JST: Smart Systems and Devices
Volume 33, Issue 2, May 2023, 001-008

Two-Layers DDoS Attack Detection Model Using Machine Learning in Software Defined Networking

Nguyen Ngoc Tuan**, Nguyen Huu Thanh
Hanoi University of Science and Technology, Ha Noi, Vietnam
**Corresponding author email: tuan.ncs16077@sis.hust.edu.vn*

Abstract

Software Defined Network (SDN) is a new architecture designed to make the network infrastructure more flexible and easier to manage. It allows network administrators to configure network parameters as well as integrating new functions using programming languages easily. Thanks to the centralized control paradigm, it is easier to collect information of the entire network, which facilitates the implementation of machine learning algorithms to detect anomaly traffic as well as network attacks. Recently, with the development of machine learning and artificial intelligence, several methods have been applied to detect and mitigate Distributed Denial of Service (DDoS) attacks. However, all of activities from monitoring data, detecting and mitigating the attack consume time and resources. To reduce unnecessary redundancy, in this paper, we divide attack detection into two phases, which are anomaly detection phase with lightweight machine learning algorithm and attack detection phase when anomaly behaviors have been detected. This reduces the in-depth analysis of normal traffic and helps to improve the use of computing resources and data transmission efficiency of the network. By setting up a testbed, we have successfully run this model as well as evaluated the accuracy of the model. The results show that our model can detect attacks quickly and accurately.

Keywords: DDoS detection, SDN, security, machine learning.

1. Introduction

Digital transformation makes more and more devices connected to the network and changes almost all aspects of life, study, work and play. Therefore, network security is increasingly concerned and deployed by organizations and enterprises. Currently, there are many types of attacks on the network such as reconnaissance attacks, password attacks, phishing attacks, denial of service attacks, etc. The most popular and influential are Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks [1]. It not only affects the attacked server, but also the intermediary devices and the network system, especially in SDN [2] architecture where the control plane separates from the data plane. The SDN architecture is illustrated in Fig. 1. OpenFlow is the de facto protocol that SDN controller uses to communicate with SDN switches. It is the first and also the most widely known SDN protocol for southbound API to provide communication between the control plane in SDN controller and the data plane in OpenFlow switch. The OpenFlow protocol is layered above the Transmission Control Protocol, leveraging the use of Transport Layer Security (TLS). Operations on route discovery and policy enforcement are concentrated on the controller. Hence, when there is a large amount of data due to a denial of service attack, in addition to affecting the connection links, it also affects the devices as controller considerably. In this paper, we focus on a solution to detect DDoS attack as the first step for further actions, such as DDoS classification and mitigation.

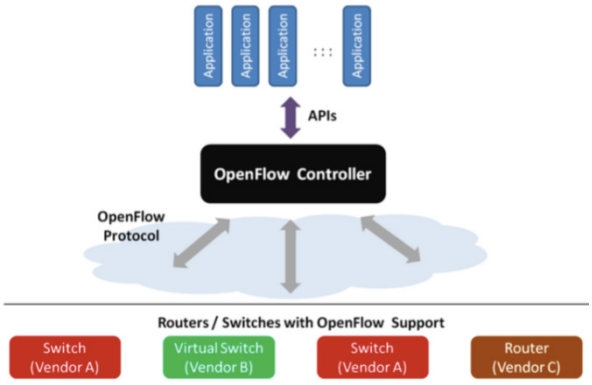


Fig. 1. Software-Defined Network Architecture

In DoS attack, an attacker sends an enormous number of requests to the victim. Victims with limited resources are overwhelmed and unable to provide services to legitimate users. The resources include bandwidth, computational resources or Operating System data structure. A DoS attack is called a DDoS attack when the origins of the attack come from many different hosts. This large number of host attackers can be a group of people agreed upon a specific attack date and time, for example, the infamous *Anonymous* group. It can also be just a single person having control over a large number of compromised hosts. In the latter case, the network of compromised hosts is called

ISSN: 2734-9373
https://doi.org/10.51316/jst.166.ssad.2023.33.2.1
Received: February 15, 2023; accepted: March 6, 2023

botnet. To create this network, the attacker must distribute malicious software by different means to the Internet. Unaware users through several means accidentally are infected and put under control of the attacker. Each host in the botnet is called zombie. Each zombie can carry out any command. Several known types of DDoS attack are divided into three groups, as shown in Fig. 2.

