Received: 21 October 2017 | Revised: 14 November 2017 | Accepted: 15 November 2017

DOI: 10.1002/dac.3497

RESEARCH ARTICLE

WILEY

Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN

Chuanhuang Li[1] | Yan Wu[1] | Xiaoyong Yuan[2] | Zhengjun Sun[1] | Weiming Wang[1] | Xiaolin Li[2] | Liang Gong[1]

[1]School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou, Zheiang 310000, China

[2]Large-scale Intelligent Systems Laboratory, University of Florida, Gainesville, FL 32611, USA

Correspondence
Chuanhuang Li, School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou, Zhejiang 310018, China.
Email: chuanhuang_li@zjgsu.edu.cn

Funding information
the National High Technology Research and Development Program (863) of China, Grant/Award Number: No.2015AA011901; the Zhejiang's Key Project of Research and Development Plan, Grant/Award Number: 2017C03058; the Zhejiang Provincial Natural Science Foundation of China, Grant/Award Number: LY18F010006; the National Natural Science Foundation of China, Grant/Award Number: No.61379120 No.61402408

Summary

Distributed denial of service (DDoS) is a special form of denial of service attack. In this paper, a DDoS detection model and defense system based on deep learning in Software-Defined Network (SDN) environment are introduced. The model can learn patterns from sequences of network traffic and trace network attack activities in a historical manner. By using the defense system based on the model, the DDoS attack traffic can be effectively cleaned in Software-Defined Network. The experimental results demonstrate the much better performance of our model compared with conventional machine learning ways. It also reduces the degree of dependence on environment, simplifies the real-time update of detection system, and decreases the difficulty of upgrading or changing detection strategy.

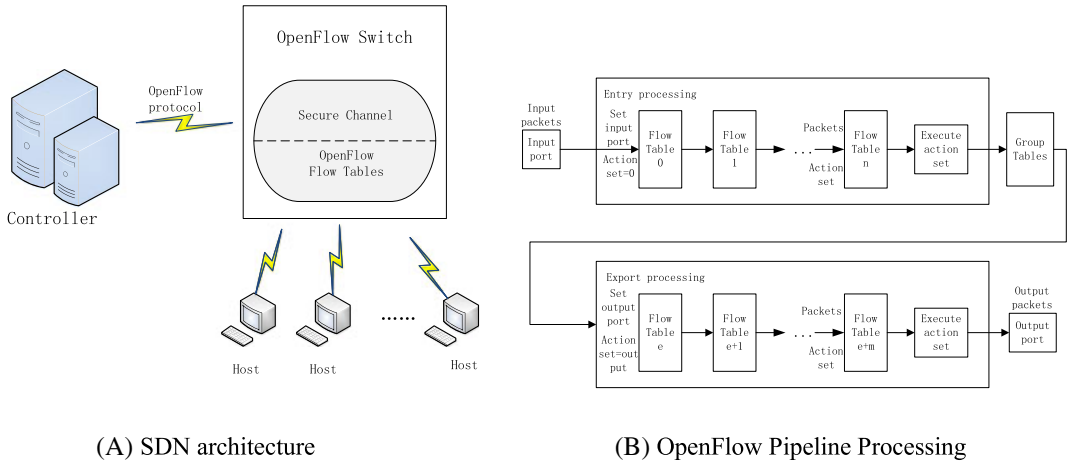
KEYWORDS

DDoS defense, DDoS detection, deep learning, distributed denial of service, Software-Defined Network

1 | INTRODUCTION

The OpenFlow-based Software-Defined Network (SDN) architecture consists of 3 parts, the controller, the OpenFlow switch, and the host.[1] Among them, the controller is the "brain" of the entire SDN network. It is responsible for the generation, delivery, and maintenance of the network forwarding flow table. When generating a flow table, it provides coordination for the convection rule builder. When the network state changes, it generates a new flow rule and updates the flow table in switches. A controller can interact with multiple switches using the same OpenFlow protocol at the same time, and optimize the allocation of tunnel coordination flow rules and select route directly over switches and other functions. The OpenFlow switch is another core component of the SDN and mainly responsible for routing and forwarding data packets. The SDN architecture is shown in Figure 1A.

A flow entry in a flow table defined in the OpenFlow1.5 protocol contains 7 components: the Matching Field, the Priority, the Counters, the Instructions, the Timeouts, the Cookie, and the Flag Bit.[1] The OpenFlow switch contains an OpenFlow Pipeline made up of a series of ordered flow tables; the ordinal number of the flow tables starts from 0.



(A) SDN architecture (B) OpenFlow Pipeline Processing

FIGURE 1 OpenFlow-based Software-Defined Network

OpenFlow Pipeline Processing defines the behavior of data packets interacting with flow tables; the process of Pipeline Processing is shown in Figure 1B. Pipeline Processing always starts from the first flow table of the entry processing phase. The use of other entry flow tables depends on the matching result of the first flow table. If the packet matches the corresponding flow entry, then apply instructions set in this flow entry. This packet may be forwarded to other flow tables by the Goto-Table instruction. If the packet does not match any flow entry, and there is no table-miss flow entry, the packet will be dropped.

Distributed denial of service (DDoS) attack is a kind of cooperative attack model, which is more fragile and larger than general distributed attack. Attackers use many puppet machines they controlled to simultaneously launch denial of service (DoS) attacks on the target. Ultimately, the system resources or network bandwidth are exhausted or even collapsed. Since the first DDoS attack occurred in 1999,[2] DDoS has become a fatal, widespread, and rapidly evolving threat in the world. According to a survey from Radware, DDoS is currently the largest threat (50% respondents in the survey) for organizations.[3] Currently, main attack vectors include UDP flood, HTTP flood, SYN flood, ICMP, DNS, etc and pose serious threats to both systems and networks.[4]

At the beginning of the design of SDN architecture,[5] the security of SDN network has become one of the key research issues. In implementation of the defense architecture for DDoS attack based on OpenFlow,[6] OpenFlow switches are often used to collect network traffic and analyze the feature information of data message. Finally, matching rule with the recorded DDoS attack rule base[7] and using controller to complete the task of intrusion response, among them, in addition to complete the flow forwarding task, switches also need to do other tasks such as data protocol analysis and DDoS attack rule base matching. In addition to routine tasks such as maintenance, control, and forwarding, the controller is also required to complete feature collection of data packets of DDoS attacks, intrusion response,[8] and other special tasks. A lot of extra tasks of controllers and switches can add burdens to the already overburdened network devices. Moreover, protocol analysis and DDoS attack rule base matching increase the processing time of switches, so the transmission delay of forwarding the normal network data packets is raised. Eventually, it will add the processing delay of the entire network environment. Some approach can improve the performance of data forwarding,[9] but in the rule matching process of the DDoS attack rule base, the rule base is difficult to identify and update. Moreover, the feature value rule base of DDoS attack is invalid to detect the flooding DDoS attack.[10] Therefore, aim at the features of the SDN network architecture, building an efficient and reasonable security defense mechanism to ensure the normal information security of network users, is one of the key issues that must be considered carefully when designing and deploying the entire SDN network architecture.

