

Received December 2, 2019, accepted December 22, 2019, date of publication December 30, 2019, date of current version January 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2963077

DDoS Attack Detection Method Based on Improved KNN With the Degree of DDoS Attack in Software-Defined Networks

SHI DONG¹ AND MUDAR SAREM^{2,3}

¹School of Computer Science and Technology, Zhoukou Normal University, Zhoukou 466001, China

²School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

³General Organization of Remote Sensing, Damascus 12586, Syria

Corresponding author: Shi Dong (njbsok@163.com)

This work was supported in part by the Key Scientific and Technological Research Projects in Henan Province under Grant 192102210125, and in part by the Study Abroad Activities of Science and Technology Project of Henan Province.

ABSTRACT The Distributed Denial of Service (DDoS) attack has seriously impaired network availability for decades and still there is no effective defense mechanism against it. However, the emerging Software Defined Networking (SDN) provides a new way to reconsider the defense against DDoS attacks. In this paper, we propose two methods to detect the DDoS attack in SDN. One method adopts the degree of DDoS attack to identify the DDoS attack. The other method uses the improved K-Nearest Neighbors (KNN) algorithm based on Machine Learning (ML) to discover the DDoS attack. The results of the theoretical analysis and the experimental results on datasets show that our proposed methods can better detect the DDoS attack compared with other methods.

INDEX TERMS DDoS attack, traffic behavior, software defined networking, gain value.

I. INTRODUCTION

Distributed Denial-of-Service (DDoS) attacks have drawn extensive attention in the cyberspace during the last few years. In the recent years, the concepts and the techniques of the Software Defined Networking (SDN) have been introduced and widely researched. The DDoS attacks can threaten the availability of the SDN due to the difference in the architecture between the SDN network and the traditional network. Especially, the SDN controller is the most vulnerable part to be affected by the DDoS attacks. In general, the DoS attack is an attempt to make the resources of a network unavailable for legitimizing users. Shin and Gu [1] initiated a DoS attack on an SDN using separated logic of the SDN in the control-data planes and developed a network scanning tool that could identify an SDN network. In their method, since the data path had different values in the flow response times for the existing and the new flows due to the querying of the controller, the time values were gathered based on the header field by the scanner which could scan the network in order to change the network header fields. Once the network was found to be considered

as a SDN network, the flow requests were transmitted to the target network, which were forwarded by the data path to the controller. However, increasing the number of the flows in the data path will make the switches suffer from flow setup requests on the controller and hence eventually cause it to be broken. Fonseca *et al.* [2] denoted that a DDoS attack on the SDN controller is where an attacker continuously sends IP packets with random headers to disrupt the controller. In [2], a secondary controller was adopted to improve the resilience. However, a DDoS detection mechanism was required since the secondary controller could also be vulnerable to the DoS or the DDoS attacks. Hence, the use of multiple controllers still could not completely resolve the problem of the DDoS attacks since it could lead to cascading fault of multiple controllers [3]. Some DDoS attack detection methods in the SDN had been proposed [4]–[8]. Lin and Wang [4] proposed a DDoS attack detection method based on the SDN. In fact, their method used three Openflow management tools with sFlow to detect anomaly network traffic. Therefore, the operation and the deployment of their proposed method were complex. Yang *et al.* [5] presented a new method that only used single flow information and an IP entropy characteristic information. Although their experimental results

The associate editor coordinating the review of this manuscript and approving it for publication was Adnan M. Abu-Mahfouz ¹.

showed that their method had high accuracy detection, but it needed more technology to determine the threshold and the multi-element weight distribution. Saied *et al.* [6] proposed an ANN algorithm to detect DDoS attacks. However, due to the need to distinguish the packet protocol, their method was complex and inefficient. Ye *et al.* [7] proposed combining Support Vector Machine classification algorithms (called as SVM) to build the DDoS attack model. In their method, six feature values (SSIP, SSP, SDFP, SDFB, SFE, and RPF) were introduced. Their experimental results showed low false alarm rates for the TCP and the UDP traffics, but the ICMP traffic's false alarm rate was high. Cui *et al.* [8] proposed a mechanism using Cognitive-Inspired Computing and Support Vector Machine classification algorithm (called as CIC-SVM) to detect the DDoS attack. Meanwhile, their detection accuracy still needs to be further improved. The authors in [9]–[11] proposed some DDoS detection methods. However, these methods were vulnerable to other factors, and the research results of these methods showed that the behavior features were very important for the DDoS detection in the SDN. So in this paper, we have proposed several features and analyzed the traffic behavior with the DDoS attack in order to provide the suggestion for the DDoS detection in the SDN network. Moreover, we have proposed DDoS Detection Algorithm based on the Degree of Attack (called DDADA) and DDoS Detection Algorithm based on Machine Learning (called DAMDL). The proposed algorithms can effectively identify the DDoS attacks in the SDN environment.

The contribution of this work can be summarized in the following points: Firstly, we have proposed four features (called flow length, flow duration, flow size, and flow ratio) in order to evaluate the DDoS attack detection performance when the SDN controller is attacked by the DDoS attack. Secondly, for the first time, a new concept called the degree of attack is proposed to detect the DDoS attack. Thirdly, based on this concept, a detection algorithm based on the degree of the attack (called DDADA) is proposed. And finally, in order to further improve the detection efficiency, another detection algorithm based on machine learning (called DDAML) is introduced to identify the DDoS attack.

The paper is organized as follows. Section II gives out an introduction and an overview of the classification of the DDoS attacks; Section III discusses the behavior features of the DDoS attack in the SDN network. In Section IV, we discuss our algorithms. The experimental results for different algorithms of the DDoS attack are shown in Section V; and finally, Section VI sums up the paper and points out to the focus of our future work.

II. SDN AND OVERVIEW FOR THE CLASSIFICATION OF DDoS ATTACKS

In this section, we survey the basics of the software defined network and discuss its architecture and features. In addition, an overview of the classification of the DDoS attacks is summarized.

