Journal of Cyber Security Technology

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tsec20

Modeling DDOS attacks in sdn and detection using random forest classifier

Aishatu Abdullahi Wabi, Ismail Idris, Olayemi Mikail Olaniyi, Joseph A. & Olawale Surajudeen Adebayo

To cite this article: Aishatu Abdullahi Wabi, Ismail Idris, Olayemi Mikail Olaniyi, Joseph A. & Olawale Surajudeen Adebayo (06 Oct 2023): Modeling DDOS attacks in sdn and detection using random forest classifier, Journal of Cyber Security Technology, DOI: 10.1080/23742917.2023.2264435

To link to this article: https://doi.org/10.1080/23742917.2023.2264435

Published online: 06 Oct 2023.

JOURNAL OF CYBER SECURITY TECHNOLOGY
https://doi.org/10.1080/23742917.2023.2264435

Taylor & Francis
Taylor & Francis Group

Modeling DDOS attacks in sdn and detection using random forest classifier

Aishatu Abdullahi Wabi, Ismail Idris, Olayemi Mikail Olaniyi, Joseph A. and Olawale Surajudeen Adebayo

Department of Cyber Security, Federal University of Technology, Minna, Minna, Nigeria

ABSTRACT
A Software-defined network paradigm provides flexibility and programmability to deal with the growing users of future networks. As a result of the centralized control attribute, it could be regarded as a single point of failure that is vulnerable to various forms of attacks, such as Distributed denial of service (DDOS) attacks. This study attempts to show a mathematical representation of DDOS attacks in SDN, together with how some five-tuple features contribute to the attacks. The studied features were used to detect DDOS using a random forest classifier. The result shows 96.3% detection accuracy and 96.45% precision.

ARTICLE HISTORY
Received 27 February 2023
Accepted 25 September 2023

KEYWORDS
SDN; DDOS attacks; detection; random forest; attack modelling

1 Introduction

The use of the internet for business, medias, transactions and so on has become a daily essential for people all over the world. The complexity in the traditional network architecture brought about some drawbacks, such as non-scalability inconsistencies in policies and its dependence on vendor [1]. The networking switch of the traditional network comprises both the control and data plane. Hence, when a new service is required, each device on the network (switch) needs to be configured or updated independently. In the bid to reduce this complexity, software-defined network (SDN), a new network paradigm, has emerged and plays a significant role in ensuring agility, programmability and centralized control of a network [2]. It innovatively disintegrates the control plane from the data plane [3,4], which has improved the overall network management. The application plane, control plane and data plane make up the architecture of the SDN as shown in Figure 1 [5]. The application plane holds business applications, security service and network service applications [23]. The control plane defines the network behaviour through a centralized controller, while the data plane are simply data forwarding devices that act based on instruction from the SDN controller.

CONTACT Aishatu Abdullahi Wabi at aishatuwabi@gmail.com Department of Cyber Security, Federal University of Technology, Minna, Minna, Nigeria

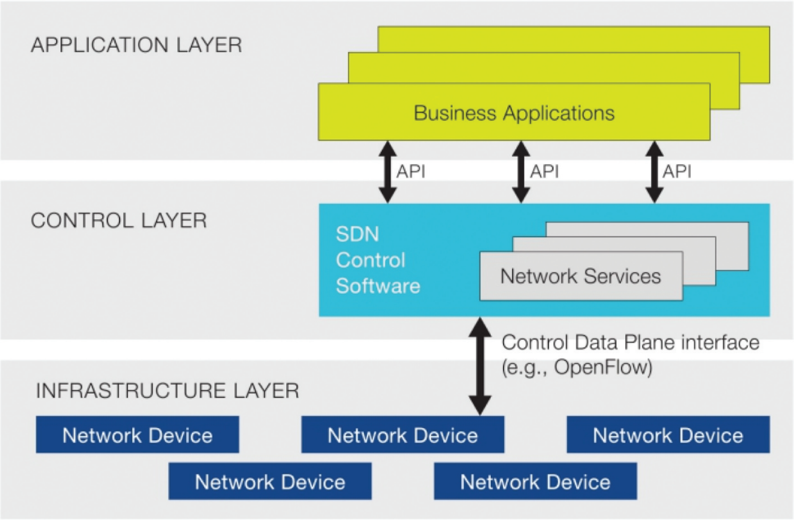

Figure 1. SDN architecture [5].