This paper introduces deep learning technology into the field of SDN network security. The major contributions are the following:

1. We construct a deep learning model to detect DDoS attack. The model is composed of input layer, forward recursive layer, reverse recursive layer, fully connected hidden layer and output layer. Recurrent neural network (RNN),[11] long short-term memory (LSTM),[12] and convolutional neural network (CNN)[13] are also used in the model. The model is proved to have high detection accuracy through the experiment results.

2. We apply the DDoS attack detection deep learning model to OpenFlow-based SDN. The SDN controller can generate drop policy and issue to the switch, according to the detection result obtained from the model.
3. We implement the deep learning DDoS defender. The results of real-time DDoS attacks experiment verify that the defender can effectively detect and defend DDOS attacks.
4. The rest of this paper is structured as follows. Section 2 describes related work. Section 3 introduces the DDoS attack detection models based on deep learning. This section also gives the results of comparison between different models and algorithms. Section 4 presents the implementation of DDoS attack defense. Section 5 shows the experiment and result analysis. Conclusion is shown in Section 6.

2 | RELATED WORK

Nowadays, summing up the researches on defense technology for DDoS attacks, they are usually implemented based on traditional network intrusion detection.[14] With the development of DDoS attack technology, current research of DDoS attack defense is still grim. There are still high misdiagnosis rate and omissive judgment rate in network traffic processing and new DDoS attack detection.

When the network encounters a DDoS attack, the network data flow cleaning device is started, or the network, which is subjected to DDoS attack, is directly isolated. Finally, by cleaning the network traffic and isolating the attacking targets is to prevent the network environment from continuing to suffer from DDoS attacks and achieve the purpose of security defense. The process consists mainly of confirming the validity of the source IP address at the network forwarding device, filtering the data traffic that source IP address is not valid,[15] and establishing a mapping table between source address and access port in routing and forwarding devices. By comparing source IP address of the data traffic and the corresponding port address is to determine whether there are DDoS attacks in the network, then to clean the DDoS attack traffic.[16]

Fonseca et al[17] ever proposed, when SDN network suffers from DDoS attacks, that the switch will lose its connection to the upper controller, and then it will look for and connect to the standby controller. This method can be used to temporarily reduce the impact of DDoS attacks but cannot completely prevent the attacks. If the standby controller also suffers DDoS attacks, the entire SDN network will be out of control.

Kim et al[18] proposed a method of predicting normal traffic based on flow threshold. The method uses Cisco's NetFlow Technology, to detect network traffic by the detection function constructed by the extracted flow features and the set threshold. However, this method requires dealing with precollected data traffic, and it takes a lot of work. Moreover, the success or failure of the test is closely related to the practical experience of the researchers.

Huawei[19] proposed an anti-DDoS cloud cleaning scheme based on SDN, using big data analysis techniques. Fine-grained analysis of data traffic from more than 60 dimensions and implementation of SDN network security defense from the source of DDoS attacks, this method has heavy workload and has too much dependence on related hardware and software.

Gaoshang et al[20] recently proposed Flood Defender, a scalable and protocol-independent system to protect OpenFlow networks against SDN-aimed DoS attacks based on 3 novel techniques: table-miss engineering, packet filter, and flow table management. However, this system has to rely on neighbors of the victim. There must be enough neighbor switches to save bandwidth.

Because the intrusion technique becomes more and more complicated, the above DDoS attack detection and defense technology by using traditional methods respectively has the problem of low detection efficiency, error or leakage report, poor adaptability etc. The intrusion detection system based on machine learning can learn from existing intrusion behaviors and master some common characteristics. However, with the complexity of intrusion data and the diversity of features, it also becomes difficult to detect the attacks by using the general machine learning methods. Because of the strong learning ability of deep learning, it is possible to apply it to DDoS attack detection.

Braga et al[10] proposed a lightweight DDoS attack detection mechanism. By the use of SDN traffic statistics function, he extracts 6 tuple fields of DDoS attack features, using self-organizing mapping of neural network algorithm, to identify attack stream. QuamarNiyaz et al[21] also proposed a deep learning–based multivector DDoS detection system in SDN. But these methods are all hard to detect low-rate attack, because it looks similar to the legitimate network traffic from the victim end. Meanwhile, DDoS attack toward victim systems must be generated over time; otherwise, it will not be malicious to the network/system resources. Our detection approach utilizes a sequence of continuous network packets and is able to learn the subtle difference between attack traffic and legitimate one. It helps to find repeated patterns representing DDoS attacks and locate them in a long term traffic sequence.

3 | DDOS ATTACK DETECTION BASED ON DEEP LEARNING

3.1 | Deep learning model and algorithm

Deep learning has created a new beginning for solving the limitations of machine learning. Deep learning algorithm is a process of feature learning, which can discover multilevel features and represent high-level features into more abstract data features. At present, deep learning has been widely used in speech recognition, computer vision, natural language processing, and other fields by many companies, such as Google,[22] Facebook,[23] Microsoft,[24] Baidu,[25] and so on. Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. It discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer.[26,27]

The most classical neural network models applied in deep learning are CNN,[13] RNN,[11] and LSTM.[12]

The CNN is a feedforward artificial neural network. It is mainly based on 3 basic ideas, local receptive field, weight sharing, and pooling. Local receptive field maps each neuron to local features, thereby reducing the weight parameters needed to be trained. Weight sharing ensures that all neurons in the same convolution kernel have the same weights, so the training parameters in the network can be greatly reduced. The size of the features can be reduced by pooling, and to ensure the invariance of features. Thus, the robustness of the input features in displacement, tilt, scaling, or other deformations can be guaranteed by CNN. The CNN model is shown in Figure 2A.