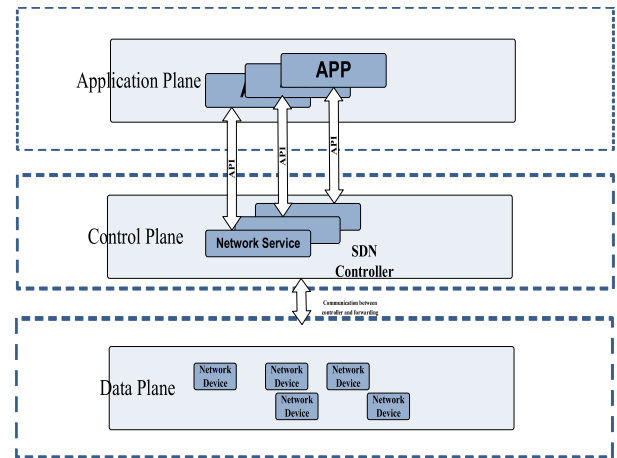


FIGURE 1. SDN architecture.

A. AN OVERVIEW OF SDN

SDN is currently attracting abundant attention of researchers, aiming to provide open, centralized, decoupled, programmable, flow-based, and dynamic network switching mechanisms. In addition, there are some core differences that distinguish SDN networks from traditional networks. So, when implementing the networks, there are some details that are only specific to the SDN networks. Unlike the SDN, in traditional networks, the networking devices decide how an incoming packet should be solely handled based on its IP destination address. Meanwhile, the SDN carries out a flow-based forwarding scheme where multiple header fields clearly state how the incoming packet should be handled by a switch. Then, all the network devices record all the traffic statistics in the SDN network. However, some statistics are performed by only few devices in the traditional networks. Due to logically centralizing the network control plane and the introducing of programmability, the SDN simplifies both the network management and the run-time deployment of the security policies. With the help of the SDN, the network security systems can quickly respond to the network anomalies and the traffic status. In order to further set forth the SDN architecture and functionality, three main functional layers (i. e., the SDN planes) are shown in Fig. 1 [12], [13]. As it can be seen from Fig. 1, the SDN architecture is respectively composed of three parts: Application Plane, Control Plane and Data Plane. The application plane is on the top of the SDN architecture and contains the SDN applications for various functionalities, such as policy implementation, network management and security services. In the application plane, the SDN application can use the programmable method to submit the network behavior to the control plane. The control plane is a logically centralized control framework that runs the Network Operation System (NOS). In this plane, the hardware abstractions are provided to the SDN applications. It maintains a global view of the network. The entity that implements the control plane functionalities is represented as the SDN controller [14], which connects the application plane through the north-bound interface. The

data plane is a combination of forwarding elements used to forward traffic flows based on instructions from the control plane. In addition, the north-bound interface is used as an interface between the application plane and the control plane. Up till now, the interface is still not standardized. The south-bound interface is referred as the interface between the control plane and the data plane.

B. CLASSIFICATION OF DDoS ATTACKS

In this sub-section, we introduce the DDoS attack and its major classification. The DDoS attack is launched by multiple compromised computers called as bots or zombies targeting a single system. The attacker remotely controls these computers (i. e., bots and zombies) to attack other computers. In order to complete a DDoS attack, four major components must be included. One is the real attacker, and the second is the compromised hosts called as handlers or masters which are capable of controlling multiple agents using software programs. The third component is the agent hosts which generate a large number of packets towards the victim host. And the fourth component is the target host which is the victim. In the next paragraphs, the classification of the DDoS will be discussed. Specht and Lee [15] introduced some tools used to launch the attack and analyzed in details the possible countermeasures. The classification of the DDoS attack is presented below.

Flood Attacks: A flood attack involves zombies sending large volumes of traffic to a victim system. The flood attack has been launched using the packets of HyperText Transfer Protocol (HTTP), Internet Control Message Protocol (ICMP), User Datagram Protocol (UDP) or Session Initiation Protocol (SIP). Hence, the flood attack is divided into the following four attacks: HTTP flood, ICMP flood, UDP flood and SIP flood attacks. The victim system is attacked by continuously sending UDP packets to a specific or a random port. In the ICMP flood attack, a large number of ICMP echo request packets with spoofed source IP addresses are sent to the victim. The HTTP flood attack is a volumetric attack done by sending abundant HTTP requests. In the SIP flood attack, the VoIP communications use Session Initiation Protocol (SIP) to send call signal. The SIP phone using SIP can be easily flooded with messages so that it cannot obtain legitimate requests. **Amplification Attacks:** An amplification attack involves the attackers or the zombies who send messages to a broadcasted IP address. This principle will cause all the systems in the subnet reached by the broadcasted address to send a reply to the victim's system. The amplification attacks are divided into Fraggle attack, Smurf attack, and Simple Network Management Protocol (SNMP) amplification attack. The Fraggle attack uses UDP packet in place of ICMP packet. Here, the victim's IP address is used as a spoofed source IP address in the attack packets. The Smurf attack is targeted against the routers and servers where the ICMP packets are redirected to these amplifiers with a spoofed source IP address. The spoofed address will be the

TABLE 1. Features used in the SDN network and observed in our paper.

Feature	Meaning
Flow length	number of Packet
Flow duration	the time interval in flow
Flow size	Bytes in flow
Flow rate	Packets per second

victim host IP address. The sources of the UDP and ICMP flood attacks can be easily tracked, but it is difficult to track the source of the Smurf attack. In the SNMP amplification attack, the SNMP is used to monitor devices such as printers, routers and firewalls. The SNMP uses default communication string which allows programs to get the configuration information of the monitored devices. In order to retrieve the configuration details, the GetBulk request can be sent. The attackers send this request using default communication string with the spoofed source IP address of the target system. Thus, the victim system is overwhelmed with responses. **Coremelt Attack:** in this attack, the zombies can be divided into two groups. The attacker designates the zombies to communicate with the zombies in other group which will lead to sending and receiving huge data. When the communication happens, it is difficult to track this attack through legitimate packets. In fact, in this Coremelt Attack, the Attack's target is not the single host, but also the zombies, and by communicating with each other, they create network flood [16]. So, large numbers of packets are sent to the same host, the destination IP address, and the port number. Then, eventually the system will crash. **TCP SYN attack:** The weakness of the TCP is used to launch this attack. The attacker sends a large number of SYN requests to the server. The server replies to the request by sending SYN + ACK packet and waits for the ACK packet from the client. Let us suppose that the attacker does not send ACK packet, and the server waits for non-existent ACK. The limited buffer queue of the server becomes full and the incoming valid requests will be rejected. **Authentication Server attack:** The authentication server verifies the bogus signature from the attacker which consumes more resources compared with generating the signature. **CGI Request attack:** The attacker sends a large number of CGI requests that consume the CPU cycles and the resources of the victim. Based on the above introduction, we will further study the characteristics of the attacks and the attack detection.