The overall goal of SDN is to ensure flexibility, programmability and centralized control of the network [6]. However, some of the features that SDN presents make it vulnerable to various security challenges, such as Distributed denial of Service (DDOS) attacks [6], thereby threatening the availability and flexibility of the SDN resources [2].

DDOS attacks are simply a means to make network resource of a system unusable or inaccessible. There is a substantial growth of DDOS attacks as a result of the internet development and usage. Hence, it has become a new threat and worldwide challenge to the availability of the internet services [7].

The DDOS attacks have the potential to affect the application plane of the SDN by utilizing unauthorized applications and make it to launch an attack [8]. The control plane can also be affected by exhausting the controller with a series of packet_in messages [9,10] and the data plane by overloading the flow table space of the SDN switch through sending of multiple unique flows [11]. The controller-data plane and control-application plane bandwidth also have potential to be affected by DDOS attacks in SDN [12]. A taxonomy of DDOS attacks in SDN is depicted in Figure 2.

2 Literature review

Several authors have worked on DDOS attack detection using methods such as statistical and information entropy-based techniques. A DDOS attack detection approach that combines entropy and ensemble learning was proposed by Yu et al. [13] using five tuple features. Fouladi et al. [14] proposed a DDOS attack detection and defense mechanism based on statistics and time series analysis. The study extracted two key features from open flow switch flow table such as

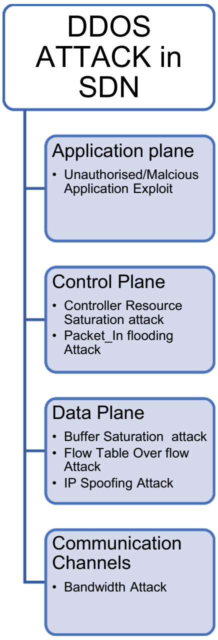
DDOS ATTACK in SDN
Application plane
• Unauthorised/Malcious Application Exploit
Control Plane
• Controller Resource Saturation attack
• Packet_In flooding Attack
Data Plane
• Buffer Saturation attack
• Flow Table Over flow Attack
• IP Spoofing Attack
Communication Channels
• Bandwidth Attack

Figure 2. DDOS attacks in SDN.

unique source IP address, and normalized unique destination IP address and used it to detect instant changes in network behaviour. Sahoo et al. [15] proposed a detection and mitigation techniques using support vector machine (SVM) in conjunction with kernel principal component analysis and genetic algorithm. The model was validated with two datasets independently with high accuracy compared with the existing SVM. In Santos and Moreno's [16] work, SVM, multiple-layer perceptron, decision tree and random forest (RF) were used to classify DDOS attacks in SDN environment. Ye et al. [17] proposed a DDOS attack classification based on SVM. The authors utilized six tuple features for DDOS detection. Myint et al. [18] used five tuple traffic features for detecting DDOS attacks using advanced SVM technique.

Most of the reviewed studies did not show a mathematical representation on how the features used for the detection approaches contribute to DDOS attacks in SDNs. However, in this study, we have demonstrated how the used features contributed to DDOS attacks in SDN and have implemented the selected features for detection of DDOS attacks in SDN environment.

3 DDOS attack model in SDN

In DDOS attack, the goal of the attacker is to overflow the resources of the target host to interrupt the benign host. In the operation of SDN, a new flow is first sent to the controller as *packet_in*, and at the same time, all or part of the flow is stored in the switch buffer while the controller decides the optimal path and sends it back as *packet-out* when large number of flows are sent concurrently at the same time. The switch buffer becomes overloaded, the controller resource becomes saturated and the control-data plane bandwidth becomes exhausted. A flow can be defined by many characteristics stored in the switch flow table such as source IP address, destination IP address, destination port, flow duration, flow packet, flow bytes and so on. Most of these flow characteristic features undergo different changes during attack and normal situation. To describe the attack model in SDN, the following assumptions are made:

(a) The traffic characteristics, such as number of source IP, number of flow entry and so on, are considered to follow Poisson arrival as it has been widely accepted that the arrival of request follows Poisson distribution [19], while the duration service time is generally distributed since each flow may have different duration in the network.