In convolution layer, the feature of the previous layer is convolution with a convolution kernel. The result through activation function is expressed as feature of current layer x_j^l and output:

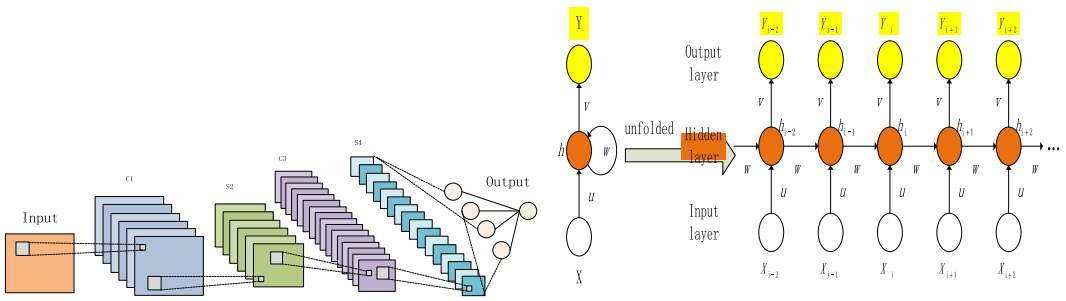
$$X_j^l = f\left(\sum_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j^l\right), \tag{1}$$

where we denote l as the number of layers in the network model, k as the number of convolution kernels, M_j as input features, and b as the offset for each output. Sampling the input data in the subsampling layer, if the number of input features is n, the number of features after sampling in subsampling layer is still n in theory and output:

$$Y_j^l = f\left(\beta_j down\left(x_l^{l-1}\right) + b_j^l\right), \tag{2}$$

where we denote β as the weight, *down(*)* as the subsampling function, and * as the input data of the subsampling function. The subsampling function is usually a sum of size $n*n$ of the input features, so the output size should be $^1/_n$ of the input size, and each output feature has independent β and b.

The RNN is a neural network about spatial depth. In Feedforward Neural Networks (FNN), the information is transmitted from the input layer to the output layer, while RNN breaks the constraint of the directional transmission of information in FNN. The RNN is different from the FNN: The FNN is the input layer and the output layer, both layers are all connected before and after, and there is no connection between the neurons in each layer; while the connections



(A) Convolution Neural Network model (B) Recurrent Neural Network model

FIGURE 2 Deep learning models

between the neurons in the hidden layer of RNN are exiting, the input of neurons at a certain moment in the layer includes the input of the input layer neuron, the input of other neurons in the same layer, and the output of the neuron at the previous moment. The RNN model is shown in Figure 2B.

3.2 | Packets feature processing

In finding a set of optimal value of data feature, which is highly dependent on the relevant researchers' experience by using traditional machine learning methods, the restrictions in search of fixed training feature values are much lower to deep learning, which is a big advantage of deep learning compared to other machine learning methods in terms of feature expression. Deep computation model can be used here; it can effectively improve the efficiency for training parameters.[28] The DDoS attack detection method used in this paper needs to process the data packets before inputting the packets to the network model, including feature extraction, format conversion, and dimensional reconstruction of the input packets. The processing is shown in Figure 3.

Twenty segments are extracted from the input m packets, as the feature field. The 20 feature fields are classified into 3 types: text fields, numerical fields, and Boolean fields. We convert Boolean fields to binary and make it as input data format. Among them, when TCP, UDP, and other Boolean-type feature fields are entering into the deep learning network model, convert them into binary values as input data format; we change TCP, UDP, HTTP, and other port numbers to a binary list of 16 bits, which is used to store binary values such as conversion of TCP, UDP, HTTP, and other port numbers. For text fields such as Frame.Protocols, we transform them using Bag of Word (BoW, lexical hypothesis) method, which is used as input data format. For numerical fields such as Tcp.Len and Udp.Len, we subtract mean value and normalize all data in each field, and then as input data formats.

We get a two-dimensional feature matrix of $m^{*}n'$ after feature transformation, where m indicates number of packets and n' indicates the number of new features after transformation. Then the matrix is cut by a series of consecutive time window of size T, and set the tag value y for each time window. The tag value y is 0, indicating that the packets in the time window are normal packets, and the tag value y is 1, indicating that the packets in the time window are DDoS attack packets. To reshape the features after cutting, a three-dimensional matrix of $(m-T)^{*}T^{*}n'$ is constructed to satisfy the input requirement of the deep learning network model.

3.3 | Overall detection model

The raw network data samples are processed by the feature processing units, we can obtain a sample data set satisfying the training conditions of a deep learning network. The next work is to train the deep network model, construct a deep learning network model, and finally, realize the detection of DDoS attacks in network, through the study of deep learning–related technologies, especially principle analysis for commonly used CNN, RNN, and LSTM neural network models of deep learning. On this basis, a bidirectional RNN model is proposed to detect DDoS attacks in network. The deep learning network model built in this paper is constructed successfully based on the deep learning technology and the open-source framework Keras. In the deep learning training model, 2 RNN layers are included in each direction; the