III. FEATURES FOR ATTACK TYPES

The traffic behavior presents the change that happens in the SDN Network after the attack. So, in this paper we consider some features in the SDN network as objects to be observed and studied which include the flow length, the flow duration, the flow size, and the flow rate as shown in Table 1.

A. INFLUENCE OF TRAFFIC BEHAVIOR AFTER ATTACK

The entropy technique introduced in [17] provided a flexible and fast approach to estimate the baseline distribution which could be considered as an anomaly detection method. In this section, we introduce the entropy as a metric to identify the traffic with the DDoS attack. The entropy is given as in Eq. (1) below:

$$Entropy(S) = \sum_{i=1}^n -P_i \log_2 P_i \quad (1)$$

where: $Entropy(S)$ is a function and P_i is a priori probability. We want to determine which attribute in a given set of training feature vectors is the most useful for discriminating between the classes that need to be learned. The gained information tells us how important a given attribute of the feature vectors is. This $Gain$ is denoted by Eq. (2) as follows:

$$Gain(S, F) = Entropy(S) - \sum_{v \in features} \frac{|S_v|}{S} Entropy(S_v) \quad (2)$$

where S_v represents the number of samples values $\in v$. We use a standard approach to normalize the $Gain$ for each feature. Let $NGain$ be as the $Gain$ value after doing standard normalization. This $NGain$ is computed by Eq. (3) as follows:

$$NGain(S, F) = \frac{Gain(S, F) - \max(Gain(S, F))}{\max(Gain(S, F)) - \min(Gain(S, F))} \quad (3)$$

Theorem 1: If entropies of the sample space of the flow length for different flows have no difference, then after the DDoS attack, the entropies still have no difference

Proof 1: Let's assume that the entropy of flow length is $e(len)$ before the DDoS attack, then after the attack, the entropy of the flow length $e_t(len)$ is given as in Eq.(4) below:

$$e_t(len = l) = - \sum_{i=1}^{\ominus} e(len = i) c_i^l p^l (1-p)^{i-l} \log_2 P_i \quad (4)$$

where: e_t represents the post-attack entropy value, e is the pre-attack entropy value. $c_i^l p^l$ and $(1-p)^{i-l}$ are constants, and $e_t(len = l)$ is the entropy of the flow length after the DDoS attack.

Suppose that we have two flow lengths $l1$ and $l2$. If the original two flow lengths have the same values (i.e., $l1 = l2$), then the entropies of the two flow lengths also have the same values (i.e., $e(len = l1) = e(len = l2)$). In two sample spaces, we know that $c_i^l p^l (1-p)^{i-l} \log_2 P_i$ is constant. So, when the pre-attack entropy values of two entropies of the sample space have no difference, by Eq. (4), we can conclude that the post-attack entropy values still have no difference (i.e., $e_t(len = l1) = e_t(len = l2)$).

Theorem 2: If the entropy of the flow features was known then, after the DDoS attack the entropy will be different, and the $NGain$ can reflect the change of the features.

Proof 2: Let's assume that the entropy of the flow feature is $e(ff)$ before the DDoS attack, then after the attack, the entropy

of the flow length $e_t(ff)$ is given as in Eq. (5) below:

$$e_t(ff = m) = - \sum_{i=m}^{\ominus} e(ff = m) c_i^m p^m (1-p)^{i-1} \log_2 P_i \quad (5)$$

where m represents flow length, c_i^m is a constant, and P_i is a priori probability. When the SDN network suffers from DDoS attack, the features including the flow length, the flow duration, the flow size and the flow rate will be increased. Then, the m value in Eq. (5) will be increased too, and the $Gain$ value becomes bigger. The $NGain$ value will also be increased. So, we can judge whether the SDN is attacked or not by a DDoS through the $NGain$ value which can reflect the change of the features.

In order to verify theorem 1, we use Mininet [18] which is a network emulator tool, however, the Open Virtual Switch (OVS) [19] was used for the network switches. Mininet natively runs on a Linux machine running Ubuntu OS. We consider the floodlight as the OpenFlow system.

Figure 2 shows the $NGain$ values for different features in 8-time intervals, where each three times is considered as a time interval. We can see that the $NGain$ values are different in the four features and have little fluctuation in the 8-time intervals. From figure 2(b) and (d), we can find that the fluctuation is low, and the flow rate has higher $NGain$ value after the DDoS attack than the flow length. At the same time, we can find that the fluctuation in Figure 2(a) and (c) is larger than the fluctuation in figure 2(b) and (d), and the flow size has higher $NGain$ value than the flow length.

IV. DDoS DETECTION ALGORITHM IN SDN

A. DDoS DETECTION ALGORITHM BASED ON DEGREE OF ATTACK

Based on the above-mentioned traffic analysis after the DDoS attack, and in order to introduce an efficient DDoS detection in an SDN network, in this section, we give out related definitions in details.

Definition 1: Degree of DDoS attack in SDN, suppose that we have the existing four features $f1$, $f2$, $f3$, and $f4$. Then, $N(f1)$ refers to the $NGain$ value of the feature $f1$. At the same time, $N(f2)$, $N(f3)$ and $N(f4)$ represent the $NGain$ values of the features $f2$, $f3$, and $f4$ respectively. Then, we define the degree of the DDoS attack (D) in the SDN as expressed by the following Eq. (6):

$$D = \frac{1}{n} \sum_{i=1}^n N(fi) \quad (6)$$

where: n equals to 4 in this paper.

