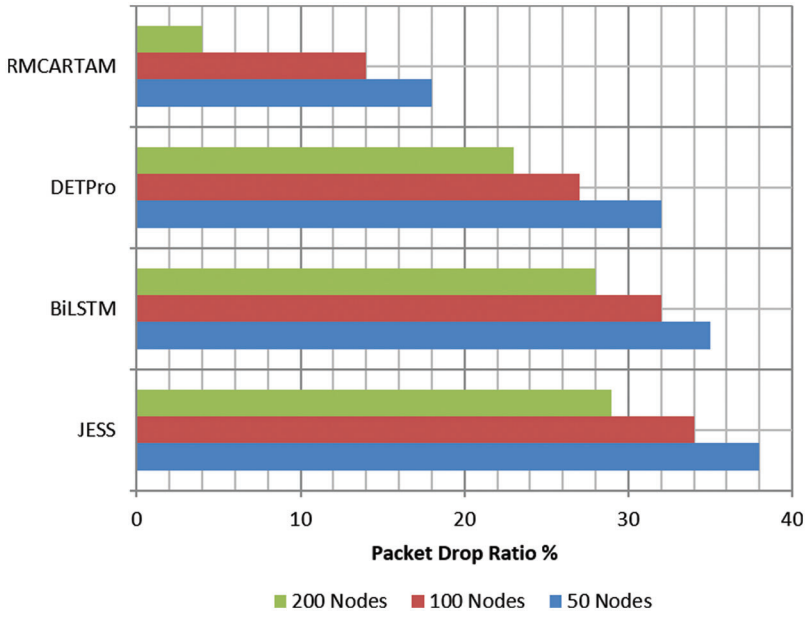

Figure 6: Analysis of PDR

5 Conclusion

This paper presented a RMCARTAM for the detection of DDoS attacks in SDN. The model monitors the traffic and extracts the features from the traffic. According to the features extracted, the method applies multi-constraint adaptive replication to handle the incoming traffic. Similarly, the extracted features are given to the TA model, which analyzes the features to compute various supports on bandwidth, data, route, time, access, and malicious access frequencies. Using all these features, the method computes the value of legitimate weight to classify the incoming packets and perform DDoS attack detection. The proposed method makes it easier for SDN networks to find DDoS attacks and improve QoS.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. M. K. Alvi, S. Faizullah, M. A. Khan, A. Alshanqiti and I. Khan, "Detecting DDoS attack on SDN due to vulnerabilities in OpenFlow," in *Int. Conf. on Advances in the Emerging Computing Technologies (AECT)*, Al Madinah Al Munawwarah, Saudi Arabia, pp. 1–6, 2020.

[2] A. T. Kyaw, M. Zin Oo and C. S. Khin, "Machine-learning based DDOS attack classifier in software-defined network," in *17th Int. Conf. on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Phuket, Thailand, pp. 431–434, 2020.

[3] B. B. Gupta and C. Chaturvedi, "Software-defined networking (SDN) based secure integrated framework against distributed denial of service (DDoS) attack in cloud environment," in *Int. Conf. on Communication and Electronics Systems (ICCES)*, Coimbatore, India, pp. 1310–1315, 2019.

[4] 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 Signal Processing and Communications Applications Conf. (SIU)*, Izmir, Turkey, pp. 1–4, 2018.

[5] B. He, F. Zou and Y. Wu, "Multi-SDN based cooperation scheme for DDoS attack defense," in *3rd Int. Conf. on Security of Smart Cities, Industrial Control System and Communications (SSIC)*, Shanghai, China, pp. 1–7, 2018.

[6] B. Nugraha and R. N. Murthy, "Deep learning-based slow DDoS attack detection in SDN-based Networks," in *IEEE Conf. on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Leganes, Spain, pp. 51–56, 2020.

[7] D. Firdaus, R. Munadi and Y. Purwanto, "DDoS attack detection in software-defined network using ensemble K-means++ and random forest," in *3rd Int. Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, Yogyakarta, Indonesia, pp. 164–169, 2020.

[8] H. A. Alamri and V. Thayananthan, "Bandwidth control mechanism and extreme gradient boosting algorithm for protecting software-defined networks against DDoS attacks," *IEEE Access*, vol. 8, pp. 194269–194288, 2020.

[9] H. Nurwarsito and M. F. Nadhif, "DDoS attack early detection and mitigation system on SDN using random forest algorithm and Ryu framework," in *8th Int. Conf. on Computer and Communication Engineering (ICCCE)*, Kuala Lumpur, Malaysia, pp. 178–183, 2021.

[10] J. A. Pérez-Díaz, 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.