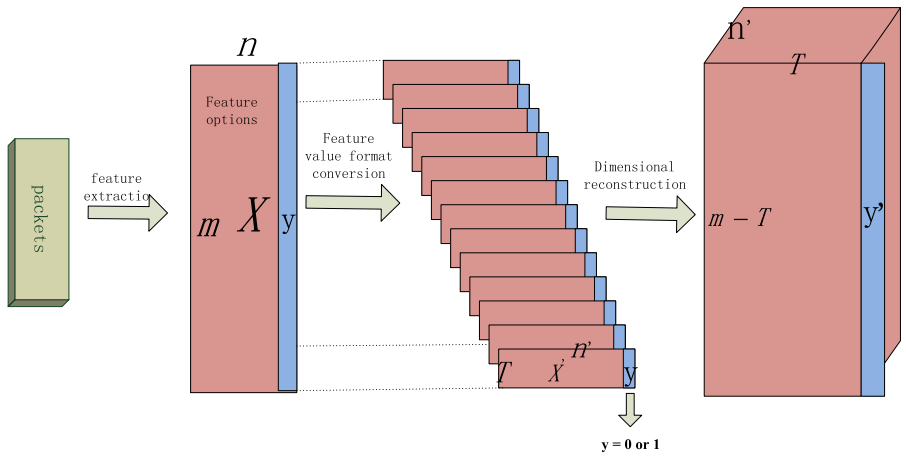


FIGURE 3 Flow chart of data packet processing

function of the recurrent layer is to track the previous time period and the historical features of network data packets. To input the processed features into the deep learning network model established in this paper to detect and determine whether the input packet is a DDoS attack packet, construct a deep learning network model with input layer, forward recursive layer, reverse recursive layer, fully connected hidden layer and output layer, as shown in Figure 4. The input data of the input layer are input to the Forward/Backward RNN layer at the same time, then the treatments of bidirectional RNN layer are combined and are input to the fully connected hidden layer; finally, fully connected hidden layer processes data then input processed data into the output layer to predict output. In the bidirectional RNN layer, adjacent neurons of the same layer are connected to each other in turn, that is, the output of the current moment of the neuron will be the input of adjacent neurons at the next time.

In the input layer, we adopt batch standardization to accelerate neuron network training. To obtain local information and simplify neural networks, the stack convolutional layer is used in the subsequent N layer. There are 128 neurons in each layer, the convolution kernel of the CNN is 5, the stride length is 1, and we use *tanh* as the nonlinear activation function. After the convolutional neural layer, the forward RNN layer and the backward RNN layer process the results simultaneously into the input layer of the fully connected hidden layer:

$$\mathrm{P}\left(y_t|\{x_d\}_{d\neq t}\right) = \varnothing\left(w_y^f h_t^f + w_y^b h_t^b + b_y\right), \tag{3}$$

$$h_t^f = \tanh\left(w_h^f h_{t-1}^f + w_x^f x_t + b_h^f\right), \tag{4}$$

$$h_t^b = \tanh\left(w_h^b h_{t+1}^b + w_x^b x_t + b_h^b\right), \tag{5}$$

where we denote y_t as output; $\{x_d\}_{d \neq t}$ as the input features; $\varnothing$ as nonlinear function, which is the softmax function in this paper; w_y^f as the weights between the hidden layer and the input layer in the forward RNN; w_h^f as the weights between the

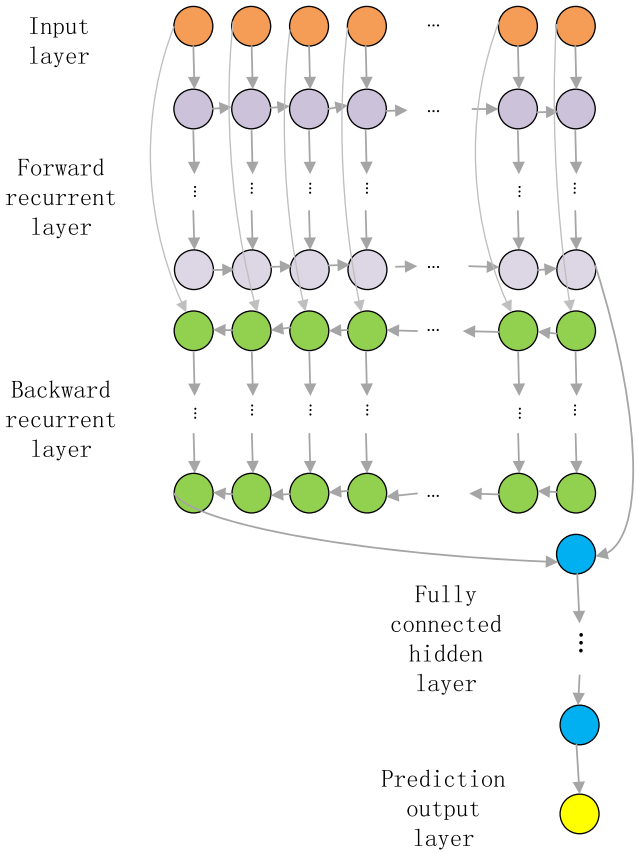


FIGURE 4 Overall deep learning model

hidden layer and the hidden layer in the forward RNN; w_x^f as the weights between the input layer and the hidden layer in the forward RNN; w_y^bas the weights between the hidden layer and the input layer in the backward RNN; w_h^b as the weights between the hidden layer and the hidden layer in the backward RNN; and w_x^b as the weights between the input layer and the hidden layer in the backward RNN. The result of the processing is merged at the input layer of the fully connected hidden layer. At the same time, to solve the gradient disappearance problem of deep RNN, we try to use LSTM and gated recurrent unit network (GRU).[29] The LSTM aims to overcome the gradient disappearance problem of deep RNN. Gated recurrent unit network is a simple variant of LSTM that can train standard LSTM; it is faster than LSTM because of fewer parameters. Above the recurrent layer, we construct the fully connected layer and set the Relu function to control the output of the entire neural network. At the output layer, we will obtain the attack of probability time window.

4 | DDOS ATTACK DEFENSE BASED ON DEEP LEARNING

4.1 | Defense architecture

For DDoS defense architecture based on deep learning in SDN, firstly, we build a deep learning DDoS defense architecture of Features Extraction, Deep Learning DDoS Detector, Model Updater, Information Statistics, and Flow Table Generator modules. Figure 5 illustrates the architecture.

Among them, the Feature Extraction module extracts features from all the packets of OpenFlow switch and constructs a feature matrix satisfying the input requirements of Deep Learning DDoS Detector module. The Deep Learning DDoS Detector module uses a trained deep learning model to learn the features after the Feature Extraction module processing. Detecting whether packets entered in the current OpenFlow switch are attack packets, the Information Statistics module extracts the features of the detected attack packets and collects statistics on the frequency of these features. According to the statistical results of the Information Statistics module, the Flow Table Generator module determines the flow entries and priorities for various attack packages that can be dropped and sends them to the OpenFlow switch. The Model Updater module automatically updates the deep learning model of the Deep Learning DDoS Detector module through the network. This will happen only when there is a new model on the server.