Definition 2: Suppose a flow is defined as F , meanwhile, F_t denotes the $NGain$ of the flow during the time t . In order to detect the DDoS attack in the SDN, the formula to compute this measure is given as in the following Eq. (7):

$$F_t = \begin{cases} 0, & D \leq 0.5; \text{ DDoS attack is not detected} \\ 1, & D > 0.5; \text{ DDoS attack is detected} \end{cases} \quad (7)$$

If $D > 0.5$, we judge that the flow is an attack flow during the time t , and the SDN network suffers from the DDoS

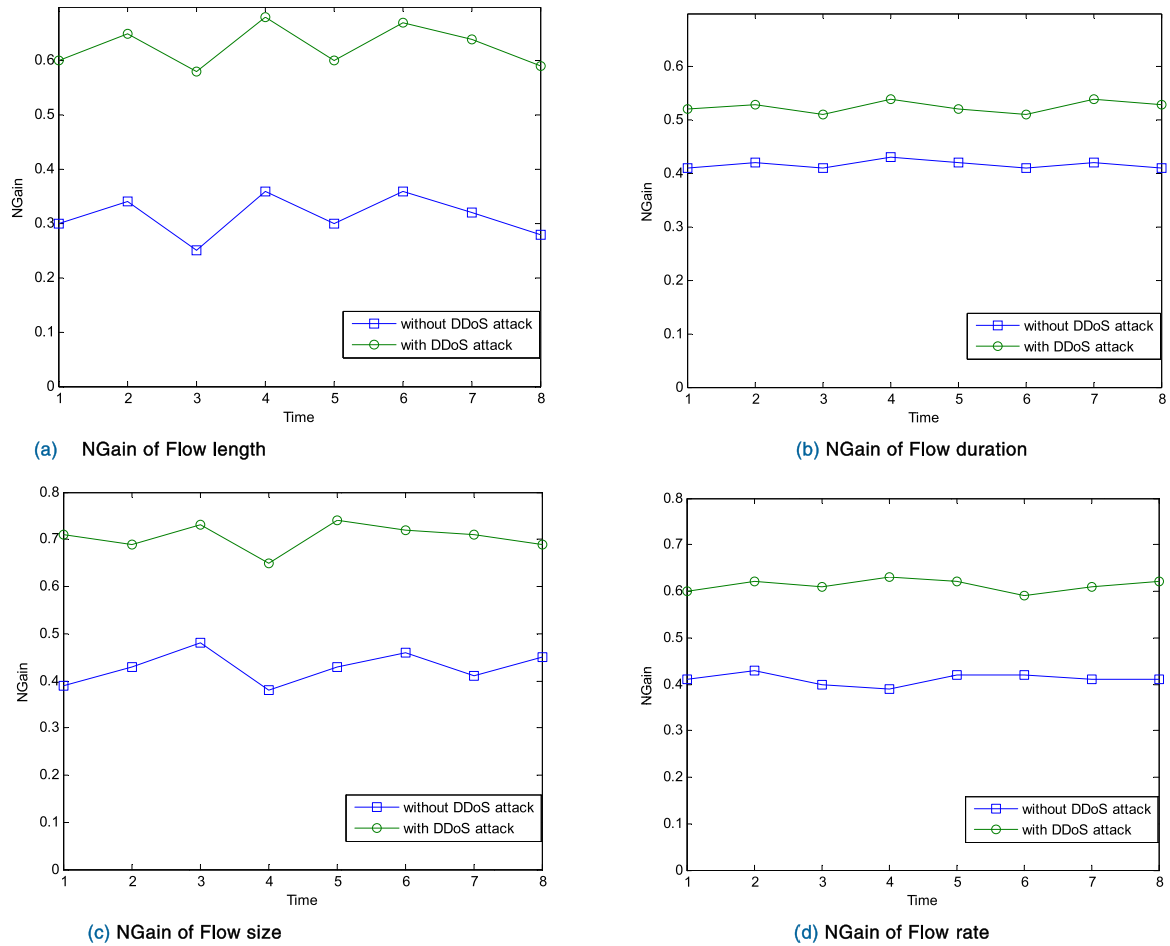


FIGURE 2. NGain values for different features.

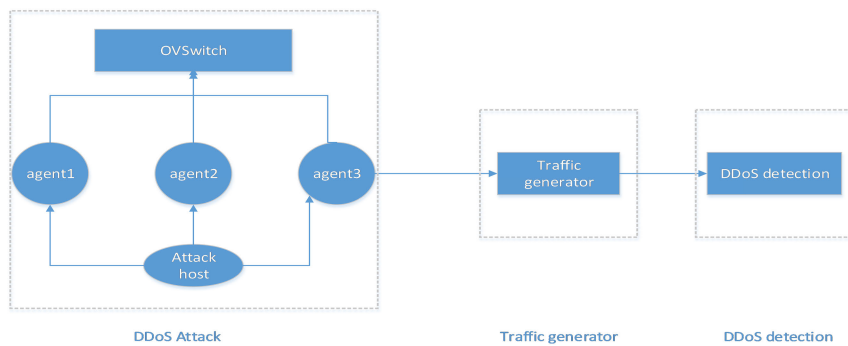


FIGURE 3. DDoS attack generation and detection.

attack. Otherwise, if $D \leq 0.5$, we judge that the flow is not an attack flow during the time t .

Figure 3 gives out the whole DDoS attack generation and detection. Firstly, the procedure of the DDoS attack is composed of the following three parts: the attack host, the agent, and the OV-Switch. The attacking host makes the task of an attack be sent to the agents 1, 2 and 3 which launch the DDoS attack to the OV-Switch part in the SDN. Then,

the followed two steps are the traffic generation and the DDoS detection. The traffic generator collects the network traffic which is obtained in different t intervals. The last step is to make the DDoS detection judge the flow according to Eq. (7).

Finally, in this section, the pseudo code of our algorithm which is called the DDoS Detection Algorithm based on the Degree of Attack (DDADA) is shown in Algorithm 1.