(b) It is assumed that each flow is on a queue, waiting to be served by a single server with one communication channel.

3.1 Mathematical model

Given an SDN controller C with service time t_c and average packet duration T connected to a switch $S = \left\{s_1, s_{2,}s_3 \ldots, s_n\right\}$ via a bandwidth K and each switch is connected to host $h = \left\{h_1, h_{2,}h_3 \ldots, h_n\right\}$
Assuming a traffic is launched from multiple host h_n to a switch S, the switch sends the traffic as packet-in to the controller C for processing via the Bandwidth K, provided it is a new traffic flow, otherwise the traffic is sent directly to the target host *h*.

3.1.1 Flow table overloading model

To estimate the probability of attack on the switch resulting to flow table overloading, we assume there is S number of switches where each switch *i* for $1 \leq i \leq S$ is connected to host *n* host and S_t is the flow table length L or capacity.

Recall that in SDN paradigm when a new flow arrives the switch, then a new flow rule will be installed, otherwise it is dropped. We assume that each flow that arrives at the switch is new since the goal of the DDOS attack is to flood the switch with large number of flows.

Furthermore, the probability of the flow table overloading is estimated by the system as M/M/L/L queuing model. This means the queuing system follows Poisson arrival with an exponential service rate with L number of servers.

The arrival rate of the system is the average speed of flow entry λ_s and the service time is considered to be the average duration time in the flow table t_s.

In an attack scenario, the number of flow entry increases per unit time and eventually overloads the flow table, so when a new flow arriving the switch whether legitimate or bot will be lost or block. In this scenario, assume that there are no queue and if a server (flow table) is occupied, the newly arriving flow will be lost and the probability that a newly arriving flow is dropped by the switch because the switch occupied is estimated in (3.10):

$$P(S) = \frac{\frac{\rho^L}{L!}}{\sum_{i=0}^{L} \frac{\rho^i}{i!}} \tag{3.10}$$

The average service rate $\mu = \frac{1}{t_s}$ and the server utilization or load (ρ) in this context is the flow length utilization $\rho = \frac{\lambda_s}{\mu}$. Therefore, $\rho = \lambda_s \cdot t_m$

3.1.2 Communication channel (bandwidth) model

In estimating the probability of bandwidth attack, the M/M/K/K queuing system is adopted where the M represents Markovian, K is the communication channel with bandwidth K = 1, that is, single channel. This is suitable for the SDN given that there is only one communication channel between the controller and switch. Then the arrival rate is the average packet rate, λ_B is Poisson distributed and service rate μ_B is generally distributed. The average size of flow packets (flow byte) is denoted by υ, and average delay d with a throughput T, where

$T = \frac{\upsilon}{d}$; $\upsilon = \frac{\sum_{i=1}^{N_{pkt}} f_i}{N_{pkt}}$ where $f = flowbytes$ per packets and N_{pkt} is total packets.

$$\mu_B = \frac{\sum_{i=1}^{K} T_i}{\upsilon} \tag{3.11}$$

Following the blocking probability formula that corresponds to the P(B), bandwidth saturation is estimated in (3.12) as follows:

$$P(B) = \frac{\frac{\rho^K}{K!}}{\sum_{i=0}^{K} \frac{\rho^i}{i!}} where, \rho = \frac{\lambda_B}{\mu_B} \tag{3.12}$$

3.1.3 Controller attack model

In modeling controller attack, the M/M/1 queuing model is adopted with no blocking or queuing. In a stable state, the speed of arrival λ_c should be less than the service time μ_c.

However, each packet will require average time L on the system. This can be computed as follows:

$L = \frac{1}{\mu_c - \lambda_c}$ when $\frac{\lambda_c}{\mu_c} = \rho < 1$ ρ represents average proportion of time the controller is occupied.