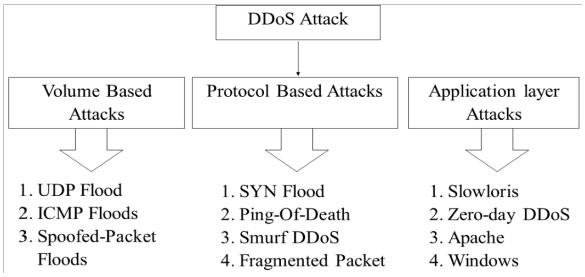


Fig. 2. DDoS attacks classification

In volume-based attack, attackers send a large amount of traffic to the target using a form of amplification or other means of generating large amount of traffic, such as requests from botnets to clog the victim's connection bandwidth. A good example of this attack is the ICMP flood attack [2].

In protocol-based attack, attackers often use weaknesses in the third layer and fourth layer of the protocol stack to send a large number of connection requests. It causes the victim to state exhaustion and causes a service discruption, e.g. TCP-SYN flood [2].

In application layer attack, attackers exploit weaknesses in the seventh layer of the protocol stack to establish a connection with the victim, and then exhaust its resources by monopolizing processes and transactions.

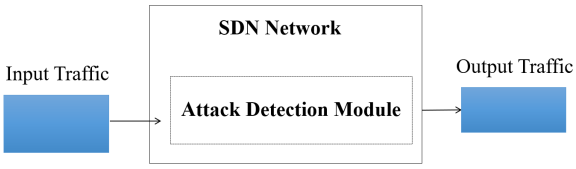
Of the three types of attack above, the volume-based attacks and protocol-based attacks are quite similar as they both send a large amount of traffic to the victim. The application layer attack creates a connection to the application and then takes over processes and transactions, without generating much traffic to the victim. This makes the detection of volume-based and protocol-based attacks easier and deployable in intermediary devices along data transmission line.

Although there are many types of DDoS attack, in this paper we especially study TCP-SYN flood, ICMP flood, UDP flood attack, thereby proposing a new model to detect these attacks.

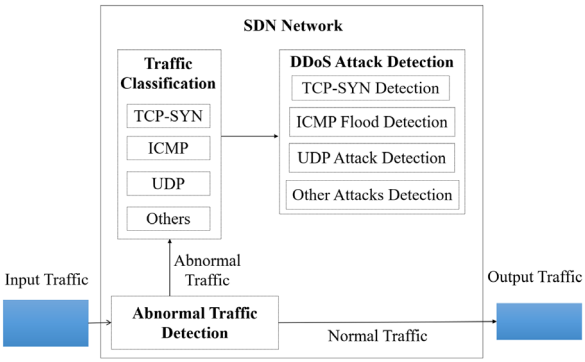
Recently, the application of machine learning algorithms to detect DDoS attacks in SDN network has been widely studied [3-12], but there are remaining limitations that need to be studied further, namely:

1) Since each DDoS type has specific properties, different machine learning detection algorithms should be deployed for each DDoS type.
2) Moreover, the detection algorithms should always monitor the incoming traffic, even there is no attacks most of the time. On the other hand, when an attack is detected, the security gateway may deploy the corresponding counter attack actions.

Thus running multiple traffic monitoring, attack detection and mitigation algorithms at the same time could be required. This poses serious issues on system performance as these complex algorithms usually have very high demand on system resource, not only memory but also computing power.



(a) Typical DDoS attack detection model



(b) Proposal DDoS attack detection model

Fig. 3. DDoS attack detection model

Fig. 3a is a typical deployed attack detection model. Only one attack detection module is deployed at the controller. The module integrates multiple detection algorithms running in parallel that process the incoming traffic.

Fig. 3b shows our proposed model, which pipelines all actions that should be taken in case attacks take place. The attack detection is divided into three modules. The first module detects anomaly traffic using lightweight machine learning algorithm to determine whether the traffic is abnormal or not. The second module further classifies the types of attacks, such as procotol-based or volumetric based on some simple behaviours of the attacks. The third module uses complex machine learning algorithms to detect and verify each type of attack, including TCP SYN, ICMP and UDP flood, etc.

The contributions of this work are as the follows.

1) Propose a new architecture of security gateway that piplines the process flow of incoming traffic, thus reducing the complexity of the DDoS detection process. The architecture composes of a

three-layer DDoS detection model, which makes use of machine learning algorithms in SDN networks.

2) For each module in the architecture, we select suitable type of attacks, such as TCP SYN, ICMP and UDP flood. The detection concept can also be extended to other types of attacks.