According to the features processed by the Feature Extraction module, the Deep Learning DDoS Detector module uses a trained deep learning model to detect whether packets entered in the current OpenFlow switch are attack packets. If so, the attack packet will be forwarded to the Information Statistics module for statistical purposes; otherwise, it will not be processed. The OpenFlow switches perform normal forwarding. The Information Statistics module performs feature statistics on all attack packets within the time interval T set by the administrators. According to the occurrence

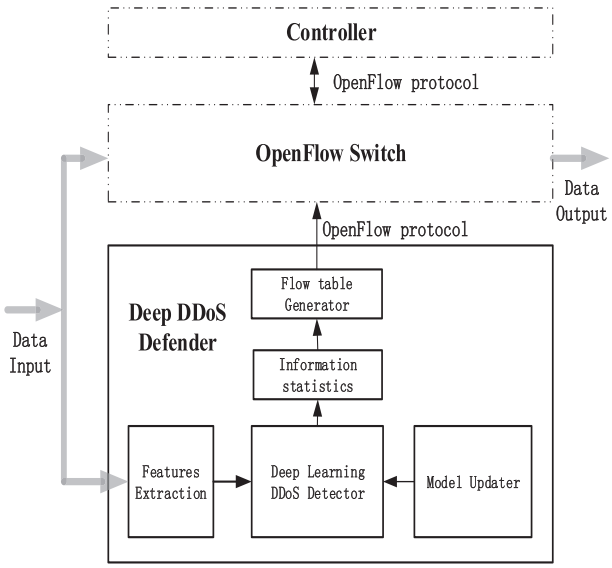


FIGURE 5 Distributed denial of service (DDoS) defense architecture based on deep learning

frequency of all features in attack packets, the weight {W} of features is determined. The related information of features is transmitted to the Flow Table Generator module; according to the weight {W} of corresponding features, it generates flow entries that can drop packets having such features and generates the priority of these flow entries, and sends them to the OpenFlow switch. The OpenFlow switch executes the flow table and drops the data packet that is currently entering the OpenFlow switch.

4.2 | Defense flow

The DDoS defense flow is illustrated as follows, in a statistical window (attack packet counter); each feature item is statistically analyzed. For each feature value, firstly, to calculate the total number of mean values of the feature, which is denoted as feature_len, and secondly, in the feature value of feature_len, according to the frequency of a specific feature type value in the total feature value feature_len, we implement sorting of frequency. Then, when the frequency of a value occurrence in the feature is not less than the set threshold Q(Q = 0.04), we keep this value; finally, the feature values are used to generate the OpenFlow flow entries based on the retained values. We need to perform internal statistics for specific feature value types and confirm the priority of the OpenFlow flow entries in the OpenFlow flow table generated by the specific feature value. According to the statistical analysis of a specific feature value type and the different final weights, to generate different priority flow entries, the specific method of determining the priority of the flow entries is, firstly, determine the frequency of the specific feature value type (eg, the source IP address) in all data values within this statistical window. The higher the data value, the greater the weight, so the higher the priority of the flow entry generated from the data feature value. The collation is based on the value of each feature counter, and the greater the value, the greater the frequency. The packet feature statistics flow chart of the Information Statistics module is shown in Figure 6.

Secondly, the frequency of dropping is less than the threshold Q(Q = 0.04), the attack packet features are corresponding to this frequency, to determine the feature number N of the remaining attack packets. And according to the TCP source port, TCP destination port, UDP source port, UDP destination port, IP source address, and IP destination address, which are corresponding to the feature item extracted from the attack packet in the Information Statistics module, to determine flow entries and to sum the weights of each feature item corresponding to each attack packet, if the sum of weights is the largest, the corresponding flow entry has the highest priority. Finally, according to the statistical frequency {P}, the greater the frequency, the greater the weight of the corresponding feature. We determine the weight of each feature in turn based on $\frac{0.5+\varepsilon}{2^n}(n = 0,1,2,\text{N-1})$, where $n = 0$ represents the weight of the largest feature, and ε is a positive number and takes 0.01. Confirm the weight of the feature {W}.

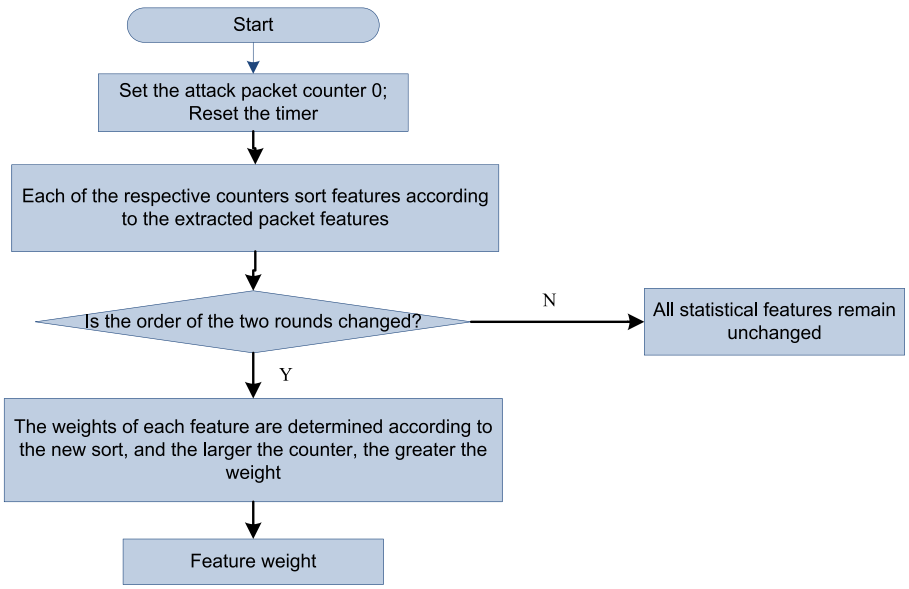


FIGURE 6 Statistical flow chart of packet features of information statistics module

The timer is retiming, the cycle is *T*, and the attack packet counter is set 0. At the same time, to judge whether the TCP source port, TCP destination port, UDP source port, UDP destination port, IP source address, IP destination address, and flow entries priority of the feature item of the attack packet are the same with the issued flow entries. If they are the same, we use the previously generated flow table by default. If they are not the same, fill the TCP source port, TCP destination port, UDP source port, UDP destination port, IP source address, and IP destination address, which are corresponding to the feature item of an attack packet to OpenFlow flow entry matching domain respectively. The instruction set of a flow entry contains the modified action set instruction (flow_mod), then set the action set instruction to drop and fill up the other fields of the flow entry to generate the flow table. The DDoS attack defense process is shown in Figure 7.