The average time spent waiting $t_w \frac{1}{(\mu - \lambda)} - \frac{1}{\mu} = \frac{\rho}{\mu - \lambda}$
The probability that the system contains i *number of packets* is $\pi_i = (1 - \rho)\rho^i$; *thus, the average packets* = $\frac{\rho}{(1-\rho)}$ *and variance* =$\frac{\rho}{(1-\rho)^2}$
The P(C) of controller attacker is estimated taking into account the time a packet is willing to wait t_w and the average time L a packet spends in the system (3.13).

$$P(C) = \begin{cases} 1, \rho \geq 1, orL \geq t_w \\ \frac{L}{t_w}, \rho < 1, and.L < t_w \end{cases} \tag{3.13}$$

As a result of the centralized control of SDN, it is regarded as a single point of failure; hence, the probability of attack on the bandwidth to affect control plane.

The P_{sdn} that the network failing is the probability 1 or 2 of the SDN company is successful

$$P_{sdn} = 1 - \overline{P(S)} \cdot \overline{P(B)} \cdot \overline{P(C)} \text{ OR } 1 - \left(\overline{P(S)} + \overline{P(B)}\right) \cdot \left(\overline{P(B)} + \overline{P(C)}\right) \tag{3.14}$$

From the model, it can be seen that the number of flow entry, flow packets, duration, flow_bytes and number of hosts are important parameters that contribute to DDOS attacks in SDN.

Hence, the eigenvalues are used to estimate the speed of flow entry, average flow packet, the increase in source IP, average duration of flows and average flow bytes (AFB) for DDOS attack detections.

3.2 Ddos detection

The workflow for the detection approach is depicted in Figure 3 and has been discussed as follows:

3.2.1 Traffic generation

This entails generating both normal and attack traffic on an SDN network and is sent to the next phase. To achieve this, three forms of DDOS attacks were launched on the SDN network, namely, TCP Syn flooding attack, UDP flooding attack and ICMP flood attack. A single network topology consisting of a single switch and five hosts was used for the study. Mininet was used as the emulator for the SDN network, Ryu was utilized as the controller and *Hping3 tool* was used to generate attack traffic. For generating the normal traffic, the *Iperf* and *ping tool* was used.

3.2.2 The statistic collection

This phase is used to collect traffic features from SDN after traffic generation. This is achieved by sending a statistics request to the switch via the SDN controller using *OFPFLOWSTATREQUEST* handler, and a response is received using the *OFPFLOWSTATREPLY* handler. The *OFPFLOWSTATREQUEST* is a method used to request the switch statistics information such as source

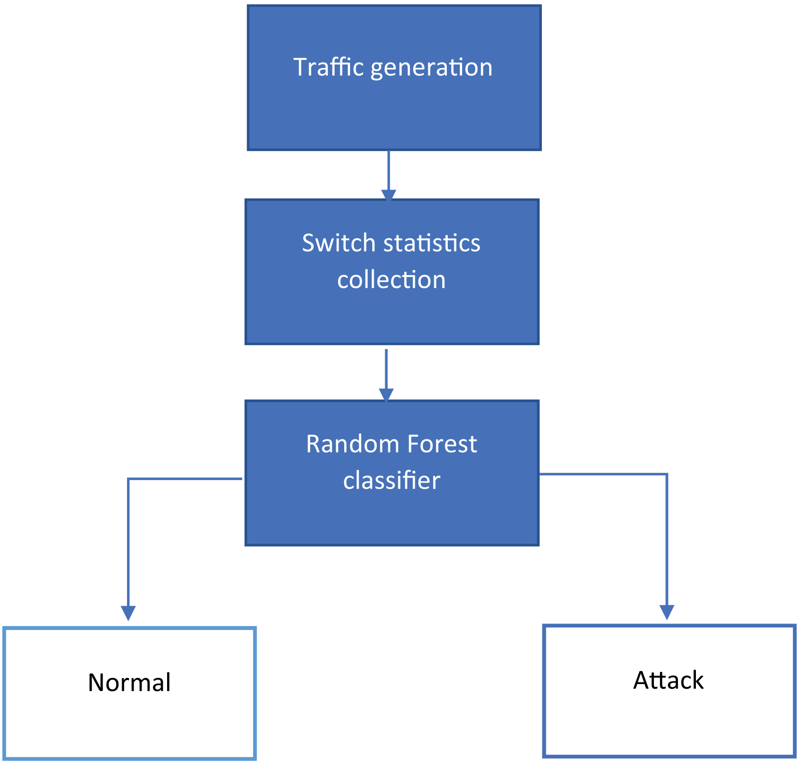

Figure 3. The workflow for the proposed detection approach.

address, destination address, flows, flow packets, flow duration and so on. In order to receive a response to this request, the *OFPFLOWSTATREPLY* handler is called to retrieve the flow statistics. The collected statistics were used to extract the following traffic features and create a dataset and saved in a comma separated values (CSV) file. The features include the number of flow entry per unit time (NFE), average flow packet, the source IP per unit time (SIP), average duration of flows and AFB.