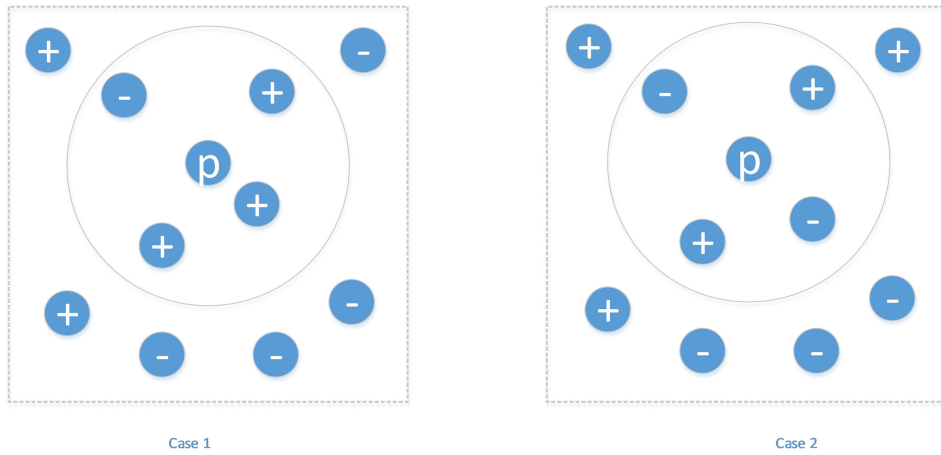


FIGURE 4. Two cases for KNN.

Algorithm 1 DDADA Algorithm

```

1 Begin
2 if t ∈ T then
3   Calculate D using Eq. (6);
4   if D > 0.5
5     network is attacked by DDoS;
6   else
7     network is normal;
8   end if
9 end if
10 End
    
```

B. DDoS DETECTION ALGORITHM BASED ON MACHINE LEARNING

In order to further improve our research on the DDoS detection method, in this section, we propose another algorithm to detect the DDoS in the SDN environment. The methods for detecting attacks based on Machine Learning (ML) have been widely used in the traditional networks [20]–[24]. However, few researches on the DDoS detection algorithms based on ML are done. Therefore, in this section, we propose an identification algorithm based on the improved K-Nearest Neighbors (KNN). The KNN is an efficient lazy learning algorithm and it has been successfully developed in many applications. Suppose all the flows as one Euclidean space R^n . We assume the flow x as a vector expressed by $\langle f_1(x), f_2(x), \dots, f_n(x) \rangle$. Where $f_m(x)$ represents the m -th feature value of the flow x . Now, let us define the distance of the flow x_i and x_j (i.e., $d(x_i, x_j)$) as the following mathematical formula:

$$d(x_i, x_j) = \sqrt{\sum_{m=1}^n (a_m(x_i) - a_m(x_j))^2} \tag{8}$$

Suppose that $f(x_p)$ is the final identification result. Then, we define $f(x_p)$ as follows,

$$f(x_p) \leftarrow \arg \max_{v \in V} \sum_{i=1}^k D(v, f(x_i)) \tag{9}$$

where $f(x_i)$ refers to the result value of the flow x_i , and v is in the range $[0, 1]$. For example, if $k = 4, f(x_1) = 0, f(x_2) = 1, f(x_3) = 1$ and $f(x_4) = 1$, then $f(x_p) = 1$

But when $f(x_1) = 0, f(x_2) = 1, f(x_3) = 0$ and $f(x_4) = 1$, then $f(x_p)$ will not get the result as it has been shown in Figure 4 (case 2). In order to resolve this problem, we will introduce the weight w which is defined by Eq. (10) as follows:

$$w = \frac{1}{d(x_p, x_i)} \tag{10}$$

Then, we can further express Eq. (9) as follows:

$$f(x_p) \leftarrow \arg \max_{v \in V} \sum_{i=1}^k w D(v, f(x_i)) \tag{11}$$

As it can be seen from Eq. (10), the weight w can resolve the problem which is shown in Figure 4 (case 2). If x_p and x_i are the same, then $d(x_p, x_i) = 0, \lim_{d \rightarrow 0} w \rightarrow \infty$, and w cannot be better computed. So, we can define $x_p = x_i$ when $d(x_p, x_i) = 0$. And the mathematical formula for $f(x_p)$ can be written as in the following formula (12):

$$f(x_p) = \begin{cases} f(x_i), & d(x_p, x_i) = 0 \\ \arg \max_{v \in V} \sum_{i=1}^k w D(v, f(x_i)), & otherwise \end{cases} \tag{12}$$

In order to better improve the weight w , we re-define w as follows,

$$w = \frac{1}{e^t} = \frac{1}{e^{d(x_p, x_i)}} \tag{13}$$

where: $t = d(x_p, x_i)$. Thus, the above-mentioned problem cannot be better resolved.

In order to evaluate our improved weight efficient method, let us suppose that the weight w in Eq. (10) is defined as w_1 , and the weight w in Eq. (13) is defined as w_2 . From Gauss distribution which is shown in Figure 5, we can notice that

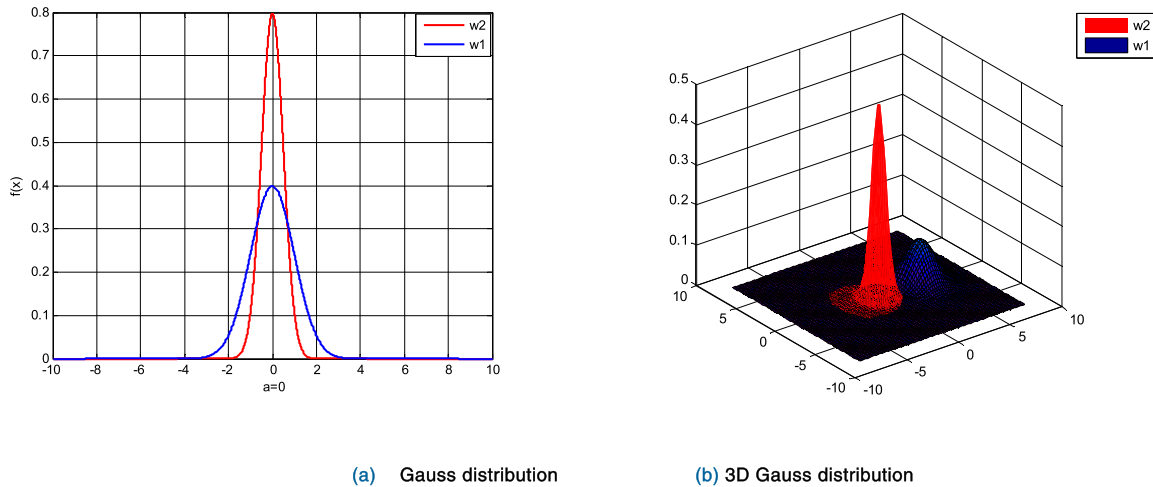


FIGURE 5. Two Gauss distribution for KNN.