The rest of this paper is organized as follows. Section 2 addresses related work. Section 3 proposes a two-layers DDoS attack detection model. Section 4 discusses the simulation results and evaluations. Finally, Section 5 concludes the work.

2. Related Works

Until now, there has been various work in the area of DDoS attack detection using machine learning. In [3], a hybrid machine learning algorithm between Seft-Organizing-Map (SOM) and K-NearestNeighbor (KNN) was used to detect ICMP attack. SOM is used to cluster traffic into attack and normal groups in training phase. KNN is then used to assign labels for network status based on the label of *k* nearest neighbors. Another machine learning-based solution proposes detection methods by determining the time of request between hosts [4], getting request time, number of source host and number of destination host, and using different algorithms (Naive Bayes, KNN, *K*-means) to classify traffic into normal and attack. Using the entropy method to determine the randomness of the flow data, Ref. [5] presents a novel solution for the early detection and mitigation of TCP SYN flooding. The entropy information includes destination IP and few attributes of TCP flags. It is implemented as an extension module in Floodlight and it is evaluated under different conditional scenarios. Hu *et al*. [6] proposed an efficient and lightweight framework to detect and mitigate DDoS attacks in SDN by using entropy of features and an SVM algorithm. Firstly, the network traffic information is collected through the SDN controller and sFlow agents. Then an entropy-based method is used to measure network features, and the SVM classifier is applied to identify network anomalies. Another approach using SVM algorithm was presented in [7]. The paper mostly focuses on anomaly traffic detection based on the entropy of IP source addresses and ports by integrating SVM into the Ryu controller. The performance of the system is relied on Mininet that can hardly be evaluated in real-time.

In terms of abnormal traffic detection, Song *et al*. [8] proposed a real-time anomaly traffic detection method based on dynamic KNN cumulative-distance anomaly detection algorithm. The authors presented the design and implementation of the method by leveraging STROM, a distributed steam computing technology. Beside *k*-NN, Lohit Barki *et al*. used more different machine learning algorithms such as Naive Bayes, *K*-means and Kmedoids to classify the traffic as normal and abnomal as presented in [9]. The performance of the algorithms is evaluated using parameters such as detection rate and efficiency. While using machine learning-based methods for anomalous detection, these algorithms appear to be complex and are used to detect only specific attack types. Zhu *et al*. [10] argued that the traditional intrusion detection system (IDS) is based on pattern recognition which only can be used for well-known network attack behavior. For that reason, machine learning classifiers is used in the work to distinguish abnormal network traffic from the normal traffic background. Decision tree and KNN algorithm are also used in this research.

In another work [11], a DDoS detection scheme based on Support Vector Machine (SVM) in SDN was proposed. By extracting flow table information features in SDN, data is derived and the data model of DDoS traffic can be trained, then DDoS abnormal traffic identification is realized. We found that the implementation of SVM in SDN is more coputational intensive than some other approaches, which will be mentioned later in this paper. In [12], SVM algorithm was also used to detect and mitigate DDoS attack. In this paper, the authors proposed an SDN framework to identify and defend DDoS attacks based on machine learning. This framework consists of three parts which are traffic collection module, DDoS attack identification module and flow table delivery module. The traffic collection module extracts traffic characteristics to prepare for traffic identification. By utilizing the flexible and multi-dimensional features of SDN network architecture in deploying DDoS attack detection system, the controller extracts the network traffic characteristics through statistical flow table information and uses SVM method to identify the attacked traffic. Then the flow table delivery module dynamically adjusts the forwarding policy to resist DDoS attacks according to the traffic identification result. KDD99 dataset is used to train and test of the approach.

3. Two Layers DDoS Attack Detection Model

In this section, we introduce the proposal model using in SDN as well as abnormal traffic detection, traffic classification and attack detection algorithms.

3.1. Proposal Model

We propose a new DDoS attack model as Fig. 4. It illustrates two-layers DDoS detection model. In this model, there are three blocks to detect the DDoS attack. The first is anomaly detection block using Local Outlier Factor (LoF) algorithm. The second is classification block using traffic type after recognizing abnormal traffic. The last is specific attack detection using DNN, KNN algorithms and so on. In this model, traffic is filtered simply first and then filter deeply using specific algorithm. It detects DDoS attack more exactly and quickly.