(i) NFE: The sum of flow entry within a sampling interval. This feature is useful because the number of flows tends to increase during a DDOS attack situation than the normal situation.

$$\text{NFE} = \frac{\sum flow\ entry}{T} \tag{3.15}$$

(ii) Average flow packet (average_pkt): The sum of flow packet within a sampling interval with the total number of flow packet. Large number of packets are generated during DDOS attack, since the goal is to flood network with large number of packets. Hence, monitoring the average number of flow packets will be useful.

$$AFP = \frac{\sum flow\ packet}{total\ flow} \tag{3.16}$$

(iii) SIP: This is the total number of IP address generated within a sampling interval. Large number of IP addresses tend to increase during DDOS attack, hence monitoring this feature will aid detection.

$$\mathrm{SIP} = \frac{\sum IP}{T} \tag{3.17}$$

(iv) Average flow duration (avg_durat): This is the mean of the total duration of each flow in seconds within a sampling interval. Flows are sent within a sampling interval during DDOS attack, hence monitoring how long the flows spent in the flow table will be useful during attack detection.

$$AFD = \frac{\sum flow\ duration}{total\ flow} \tag{3.18}$$

(v) Average flow byte (average_byte): This is the mean number of packet bytes within a sampling interval. Since large packets are sent during attack, monitoring the flow byte will aid detection.

$$AFB = \frac{\sum flow\ byte}{total\ flow} \tag{3.19}$$

3.2.3 Random forest

Classifier to categorize the traffic as either normal or attack traffic, RF is utilized as the classifier. It employs many decision trees for the classification process. Other trees can make up for a choice that was made incorrectly by one decision tree. The classification outcome is provided by each decision tree, and the categorization recommendation is made using the majority of votes [20]. The steps for implementing RF [21] are as follows:

(a) Take N as the number of training data instances in the samples.
(b) Let M be the number of attributes in given input dataset.
(c) Let m be the number of parameters in the input that determines the next attribute to be chosen at each tree node (where m is lesser than M).
(d) The training samples are taken and a tree is constructed for each sample with replacement.
(e) For tree node, arbitrarily select m attributes in that particular node.
(f) The best split is computed based on the m input attributes of the sample dataset.

(g) Each tree is grown without pruning.

3.2.4 Evaluation

The following metrics were used to evaluate the performance of the detection approach.

(i) **Accuracy**: The accuracy rate is used as an evaluation index to evaluate the detection performance of the model [22].

$$Accuracy = \frac{TP + FN}{TP + TN + FP + FN} \tag{3.20}$$

FN: false negative; FP: false positive; TN: true negative; TP: true positive.

(ii) **Precision**: The precision is the measurement of the retrieved instances that is relevant to these instances [15,17].

$$Precision = \frac{TP}{TP + FP} \tag{3.21}$$

4 Experiment

The experiment was conducted on HP Laptop 4 GB RAM 1 TB Intel Core i7 with a 64 bit processor. A single network topology as shown in Figure 4 was used in the experiment. The topology consisted of five hosts, which are connected to one open virtual switch connected to a single RYU controller on Mininet emulator. Both normal attack and attack traffic were generated. Hping3 was used to implement the attack, while Iperf was used to generate and ping was used to implement normal traffic.

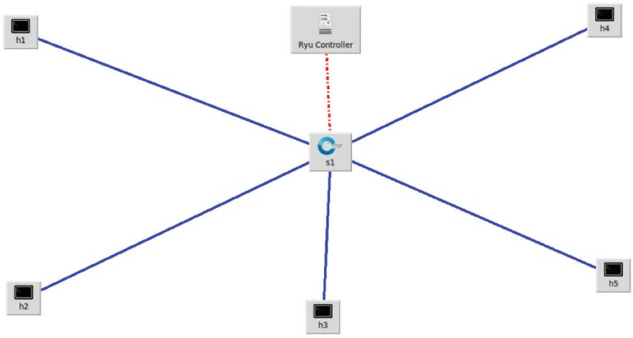

Figure 4. The topology used for the experiment.