5 | EXPERIMENT AND RESULT ANALYSIS

5.1 | Experimental environment

We use ISCX data set[30] training detection model and verify the DDoS defense architecture through real-time DDoS attacks. Training a deep computation model poses a significant challenge since a deep computation model typically involves a large number of parameters. Specially, it needs a high-performance computing server with a large-scale memory and a powerful computing unit.[31,32] Our deep learning model training experiment is based on the hardware environment of 2 NVIDIA K80 GPU and 128 GB memory, and the software environment of the Ubuntu 14.04 operating system and the Keras deep learning framework. The normal data packets and DDoS attack packets needed for training and testing are collected through the ISCX2012 data set. The real-time DDoS attack detection and defense system is built under the ubuntu14.04 operating system and the Keras architecture,[33] and OpenFlow switches are built by using OpenVSwitch. The OpenVSwitch command is used to send the flow list to the OpenFlow switch. In the experiment, we also use Spirent contracting tools (Testcenter (C1)) to produce data packets for simulating data traffic in a real SDN network. Testcenter (C1) can customize the generation of packet traffic, so it can simulate a variety of common DDoS attack packet traffic in the real network, such as ARPFlood, Land, PingofDeath, Smurf, SynFlood, UDPFlood, Teardrop, and so on. C1 can work in various flexible ways to generate data packet traffic, for example, to generate the DDoS attack packet with different attack rates, and multiple attack packets mixed with normal packets. It can also count packets.

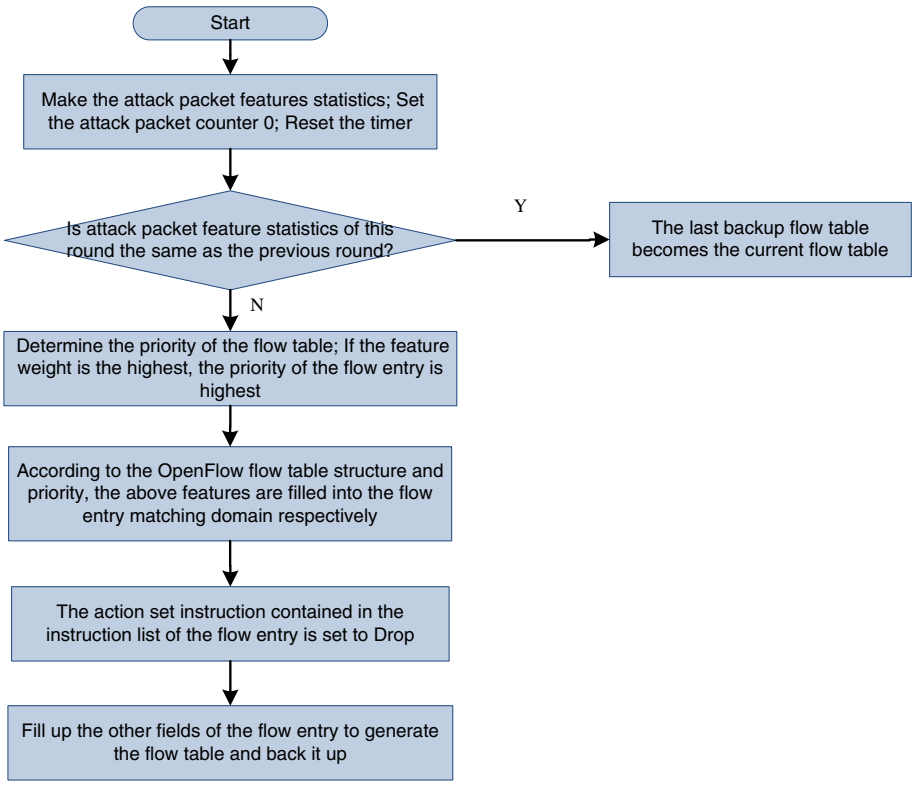


FIGURE 7 Distributed denial of service attack defense flow chart

5.2 | Model training and DDoS attack test

The experimental results presented in this section are based on the deep learning network model, of which the input sample window size is 100. To reduce the unreliability of data sets, the deep learning network model will undergo 10 rounds of training. In the process of deep learning network model training, for each round of training, the first 90% of the data set is used to train the model, and the remaining 10% is used for precision detection. That is to say, firstly, input the first 90% of the training set to the deep network model to train model in each round, and then use the remaining 10% as the validation data sets of current round after the training. We repeat 10 rounds of training to improve the training accuracy of the model, and finally, preserve the ultimate deep learning network model.

The ISCX2012 data set is used as a sample data set for training the deep learning network model of DDoS attack behavior detection in network. ISCX2012 records 7-day traffic information in real network environment, including legitimate network traffic and multiple types of malicious DDoS attack traffic. In the 7-day network traffic records, the DDoS attack takes a period of 2 days from 2010/6/14 to 2010/6/15. We extracted the network traffic for 2 days and saved them into 2 files named Sample_Data_14 and Sample_Data_15. Among them, Sample_Data_14 contains more than about 9.6 million data messages, and Sample_Data_15 contains nearly 35 million data messages.

The ISCX2012 data set lists information related to legal or each DDoS attack data messages, including data type name, capture time, source or destination IP address, TCP/UDP message source or destination port number, and so on. We make analysis and statistics on ISCX2012 data set. Through the statistics and analysis of the network traffic recorded by Sample_Data_14 and Sample_Data_15, respectively, the main attack types of the 2 files and the corresponding data message information are shown in Figure 8.