the weight parameter w_2 can reflect the function better than the weight parameter w_1 .

Algorithm 2 DDAML Algorithm

Input: Training_samples

Output: Class label

1 **Begin**

2 **if** flow x_p in training_samples **then**

3 **for** $i = 0, i++, i < n$

4 compute $d(x_p, x_i)$;

5 **if** $x_p == x_i$ **then**

6 $f(x_p) = f(x_i)$

7 **else**

8 **according to Eq. (6) and (13)**

9 $f(x_p) = \arg \max_{v \in V} \sum_{i=1}^k wD(v, f(x_i))$

10 **return** $f(x_p)$

11 **end if**

12 **end for**

13 **end if**

14 **End**

Our proposed algorithm to resolve this detection problem of the DDoS attack is called the DDoS Detection Algorithm based on Machine Learning (DDAML). The pseudo code of this algorithm is shown in Algorithm 2. In this algorithm, we notice that x_p is the detection object which is in the flow set, and n is the number of the flow.

V. ALGORITHM COMPARISON AND PERFORMANCE EVALUATION

This section presents the performance evaluation and the experimental results of the DDoS detection in the SDN.

To simulate the proposed algorithms, we have implemented our experiments over a Floodlight which is an OpenFlow controller [25]. Also, we have built a small network

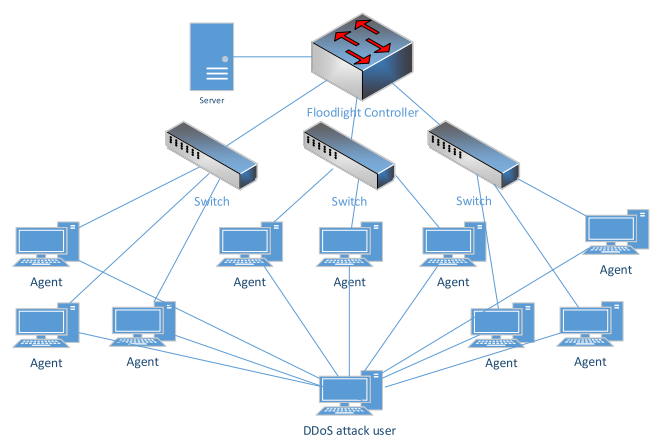


FIGURE 6. Simulation network topology.

TABLE 2. Experimental data.

Type of attack	Data
TCP flood	1000
UDP flood	300
ICMP flood	100

topology which consists of one server and ten clients (including nine agents and one DDoS user) as shown in Figure 6.

We have simulated the DDoS attack and collected the traffic flow from the traffic generator as it is shown in Figure 3.

A. EXPERIMENTAL DATA

The collected network traffic is composed of UDP, TCP, and ICMP, and the percentages of the traffic over these protocols are 70%, 20%, and 10% respectively. The DDoS traffic is generated by hping3, which can produce TCP, UDP, and ICMP flood attack traffics. The experimental data is shown in Table 2.

TABLE 3. Detection evaluation results.

Method	Evaluation				
	TPR	FPR	Precision	Recall	F-measure
NB	0.982	0.024	0.981	0.982	0.9815
KNN	0.985	0.018	0.983	0.985	0.984
SVM	0.985	0.018	0.9835	0.985	0.9842
CIC-SVM	0.986	0.017	0.9846	0.986	0.9853
DDADA	0.987	0.016	0.985	0.987	0.986
DDAML	0.994	0.009	0.993	0.994	0.9935

B. MEASUREMENTS EVALUATION

In order to evaluate the detection performance of the DDoS attack, we have introduced the appropriate measurements of the DDoS attack detection performance in this sub-section. These measurements are expressed as follows,

$$TPR = \frac{TP}{TP + FN} \tag{14}$$

$$FPR = \frac{FP}{FP + TN} \tag{15}$$

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

$$Recall = TPR = \frac{TP}{TP + FN} \tag{17}$$

$$F - measure = \frac{2 * Recall * Precision}{Recall + Precision} \tag{18}$$

C. PERFORMANCE EVALUATION AND RESULTS

In this paper, we mainly evaluate the performance of our results from the following aspects of the experiments: the accuracy of detection, the Receiver Operating Characteristic (ROC), and the Area Under the ROC Curve (AUC). Table 3 presents the comparison results of our two proposed DDADA and DDAML algorithms with the NB algorithm [26], the KNN algorithm [27], the SVM algorithm [7], and the CIC_SVM algorithm [8]. From Table 3, we can note that the TPR values for our DDADA and DDAML algorithms are 0.987 and 0.994 respectively which are higher than the TPR values of the other compared algorithms. Also, as we can see in Table 3, the FPR values for our DDADA and DDAML algorithms are 0.016 and 0.009 respectively which are lower than the FPR values of the other compared algorithms, which show that the error classifications in our algorithms are less than that in the other algorithms. Meanwhile, the Precision, the Recall and the F-measure of our algorithms are higher than those in the other algorithms.