Table 1. The attack types and packet rate.

Attack type	Number of packet/requests	Inter-packet delay (s)
SYN flooding	1000	0.01
		0.025
ICMP flooding		0.02
		0.035
UDP flooding		0015
		0.03

Table 2. Dataset instances.

Sample	Attack	Normal
	118	1189

During the training, Host 1 (h1) and Host 2 (h2) were used as attacker while Host 5 (h5) was used as the victim. The generated traffic for both attack and normal consists of TCP, ICMP and UDP traffic.

During each attack traffic generation, high- and low-rate attack is launched using varying inter-packet delay as depicted in Table 1. The traffic was generated for a period of 3600 s, and statistics collected every 5 s. The collected traffic statistic was saved in a CSV file and used as dataset. The dataset instances are shown in Table 2.

4.1 Results and discussion

The RF algorithm was applied on the dataset with different training and test sets. The accuracy and precision scores of each of the split and for different training sample is depicted in Table 3 and graphically shown in Figures 5 and 6, respectively. The average of the accuracy and precision was computed to be 96.3% and 96.45%, respectively, as shown in Figure 7. From the accuracy score, it can be deduced that the features used for detection have significant influences on the performance of the classification model. The precision score shows a high number of positive predictions, and since the goal is to detect DDOS attacks, achieving 96.45% precision score is noteworthy.

From the result achieved, it is evident that a network administrator of an organization can incorporate these traffic features and the use the RF classifier to detect DDOS attacks at an early stage. The experiment was conducted using the generated dataset on an emulator, together with the software version of a real switch. Hence, the achieved results will be valid in real time. Although the

Table 3. Comparison of accuracy and precision for different training and test sets.

Training %	Test %	Detection accuracy %	Precision %
66	34	95.4	95.7
70	30	95.48	95.6
80	20	96.56	96.7
90	10	97.71	97.8
		96.3	96.45

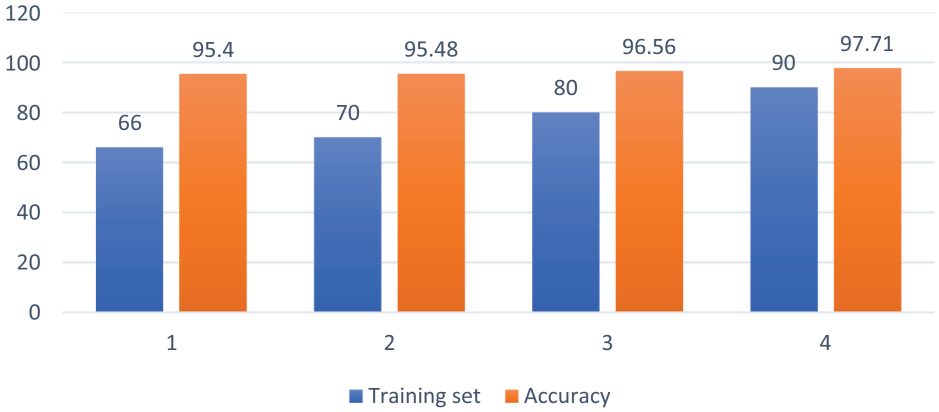

Figure 5. Comparison of accuracy for different training samples.