As we can see from Figure 9, the frequency of the major attack types and their data packets' appearance is known. Firstly, for each normal data message and attack data message in the network traffic, data packets are matched and tagged based on the known attack types. Each data message will be tagged to mark the datagram as an attack packet or nonattack packet. Most of the traffic information in Sample_Data_14 and Sample_Data_15 is legitimate. To eliminate data deviations, when we want to input data to train the deep learning network model, the scheme adopted in this paper is that all attack data packets are mixed with random number of legitimate data packets and then resample to obtain input data for training network model.

In deep learning open-source architecture based on Keras, we construct a variety of deep learning network models based on different LSTM network models to do experiments in this paper and compare the final training accuracy, thus, to select the best deep learning network model for Deep Learning DDoS Detector module being used to detect DDoS attacks in the network environment. The different deep learning network models are illustrated in Table 1.

As we can see from Table 1, the deep learning network models built on different LSTM network models will also be different. We define different names for them (see table item, model name), respectively, named LSTM, CNN/LSTM, GRU, and 3LSTM deep learning network model; meanwhile the number of specific network model layers, the number of neurons, and the type of activation function in each model are given. In view of the 4 deep learning network models based on different LSTM network models, the corresponding comparison results are shown in Figure 10.

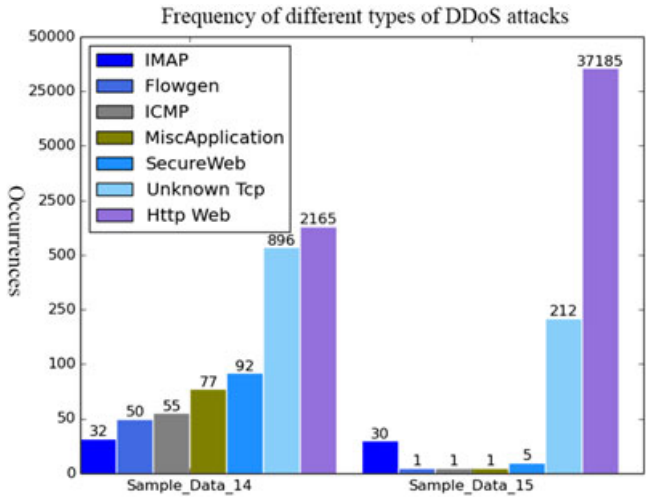


FIGURE 8 ISCX2012 message statistics

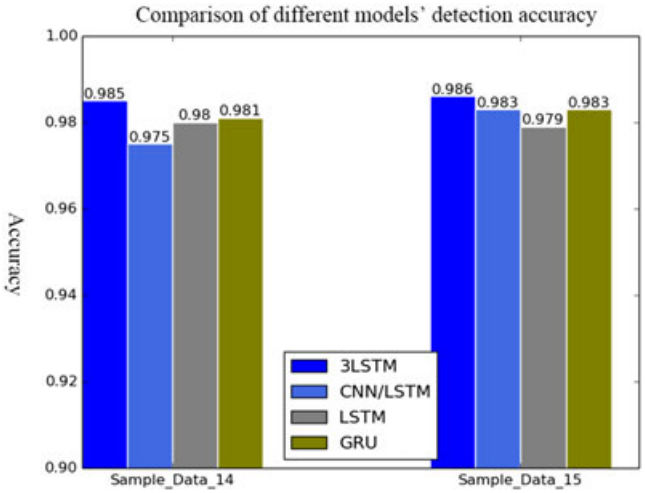
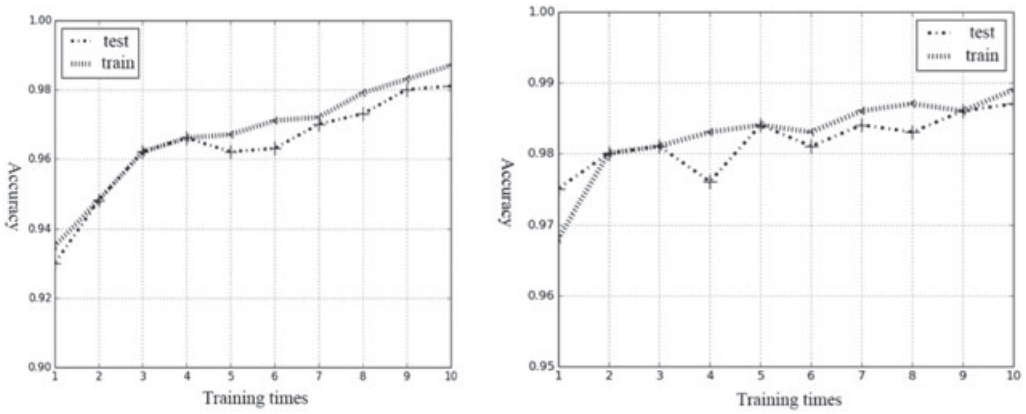


FIGURE 9 Comparison of packets sent and received

TABLE 1 Comparison of 4 different deep learning network models

Model name	LSTM	CNN/LSTM	GRU	3LSTM
LSTM/GRU	4	4	4	6
Neuron number	64	64	64	64
Activation function	tanh	tanh	tanh	tanh
CNN layer	–	2	–	–
Neuron number	–	128	–	–
Activation function	–	relu	–	–
Fully connected layer	8	8	8	8
Neuron number	128,1	128,1	128,1	128,1
Activation function	relu	relu	relu	relu



(A) Sample_Data_14 sample data set (B) Sample_Data_15 sample data set

FIGURE 10 Comparison of model detection accuracy

According to the result of the training accuracy shown in Figure 11, in Sample_Data_14 and Sample_Data_15 data sets, the detection accuracy of 3LSTM deep learning network model is the highest. Combined with the above training results, the 3LSTM deep learning network model has better results for large-scale data traffic detection.