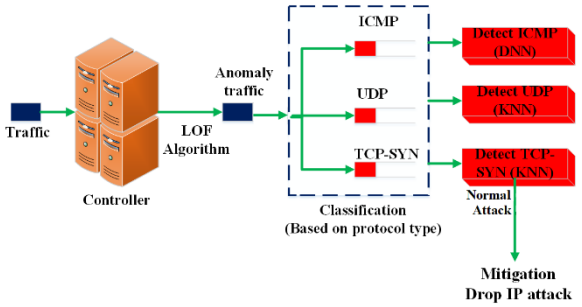


Fig. 4. Two layers DDoS detection model

Local outlier factor (LoF) is an algorithm proposed by Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng and Jrg Sander in 2000 for finding anomalous data points by measuring the local deviation of a given data point with respect to its neighbors. The purpose of LoF algorithm to find the number of abnormal points in data.

Basic idea of LoF is based on a concept of a local density. The locality is given by k nearest neighbors that their distance is used to estimate the density. By comparing the local density of a point to its neighbors, we can identify the abnormal points that have lower density than their neighbors.

K-Nearest-Neighbor (KNN) is one of the simplest among all machine learning algorithms yet it works very well especially in low dimensional feature space. KNN is a non-parametric method. Which means it assumes nothing about the distribution of the data samples. In machine learning, KNN is instance-based learning which is a kind of lazy learning that defers all generalization attempt until a query is made. Instance-based learning algorithm performs the generalization by comparing new instances with training instances. The training instances for KNN are feature vectors in a multidimensional feature space. Each vector has an explicit label. The training phase just consists of storing all vectors and their corresponding labels.

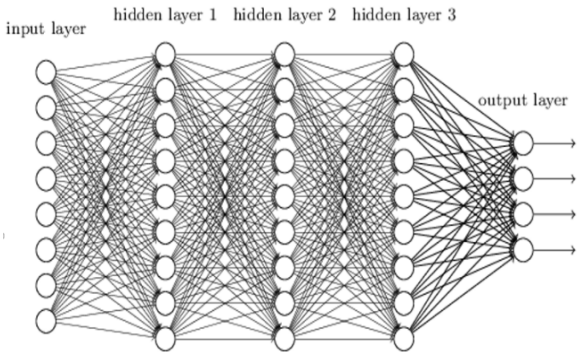


Fig. 5. Deep Neural Network Architecture

Deep Neural Network (DNN) is a deep learning algorithm, a type of machine learning. Architecture and activity of deep learning algorithm is modeling as brain. It is also called artificial neural network.

Fig. 5 is the model of DNN algorithm. There are three components: input layer, hidden layer and output layer. The input layer receives input data. The hidden layer filters and tunes data so that it reaches to the result. The output layer presents the result of predication. The complex of DNN depends on the number of hidden layers and the number of nodes in each layer.

3.2. Anomaly Detection Process

In the anomaly detection block, there are a few features affected when DDoS attack begins. In our works, we choose five most importance features and give them to the input of machine learning algorithms.

The five features are *source IP address*, *source port address*, *destination port address*, *packet type* and *total packet*.

Since these features are not synchronous types and values, so we need to preprocess these features to make normalized data before applying algorithms. Data preprocessing cleans noise inside the dataset to make it much cleaner using various techniques. It will be much easier and faster for the smart algorithms to handle a clean dataset rather than a dataset filled with broken instances.

There are two phases in this process. The first phase is to collect data from SDN switch every five seconds. The second phase is to create normalized dataset from these raw data.

The changes of five features in attack phase are quite clear. For example, the number of source IP, source port and total packet increases very fast while destination port and packet type only focus on one or several values. This information need to reduce the amount of data and calculate the randomness without much loss information. Entropy is one of the most famous methods to do that as the higher randomness, the higher entropy value. For instance, when network status is normal, the number of source IP less than attack status. Therefore, entropy value is lower. The formula of entropy equation is given below:

$$H = -\sum_{i=1}^{n} P_i \log(P_i) \quad (1)$$

P_i: probability of occurrence of the i^{th} IP. In each phase of this article, we use some or all of the representative features for classification purposes. These classification's features are described as follows:

- Entropy of source IP source: average number of source IP in attack phase is very high to compare normal phase;
- Entropy of source port: average number of port source in attack phase is very high to compare normal phase;
- Entropy of destination port: average number of

destination port in attack phase is very high to compare normal phase;

- Entropy of packet type: average number of packet type in attack phase is very high to compare normal phase;
- Entropy of total packet: average number of total packet in attack phase is very high to compare normal phase.