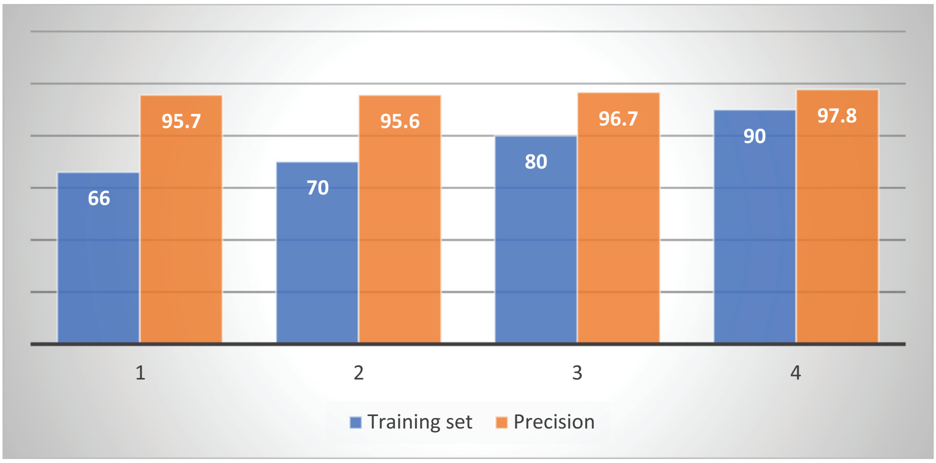

Figure 6. Comparison of precision for different training data.

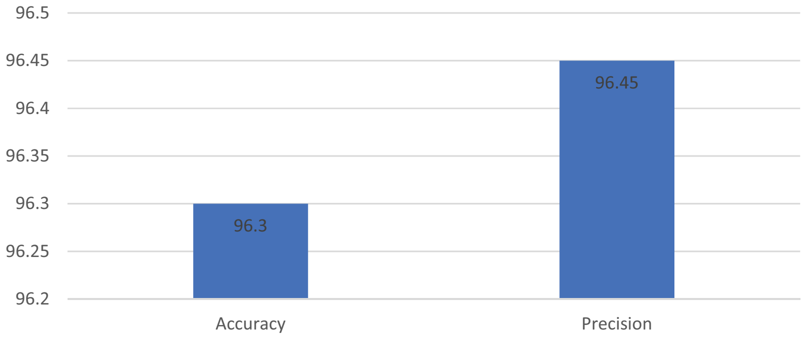

Figure 7. Average accuracy rate and precision for the detection approach.

dataset was not comprehensive enough, the features used in the dataset can be utilized to generate a more comprehensive dataset using the described method in real time.

5 Conclusion

In this study, a mathematical representation of DDOS attacks in SDN was presented together with how some traffic features contributed to DDOS attack in SDN. The values of these features were collected from the SDN switch by the controller and used to detect DDOS attacks in SDN. The result shows a weighted average accuracy of 96.3%. This means that the features used in the dataset for the DDOS attack detection are significant and effective. While the 96.45% precision shows that the RF classification model was able to predict the class labels correctly in most cases.

5.1 Future work

The dataset used in this work was not comprehensive enough; hence, a more comprehensive dataset could be generated using the discussed traffic feature as a future work. Additionally, using a larger network topology and detecting attacks in real time shall be considered in the future research.

Acknowledgements

The authors would like to thank the referees, reviewers and editors for their valuable comments and their feedback for better improvement of this article.

Disclosure statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

ORCID

Aishatu Abdullahi Wabi http://orcid.org/0000-0002-5493-9385
Ismail Idris http://orcid.org/0000-0003-1462-1521

References

[1] Gong Y, Huang W, Wang W, et al. A survey on software defined networking and its applications. Front Comput Sci. 2015;9(6):827–845. doi: 10.1007/s11704-015-3448-z

[2] Karnani S, Shakya HK. Mitigation strategies for distributed denial of service (DDoS) in SDN: a survey and taxonomy. Inf Secur J A Global Perspect. 2022;32(6):444–468. doi: 10.1080/19393555.2022.2111004

[3] Ramprasath J, Krishnaraj N, Seethalakshmi V. Mitigation services on SDN for distributed denial of service and denial of service attacks using machine learning techniques. IETE J Res. 2022;1–12. doi: 10.1080/03772063.2022.2142163

[4] Xie J, Richard Yu F, Huang T, et al. A survey of machine learning techniques applied to software defined networking (SDN): research issues and challenges. IEEE Commun Surv Tutorials. 2019;21(1):393–430. doi: 10.1109/COMST.2018.2866942

[5] Open Networking Foundation. Software-defined networking: the new norm for Networks [white paper]. *ONF White Paper*. 2012; 1–12.