[11] K. Bhushan and B. B. Gupta, "Detecting DDoS attack using software-defined network (SDN) in cloud computing environment," in *5th Int. Conf. on Signal Processing and Integrated Networks (SPIN)*, Noida, India, pp. 872–877, 2018.

[12] K. Hong, Y. Kim, H. Choi and J. Park, "SDN-assisted slow HTTP DDoS attack defense method," *IEEE Communications Letters*, vol. 22, no. 4, pp. 688–691, 2018.

[13] K. Kalkan, L. Altay, G. Gür and F. Alagöz, "JESS: Joint entropy-based DDoS defense scheme in SDN," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2358–2372, 2018.

[14] L. Wang and Y. Liu, "A DDoS attack detection method based on information entropy and deep learning in SDN," in *IEEE 4th Information Technology, Networking, Electronic and Automation Control Conf. (ITNEC)*, Chongqing, China, pp. 1084–1088, 2020.

[15] M. Klymash and O. Shpur, "Concept of intelligent detection of DDoS attacks in SDN networks using machine learning," in *IEEE Int. Conf. on Problems of Info Communications. Science and Technology (PIC S&T)*, Kharkiv, Ukraine, pp. 609–612, 2020.

[16] N. Ahuja and G. Singal, "DDOS attack detection & prevention in SDN using OpenFlow statistics," in *IEEE 9th Int. Conf. on Advanced Computing (IACC)*, Tiruchirappalli, India, pp. 147–152, 2019.

[17] N. I. G. Dharma, M. F. Muthohar, J. D. A. Prayuda, K. Priagung and D. Choi, "Time-based DDoS detection and mitigation for SDN controller," in *17th Asia-Pacific Network Operations and Management Symp. (APNOMS)*, Busan, Korea (South), pp. 550–553, 2015.

[18] Q. Yan, F. R. Yu, Q. Gong and J. Li, "Software-Defined Networking (SDN) and distributed denial of Service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 602–622, 2016.

[19] R. Li and B. Wu, "Early detection of DDoS based on φ-entropy in SDN networks," in *IEEE 4th Information Technology, Networking, Electronic and Automation Control Conf. (ITNEC)*, Chongqing, China, pp. 731–735, 2020.

[20] R. M. Thomas and D. James, "DDOS detection and denial using third party application in SDN," in *Int. Conf. on Energy, Communication, Data Analytics and Soft Computing*, Chennai, India, pp. 3892–3897, 2017.

[21] R. Sanjeetha, K. N. A. Shastry, H. R. Chetan and A. Kanavalli, "Mitigating HTTP GET FLOOD DDoS attack using an SDN controller," in *Int. Conf. on Recent Trends on Electronics, Information, Communication & Technology*, Bangalore, India, pp. 6–10, 2020.

[22] 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.

[23] S. Gupta and D. Grover, "A comprehensive review on detection of DDoS attacks using ML in SDN environment," in *Int. Conf. on Artificial Intelligence and Smart Systems*, Coimbatore, India, pp. 1158–1163, 2021.

[24] R. Sanjeetha, A. Pattanaik, A. Gupta and A. Kanavalli, "Early detection and diminution of DDoS attack instigated by compromised switches on the controller in software defined networks," in *IEEE Int. Conf. on Distributed Computing, VLSI, Electrical Circuits and Robotics*, Manipal, India, pp. 1–5, 2019.

[25] W. Sun, Y. Li and S. Guan, "An improved method of DDoS attack detection for controller of SDN," in *2nd Int. Conf. on Computer and Communication Engineering Technology*, Beijing, China, pp. 249–253, 2019.

[26] 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.

[27] Y. Chen, J. Pei and D. Li, "DETPro: A high-efficiency and low-latency system against DDoS attacks in SDN based on decision tree," in *ICC 2019-2019 IEEE Int. Conf. on Communications (ICC)*, Shanghai, China, pp. 1–6, 2019.

[28] Z. Liu and Y. He, "DDoS attack detection scheme based on entropy and PSO-BP neural network in SDN," *China Communications*, vol. 16, no. 7, pp. 144–155, 2019.

[29] E. Fenil and P. M. Kumar, "Towards a secure software-defined network with adaptive mitigation of DDoS attacks by machine learning approaches," in *Int. Conf. on Advances in Computing, Communication and Applied Informatics*, Chennai, India, pp. 1–13, 2022.

[30] J. Bhayo, R. Jafaq, A. Ahmed, S. Hameed and S. A. Shah, "A time-efficient approach toward DDoS attack detection in IoT network using SDN," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3612–3630, 2022.