We apply above formula for 4 features: IP source, port source, port destination and packet type. Finally these features need to be normalized to put values into the same range. In particular, a L-dimentional vector $(d_1, d_2,..., d_L)$ is normalized into $(n_1, n_2,..., n_L)$ by the following formula:

$$n_i = \frac{1}{2}\left\{\tanh\left(0.1\left(\frac{d_i - \mu}{\sigma}\right)\right) + 1\right\} \quad (2)$$

where i = 1, 2, ..., n; μ and σ are the mean and the standard deviation, respectively, of the i^{th} dimension in the training set. At the end of the normalization process, the calculated means and standard deviation values for each dimension are saved for later use in real-time operation.

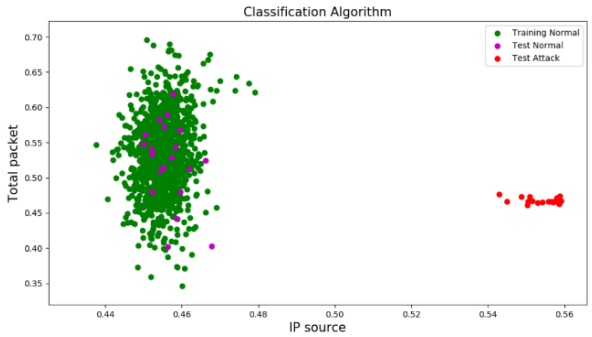


Fig. 6. Data after classification using LoF

The mission of SDN switch focuses on detecting DDoS attack as soon as possible, so it always run abnormal traffic detection algorithm. Due to that, the number of calculation is as small as possible. We propose a different applying method of Local Outlier Factor (LoF) algorithm in the mission of detecting abnormal traffic. Instead of using a set of data to find out abnormal points, we collect some normal data from switch (in several days) and using these data to the training data of LoF algorithm. After the training phase, running algorithm with input data is the network status in each five seconds and labeling these inputs as normal or attack based on the threshold from training phase. Fig. 6 shows result of LoF after classification. Green points are normal training data (9000 points) while purple and red points are test data (20 normal points and 20 attack points). After classification, if the test point is labeled as normal, it will be painted purple color, and red color in other cases.

Algorithm 1. LoF

Input: $W_i = (w_i^1, w_i^2, \dots, w_i^p)$, real-time instance $V = (v_i^1, v_i^2, \dots, v_i^p)$, $K_{th\_nearest\_neighbors} = 11$

// Where i = 1,2,…,L, L is the number of example in input data isAnomaly = **False**

for i = 1 to L **do**:

compute reachability distance of Wi

compute local reachability density of Wi

compute LoF of Wi

endfor

Determine max of all LoF value and assign it to M Foreach V in real-time, compute LoF(V)

If LoF (V) > M **then**:

isAnomaly = **True**

else:

isAnomaly = **False**

endif

output isAnomaly

3.3. Classification Traffic

As presinting in Internet Assigned Numbers Authority (IANA) website, in the Internet Protocol version 4 (IPv4) there is a field called "Protocol" to identify the next level protocol. This is an 8 bit field. In Internet Protocol version 6 (IPv6), this field is called the "Next Header" field. The value of this field is used to recognize TCP, ICMP, UDP and other traffic. In the SDN network, this value of packet in is collected and storaged in flow table.

In this proposal, after the traffic is detected as anormaly, it needs to be classified into particular traffic types. We simplify this phase by using the threshold with counting the number of packets in attribute protocol type field of flow table. We initiate threshold based on maximum values of the number of packet corresponding to each particular packet type (ICMP, TCP SYN, UDP). After that, these threshold is updated when an attack takes place. If the packet type we have counted is over the threshold, we label these traffic corresponding to the attack type and change the system to specific attack detection phase. By this method, we can label many attack types at the same time.

3.4. Specific Attack Detection

As discussing above, KNN is a supervised learning algorithm. It is one of the most simple machine learning algorithms. The data is used for training includes both input data and label in the output. KNN will not learn anything until it determines the label of an input data point. So, the data be stored in the entire training material, which in most case is

extremely large. In addition, KNN could have to compute to each data point in the training set, this is going to take a lot of time as well as system resources, especially with databases of multidimensional space or a huge number of data points. As the output is indicated based on the nearest data points, when K is small, KNN is easy to make false predictions if interference occurs. In conclusions, KNN is only productive in mathematical problems where the distinctness between the inputs is adequately big and thereby creating a difference in outcomes.

In the TCP-SYN attack, the number of packet and packet size does not increase as much as ICMP attack, on the other hand, the number of flows raises dramatically. So, KNN algorithm is appropriate for TCP-SYN attack detection.