Therefore, we can now conclude the performance evaluation of the previous results in the following aspects:

(1) Accuracy of detection

As an important parameter factor to evaluate the detection efficiency, the accuracy of detection analysis reflects

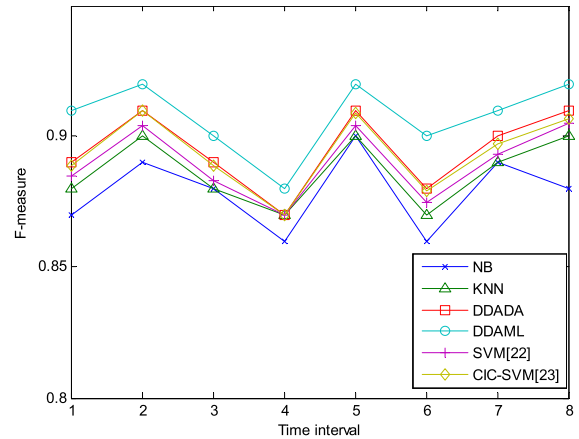


FIGURE 7. Comparison of F-measure in different time intervals.

the important indicator of the detection performance. The experimental results are shown in Figure 7.

It can be seen from Figure 7 that with different time intervals, the F-measures of the six compared algorithms are different. However, it can be seen clearly that our DDADA and DDAML algorithms have the best performance compared with the other NB, KNN, SVM, and CIC_SVM algorithms. This indicates that the DDADA and DDAML algorithms proposed in this paper own the best detection accuracy.

(2) ROC and AUC curves

In statistics, a Receiver Operating Characteristic curve (ROC curve) is a graphical plot that illustrates the performance of a binary classifier system as long as its discrimination thresholds vary. The curve is created by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. The true-positive rate is also known as the sensitivity, the recall, or the probability of the detection [28] in machine learning. The false-positive rate is also known as the fall-out or the probability of the false alarm [28], and it can be calculated as (1 – specificity). Thus, the ROC curve is the sensitivity as a function of the fall-out. In general, if the probability distributions for both the detection and the false alarm are known, then the ROC curve can be generated by plotting the cumulative distribution function

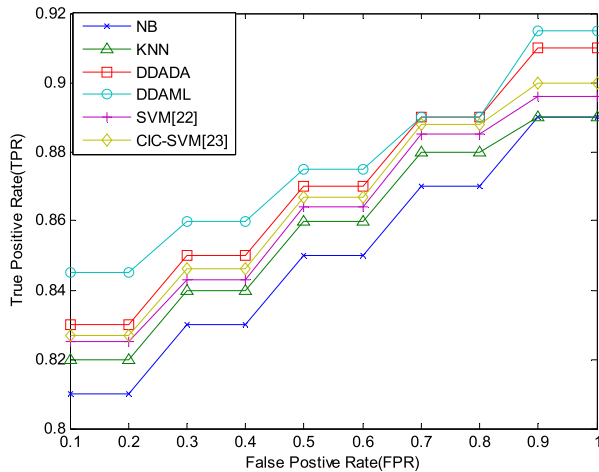


FIGURE 8. Comparison of ROC curves.

(i. e., the area under the probability distribution from $-\infty$ to the discrimination threshold) of the detection probability in the y-axis versus the cumulative distribution function of the false-alarm probability in the x-axis. The accuracy is measured by the Area Under the ROC Curve and it is called AUC. The level of this area is defined as follows,

$$AUC_l = \begin{cases} 0.9 \sim 1, & \text{excellent} \\ 0.8 \sim 0.9, & \text{good} \\ 0.7 \sim 0.8, & \text{middle} \\ 0.6 \sim 0.7, & \text{poor} \end{cases} \quad (19)$$

where: AUC_l represents the level of the AUC.

The ROC curves of the six compared algorithms in this paper for the DDoS detection are shown in Figure 8.

It can be seen from Figure 8 that the ROC curves vary for different algorithms due to the changing of the FPR and the TPR settings. The ROC curve of the DDAML algorithm outperforms all the other algorithms for the FPR and the TPR settings, while the NB and the KNN algorithms cover a small area of the ROC curve for the FPR and the TPR settings. The AUC values are shown in Table 4. The AUC value of the DDAML algorithm is 0.912, denoting excellent predication. As well, the NB, the SVM, the CIC-SVM, and the DDADA algorithms have AUC values of 0.891, 0.893, 0.895, and 0.899 respectively, which means that they have good predications too.

Thus, it can be observed from the above simulation experiments that our proposed DDADA and DDAML algorithms in this paper own better performance than the other traditional algorithms in the field of DDoS attack detection.

VI. CONCLUSION

The DDoS attack is currently the most serious threat to network security in the SDN network. The detection of the DDoS attack is critical to the defence against the DDoS attack. The recent DDoS attack detection methods still have low accuracy of identification and they are vulnerable to other factors.

TABLE 4. AUC values for the DDoS detection.

Method	AUC
NB	0.891
KNN	0.852
SVM	0.893
CIC-SVM	0.895
DDADA	0.899
DDAML	0.912

To address the above problems, we have completed the following achievements: Firstly, our proposed four features (i. e., flow length, flow duration, flow size, and flow ratio) are analysed when the SDN controller is attacked by the DDoS attack. Secondly, for the first time, a new concept called the degree of attack is proposed and presented to detect the DDoS attack. Based on this concept, a detection algorithm called DDADA algorithm is proposed. In addition, in order to further improve the detection efficiency, another detection algorithm called DDAML algorithm is introduced to identify the DDoS attack. The experimental results show that our proposed algorithms can identify the DDoS attack better, and they have achieved higher detection rates compared with the existing solutions. Finally, the experimental results indicate that the DDAML algorithm can outperform the other algorithms on different measurements of performance. In our future work, we will further improve the DDADA and the DDAML algorithms in order to apply them into the real SDN environment.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments on improving this article.