[6] Hussein A, Chadad L, Adalian N, et al. Software-defined networking (SDN): the security review. J Cyber Secur. 2020;4(1):1–66. doi: 10.1080/23742917.2019.1629529

[7] Suhag A, Daniel A. Study of statistical techniques and artificial intelligence methods in distributed denial of service (DDOS) assault and defense. J Cyber Secur. 2023;7(1):21–51. doi: 10.1080/23742917.2022.2135856

[8] Hafizah S, Ariffin S, Muazzah N, et al. A review of anomaly detection techniques and Distributed Denial of Service (DDoS) on Software Defined Network (SDN). Tech App Sci Res. 2018;8(2):2724–2730. doi: 10.48084/etasr.1840

[9] Polat H, Polat O, Cetin A. Detecting DDoS attacks in software-defined Networks through feature selection methods and machine learning models. Sustainability. 2020;12(3):1035. doi: 10.3390/su12031035

[10] Raghunath K, Krishnan P Towards a secure SDN architecture. *2018 9th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2018*; 2018. p. 1–7. 10.1109/ICCCNT.2018.8494043

[11] Wu X, Liu M, Dou W, et al. DDoS attacks on data plane of software-defined network: are they possible ? Secur Commun Networks. 2016;9(18):5444–5459. doi: 10.1002/sec

[12] Mladenov B Studying the DDoS attack effect over SDN controller southbound channel. *2019 X National Conference with International Participation (ELECTRONICA)*, Technical University of Sofia, Bulgaria, May 16–17, 2019. 2019. p. 1–4.

[13] Yu S, Zhang J, Liu J, et al. A cooperative DDoS attack detection scheme based on entropy and ensemble learning in SDN. EURASIP J Wireless Communicat Network. 2021;2021(1). doi: 10.1186/s13638-021-01957-9

[14] Fouladi RF, Ermiş O, Anarim E. Journal of information security and applications a DDoS attack detection and defense scheme using time-series analysis for SDN. J Inf Secur Appl. 2020;54(August):102587. doi: 10.1016/j.jisa.2020.102587

[15] Sahoo KS, Tripathy BK, Naik K, et al. An evolutionary SVM model for DDOS attack detection in software defined Networks. IEEE Access. 2020;8:132502–132513. doi: 10.1109/ACCESS.2020.3009733

[16] Santos R, Moreno E. Machine learning algorithms to detect DDoS attacks in SDN. Concurr Comput. 2019;32(16):1–14. doi: 10.1002/cpe.5402

[17] Ye J, Cheng X, Zhu J, et al. A DDoS attack detection method based on SVM in software defined network. Secur Commun Networks. 2018;2018. doi: 10.1155/2018/9804061

[18] Myint Oo M, Kamolphiwong S, Kamolphiwong T, et al. Advanced Support Vector machine-(ASVM-) based detection for distributed denial of service (DDoS) attack on software defined networking (SDN). J Comp Networks Commun. 2019;2019. doi: 10.1155/2019/8012568

[19] Yuan B, Zou D, Yu S, et al. Defending against flow table overloading attack in software-defined networks. Methods Mol Biol. 2016;1374(c):1–22. doi: 10.1109/TSC.2016.2602861

[20] Ahuja N, Singal G, Mukhopadhyay D, et al. Journal of network and computer applications automated DDOS attack detection in software defined networking. J Network Comput Appl. 2021;187(November 2020):103108. doi: 10.1016/j.jnca.2021.103108

[21] Shaik AB, Srinivasan S. A brief survey on random forest ensembles in classification model. In: Lecture notes in networks and systems. Vol. 56. Singapore: Springer; 2019. pp. 253–260. doi: 10.1007/978-981-13-2354-6_27

[22] Sun W, Li Yi, Guan S. An improved method of DDoS attack detection for Controller of SDN. 2019 IEEE 2nd International Conference on Computer and Communication Engineering Technology (CCET), Beijing, China,16 -18 August 2019. 2019. IEEE. p. 249–253. doi: 10.1109/CCET48361.2019.8989356

[23] Cabaj K, Wytrębowicz J, Kukliński S, et al. SDN Architecture Impact on Network Security. Federated Conference on Computer Science and Information Systems (FEDCSIS). 29 September 2014 Warsaw, Poland.