For DNN algorithm, unlike other machine learning algorithms, it does not use user-entered parameters manually. DNN is going to find out the most optional and necessary features for detecting and blocking attacks. For example, by predicting sensibility, the seven parameters are considered to have the most influence on the output, still, we can not be sure which parameters are more important than which ones. In addition, if the measurement and calculation are executed on all seven parameters, we will usually need very large computing resources and it is going to take a long time. DNN will solve this problem by learning by itself and finding the optimal parameters for the calculation and prediction of the output. Moreover, DNN calculates and learns in training process for a long period of time, nevertheless, the results can be predicted and delivered rapidly and accurately.

In the case of DDoS attacks by ICMP or UDP packet, the attacker typically increases the larger number of packet with large size. KNN algorithm is not suitable for using to detect. So we use DNN algorithm in these cases. Fig. 7 shows the result of DNN algorithm after classification.

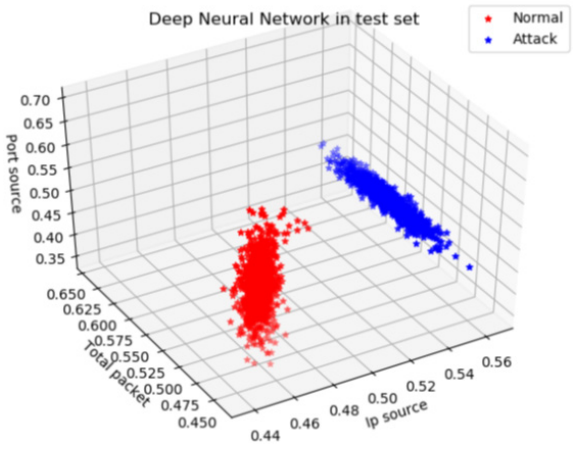


Fig. 7. Classification ICMP attack using DNN

Algorithm 2. Deep Neural Network

Input: $W_i = (w_i^1, w_i^1, \dots, w_i^p)$, real-time instance $V = (v_i^1, v_i^2, \dots, v_i^p)$, Hidden layer h = 5

- Number of nodes in each hidden layer: 5, *Learning_rate = 0.0075*

- Number of nodes in output layer: 2

- Activation in hidden layer: ReLU

- Activation function in output layer: softmax

// Where i = 1,2,…,L, L is the number of example in input data isAttack = **False**

iteration = 5000

while iter < 5000 :

for i = 1 to h **do**:

compute forward propagation in layer i

Save Z and A matrix of layer i

if i == 2 **then**:

Compute loss function

Save loss function

endif

endfor

for i = h to 1 **do**:

Compute gradient Descent for layer i

Save dW and db

Update W and b with formula:

*W = W - learning_rate *dW*

*b = b - learning_rate *db*

endfor

set threshold for $\hat{y}$

If $\hat{y}$ > 0.5 **then**:

$\hat{y}$ = 1

else:

isAttack = **False**

endif

endwhile

output :isAttack

Algorithm 3. K-nearest-neighbor

Input: $W_i = (w_i^1, w_i^1, \dots, w_i^p)$, real-time instance $V = (v_i^1, v_i^2, \dots, v_i^p)$,

// Where i = 1,2,…,L, L is the number of example in input data K=3

isAttack = **False**

for i = 1 to L **do**:

compute distance from of Wi to V

endfor

- Sort $d(W_i, V)$ and choose 3 nearest neighbors of V

- Count label of 3 nearest neighbors and vote the class for V depend on the label of 3 neighbors

If Label(V) == 1 **then**:

isAttack = **True**

else:

isAttack = **False**

endif

output :isAttack

4. Testbed and Performance Evaluation

Fig. 8 presents our testbed model. We use traffic generator to replay attack traffic CAIDA dataset 2007. Attack traffic types include ICMP, TCP SYN and UDP. The traffic goes into SDN switches and attacks to victim. All modules are performed in POX controller. There are 4 modules in POX: Statistic, Anomaly Detection, Classification and Specific attack detection.

Dataset is used include CAIDA, normal data from system in Future Internet lab and data from Ministry of Public Security.