REFERENCES

- [1] S. Shin and G. Gu, "Attacking software-defined networks: A first feasibility study," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, 2013, pp. 165–166.
- [2] P. Fonseca, R. Bennesby, E. Mota, and A. Passito, "A replication component for resilient OpenFlow-based networking," in *Proc. IEEE Netw. Oper. Manage. Symp.*, Apr. 2012, pp. 933–939.
- [3] G. Yao, J. Bi, and L. Guo, "On the cascading failures of multi-controllers in software defined networks," in *Proc. 21st IEEE Int. Conf. Netw. Protocols (ICNP)*, Oct. 2013, pp. 1–2.
- [4] H. Lin and P. Wang, "Implementation of an SDN-based security defense mechanism against DDoS attacks," in *Proc. Joint Int. Conf. Econ. Manage. Eng. (ICEME), Int. Conf. Econ. Bus. Manage. (EBM)*, Philadelphia, PA, USA, 2016.
- [5] J. G. Yang, X. T. Wang, and L. Q. Liu, "Based on traffic and IP entropy characteristics of DDoS attack detection method," *Appl. Res. Comput.*, vol. 33, no. 4, pp. 1145–1149, 2016.
- [6] A. Saied, R. E. Overill, and T. Radzik, "Detection of known and unknown DDoS attacks using artificial neural networks," *Neurocomputing*, vol. 172, pp. 385–393, Jan. 2016.
- [7] J. Ye, X. Cheng, and J. Zhu, "A DDoS attack detection method based on SVM in software defined network," *Secur. Commun. Netw.*, vol. 2018, Apr. 2018, Art. no. 9804061.
- [8] J. Cui, M. Wang, and Y. Luo, "DDoS detection and defense mechanism based on cognitive-inspired computing in SDN," *Future Gener. Comput. Syst.*, vol. 97, pp. 275–283, Aug. 2019.

- [9] Y. Yu, L. Guo, Y. Liu, J. Zheng, and Y. Zong, "An efficient SDN-based DDoS attack detection and rapid response platform in vehicular networks," *IEEE Access*, vol. 6, pp. 44570–44579, 2018.
- [10] M. V. De Assis, M. P. Novaes, C. B. Zerbini, L. F. Carvalho, T. Abrao, and M. L. Proenca, "Fast defense system against attacks in software defined networks," *IEEE Access*, vol. 6, pp. 69620–69639, 2018.
- [11] Y. Wang, T. Hu, G. Tang, J. Xie, and J. Lu, "SGS: Safe-guard scheme for protecting control plane against DDoS attacks in software-defined networking," *IEEE Access*, vol. 7, pp. 34699–34710, 2019.
- [12] S. Dong, K. Abbas, and R. Jain, "A survey on distributed denial of service (DDoS) attacks in SDN and cloud computing environments," *IEEE Access*, vol. 7, pp. 80813–80828, 2019.
- [13] J. Liu, Y. Lai, and S. Zhang, "FL-GUARD: A detection and defense system for DDoS attack in SDN," in *Proc. Int. Conf. Cryptogr., Secur. Privacy*, 2017, pp. 107–111.
- [14] H. Xie, T. Tsou, D. Lopez, H. Yin, and V. Gurbani, *Use Cases for Alto With Software Defined Networks*, document Internet-Draft draft-xie-alto-sdn-extension-use-cases-01, Working Draft, IETF Secretariat, 2012.
- [15] S. M. Specht and R. B. Lee, "Distributed denial of service: Taxonomies of attacks, tools, and countermeasures," in *Proc. ISCA PDCS*, 2004, pp. 543–550.
- [16] A. Studer and A. Perrig, "The core melt attack," in *Proc. Eur. Symp. Res. Comput. Secur. (ESORICS)*, Saint Malo, France, Sep. 2009.
- [17] M. S. Roodposhti, J. Aryal, H. Shahabi, and T. Safarrad, "Fuzzy Shannon entropy: A hybrid GIS-based landslide susceptibility mapping method," *Entropy*, vol. 18, no. 10, p. 343, Sep. 2016.
- [18] (2014). *Mininet*. [Online]. Available: <http://mininet.org>
- [19] (2014). *Open Vswitch*. [Online]. Available: <http://openvswitch.org>
- [20] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "Intrusion detection by machine learning: A review," *Expert Syst. Appl.*, vol. 36, no. 10, pp. 11994–12000, 2009.
- [21] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, and K. Dai, "An efficient intrusion detection system based on support vector machines and gradually feature removal method," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 424–430, Jan. 2012.
- [22] W. Feng, Q. Zhang, G. Hu, and J. X. Huang, "Mining network data for intrusion detection through combining SVMs with ant colony networks," *Future Gener. Comput. Syst.*, vol. 37, pp. 127–140, Jul. 2014.
- [23] F. Kuang, W. Xu, and S. Zhang, "A novel hybrid KPCA and SVM with GA model for intrusion detection," *Appl. Soft Comput.*, vol. 18, pp. 178–184, May 2014.
- [24] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors," *Knowl.-Based Syst.*, vol. 78, pp. 13–21, Apr. 2015.
- [25] *Floodlight*. Accessed: 2011. [Online]. Available: <http://floodlight.openflowhub.org>
- [26] E. Frank, L. Trigg, and G. Holmes, "Technical note: Naive Bayes for regression," *Mach. Learn.*, vol. 41, no. 1, pp. 5–25, 2000.
- [27] D. T. Larose, "K-nearest neighbor algorithm," in *Discovering Knowledge in Data: An Introduction to Data Mining*, 1st ed. Hoboken, NJ, USA: Wiley, 2004, pp. 90–106.
- [28] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.



SHI DONG received the M.S. degree in computer science from the University of Electronic and Technology of China, in 2009, and the Ph.D. degree in computer science from Southeast University. He was a Postdoctoral Researcher with the Huazhong University of Science and Technology. He is a Visiting Scholar with Washington University in St. Louis. He is currently a Distinguished Professor with Zhoukou Normal University. His current research interests include network management and network security.



MUDAR SAREM received the B.S. degree in electronic engineering from Tishreen University, Lattakia, Syria, in 1989, and the M.S. and Ph.D. degrees in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 1997 and 2002, respectively. He worked as an Associate Professor with the School of Software Engineering, Huazhong University of Science and Technology, from 2005 to 2009 and a Visiting Professor, from 2010 to 2018. He is currently a Main Researcher with the General Organization of Remote Sensing (GORS), Damascus, Syria. He has published over 80 articles in refereed conferences and journals. His current research interests include image processing, computer networks, and distributed systems.

...