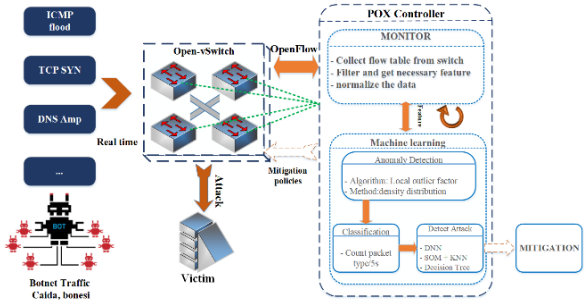


Fig. 8. Testbed model

To evaluate effectively of LoF, we use two parameters, which are Accuracy and Recall displayed in Fig. 9. The higher k^{th} nearest neighbors are, the higher accuracy of LoF is. The reason we use recall parameter is that the number of positive class is much less than negative class. The formula for Recall is:

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

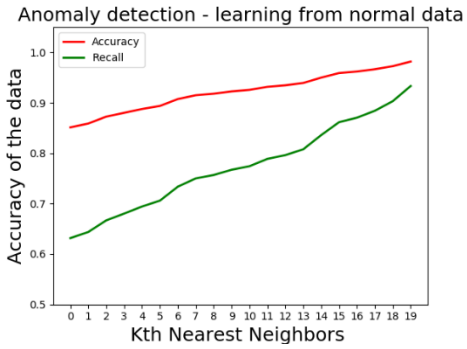


Fig. 9. Accuracy of LoF with parameter k

The best advantage of our LoF algorithm is the time processing in real-time. Table 1 compares the processing time between LoF and SVM oneclass. LoF is up to 100 times faster than SVM oneclass. This guarantees that SDN controller can run this algorithm all the time without consuming much resources system. Hence, we choose LoF for the first step of detection.

Table 1. Compare processing time between oand SVM Oneclass

Number of Instance	SVM oneclass	LoF
100	$4.872 * 10^{-3}$	$4.386 * 10^{-5}$
200	$5.248 * 10^{-3}$	$4.506 * 10^{-5}$
300	$5.589 * 10^{-3}$	$5.388 * 10^{-5}$
400	$5.961 * 10^{-3}$	$5.879 * 10^{-5}$
500	$6.712 * 10^{-3}$	$6.536 * 10^{-5}$

The main purpose of the DNN algorithm is to minimize the loss function of DNN (the loss function is used to evaluate the difference between the true label and predicted label). Fig. 10 shows the value of loss function (cost) when increasing the iteration. We can easily see that the cost value will reduce when we increase iteration. Moreover, the cost value is quite stable when iteration is greater than or equal to 100.

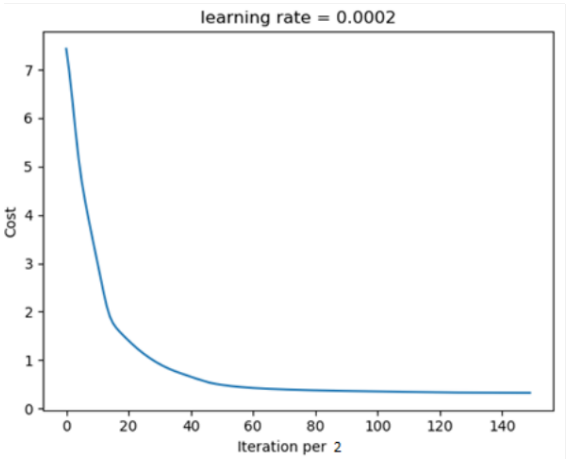


Fig. 10. Loss function value after iterations

After evaluating cost function, we need to consider the accuracy of DNN when increasing iteration. Fig. 11 shows the value of accuracy of DNN algorithm when iteration is increased. When the iteration is greater than 100, the accuracy is quite stable and approximately up to 1.

Table 2 is comparison results of four popular machine algorithms, namely DNN, KNN, SVM and Decision Tree when they are applied to detect ICMP and TCP-SYN attack. The results show that the accuracy of DNN and KNN is better.

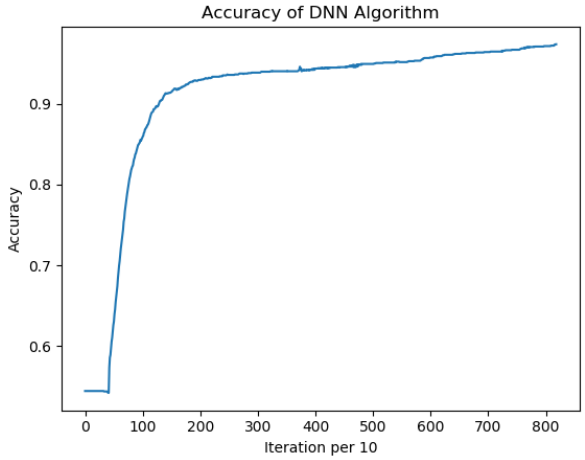


Fig. 11. Accuracy of DNN algorithm

Table 2. Comparison the results of four Algorithms in detecting ICMP and TCP-SYN attack

Parameter	DNN	KNN	SVM	Decision Tree
Training Time (s)	0.5542	0.0009	0.0407	0.0024
Prediction Time (s)	0.0003	0.0022	0.0077	0.0001
Accuracy (%)	99.906	97.790	94.568	97.737
Precision (%)	100.00	96.130	100.00	97.737
Recall (%)	99.794	100.00	88.037	98.140
F1-Score	99.897	98.027	93.638	97.938

5. Conclusion

This paper presents a new model of detecting DDoS attack using machine learning in SDN network. The attack detection part is divided into two modules. The first is anomaly detection module with lighweight machine learning algorithm. The second is specific attack detection module for each type of attack as ICMP, TCP-SYN and so on. The results of testbed show that the choice of algorithms LoF, DNN and KNN is reasonable and the results of detection are quick and exact.

In the next research, we will continue tesing more machine learning algorithms to choose the suitable algorithm for each type of DDoS attack.

References

[1] T. Mahjabin, Y. Xiao, G. Sun, W. Jiang, A survey of distributed denial-of-service attack, prevention, and mitigation techniques, International Journal of Distributed Sensor Networks, Dec. 2017, https://doi.org/10.1177/1550147717741463.

[2] Software-Define Networking: The new norm for networking, Open Networking Foundation, Apr. 2012. [Online]. Available: https://opennetworking.org/sdn-resources/whitepapers/software-defined-networking-the-new-norm-for-networks.

[3] T.M. Nam, P. H. Phong, T. D. Khoa, T. T. Huong, P. N. Nam, N. H. Thanh, Self-organizing map-based approaches in ddos flooding detection using SDN, In Proceedings of the 2018 International Conference on Information Networking, Chiang Mai, Thailand, Jan. 2018, https://doi.org/10.1109/ICOIN.2018.8343119.

[4] S. Hameed, H, A. Khan, SDN based collaborative scheme for mitigation of ddos attacks, Future Internet, vol. 10, no. 3, Feb. 2018, https://doi.org/10.3390/fi10030023

[5] P. Kumar, M. Tripathi, A. Nehra, M. Conti, C. Lal, SAFETY: Early detection and mitigation of TCP SYN flood utilizing entropy in SDN, IEEE Trans. Network and Service Management, vol. 15, no. 4, pp. 1545-1559, Dec. 2018, https://doi.org/10.1109/TNSM.2018.2861741

[6] D. Hu, P. Hong, Y. Chen, FADM: DDoS flooding attack detection and mitigation system in software-defined networking, IEEE Global Communications Conference, Singapore, Dec. 4-8, 2017, https://doi.org/10.1109/GLOCOM.2017.8254023

[7] L. Yang, H. Zhao, DDos attack identification and defense using SDN based on machine learning method, International Symposium on Pervasive Systems, Algorithms and Networks (I-SPAN), Yichang, China, Oct. 2018, https://doi.org/10.1109/I-SPAN.2018.00036

[8] R. Song and F. Liu, Real-time anomaly traffic monitoring based on dynamic k-NN cumulative-distance abnormal detection algorithm, IEEE 3rd International Conference on Cloud Computing and Intelligence Systems, Shenzhen, Nov. 27-29, 2014, https://doi.org/10.1109/CCIS.2014.7175727

[9] L. Barki, A. Shidling, N. Meti, D. G. Narayan, M. M. Mulla, Detection of distributed denial of service attacks in software defined networks, International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, India, Sep. 21-24, 2016, https://doi.org/10.1109/ICACCI.2016.7732445.

[10] M. Zhu, N. Guo, Abnormal network traffic detection based on semi-supervised machine learning, Computer Science, DEStech Transactions on Engineering and Technology Research, Feb. 14, 2018, https://doi.org/10.12783/dtetr/ecame2017/18466

[11] Z. Ma, J. Huang, Research on DDoS abnormal traffic detection under SDN network, In: H. Shen, Y. Sang, Parallel Architectures, Algorithms and Programming. PAAP 2019, Communications in Computer and Information Science, Springer, vol. 1163, pp. 368-379, 2020, https://doi.org/10.1007/978-981-15-2767-8_33

[12] L. Yang, H. Zhao, DDoS attack identification and defense using SDN based on machine learning method, 15th International Symposium on Pervasive Systems, Algorithms and Networks (I-SPAN), Yichang, China, Oct. 2018, https://doi.org/10.1109/I-SPAN.2018.00036