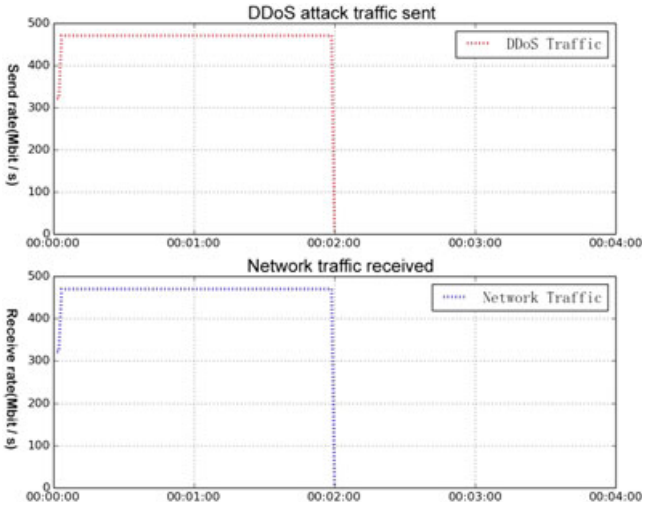


FIGURE 11 Comparison of packets sent and received

In this paper, we introduce the concept of time window when inputting data into deep learning network model. In this way, the general DDoS attack detection system based on the detection mode of single data packet input is changed into DDoS attack detection based on window traffic input. We use custom change timestamp to detect DDoS attacks in network environment. The 4 models use different timestamps and compare the detection accuracy as shown in Table 2.

A larger timestamp can store a longer attack time sequence, and it can reflect more complete attack activity. Therefore, the introduction of timestamp is more convincing for the detection of DDoS attacks. Change the timestamp by changing the size of the test window and the stride. In experiments, timestamp is 50 or 100, to compare the performance of different models with different timestamps on the detection accuracy. As is known from the table, when the window size is 500 and the stride length is 5 (the timestamp is 100), the detection accuracy and the F1 score of the model are the largest. Finally, the timestamp is set to 100 by default.

Training 3LSTM deep learning network model is based on LSTM network model. Sample_Data_14 and Sample_Data_15 sample data sets are trained to verify the results, and the results are shown in the Figure 12. Among them, Figure 12A recorded the relation graph of the precision value with the training times in the training and verification process of the Sample_Data_14 sample data set in the 10 rounds of repeated training; in the same way, Figure 12B recorded the relationship between the accuracy value and the training times during the training and verification process of the Sample_Data_15 sample data set in the 10 rounds of repeated training.

From the model training and verification results, it can be seen that the sample data set is trained repeatedly 10 times and then tested, with the increasing number of training, the training accuracy is higher and higher, and the accuracy of the verification of the data set (accuracy parameter values) will also be higher. In the 10th repeated training, the accuracy of model training and detection reached the highest value. The highest accuracy of the Sample_Data_14 data set tends to be 98%, while the Sample_Data_15 dataset has a maximum precision of up to 99%. Thus, increasing the number of repeated training, in other words, increasing the size of training sample data sets, will make the accuracy of training and verification of deep learning network significantly improved.

TABLE 2 Comparison of detection accuracy of different timestamp models

Window size	Stride	LSTM	CNN/LSTM	GRU	3LSTM
500	50	89.51%	88.66%	91.56%	96.35%
1000	50	87.77%	87.80%	89.24%	96.45%
5000	50	97.85%	96.84%	91.32%	94.39%
500	10	98.54%	97.26%	95.46%	98.77%
1000	10	98.83%	99.46%	98.79%	99.60%
500	5	99.88%	99.37%	99.55%	99.79%

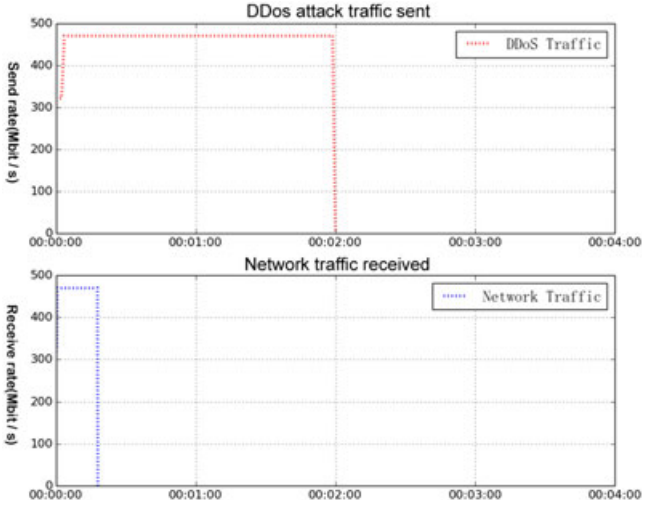


FIGURE 12 Training accuracy with 2 data sets

5.3 | Real-time DDoS attack test

In this experiment, the total amount of data stream contains 4 017 760 packets, including 3 014 110 legitimate data packets and 1 003 650 DDoS data packets. According to defense architecture in Figure 8, using Spirent tester to send DDoS attack to verify defense system, Spirent tester can simulate the client and server at the same time. Considering the security of the system, use Spirent tester to send 500 Mbits/s in the testing process of the DDOS attack, and the duration is 2 minutes.

During the experiment, the DDoS attacks of data traffic generated by the Spirent C1 tool contain 5 type attacks: ARP flood inundation attack, Ping Of Death attack, Smurf attack, SYN flood inundation attack, and UDP flood inundation attack. They are used to simulate complex uncontrollable large-scale DDoS attacks suffered by SDN networks. According to the source IP address and the statistically analysis of data traffic generated by C1, the result is shown in Table 3.

As shown in Table 3, the total number of data traffic packets is 14 307, and the number of DDoS attack packets with the source IP address is 12 400.

During the experiment, testcenter was used to generate the corresponding DDoS attack packets. When the Deep DDoS Defender module is not started, the packets sent from the Testcenter (C1) port1 pass through the OpenFlow switch are all forwarded back to the testcenter through the port2. The network traffic of port1, that is, the DDoS attack. The port2 receives all the DDoS attack packets and compares the number of packets sent and received. Figure 9 illustrates the result.

When the DDoS attack occurs, the Deep DDoS Defender module is launched immediately. Firstly, the Deep Learning DDoS Detector module is used to detect the packets generated by C1. During the experiment, each test result indicates the result of DDoS attacks detection of data flow in one window. When the output *y_proba* represents the probability that the real-time data traffic is detected as DDoS attack in a window, if $y\_proba >= 0.5$, there is a DDoS attack behavior occurring within a detection window. Thus, the corresponding *y_predict* is 1, otherwise 0.

According to the above test results, the data of SDN network traffic information are calculated when DDoS attack behavior occurs, and DDoS attack network segment source IP address information can be statistics. The value field of

TABLE 3 Data flow according to source IP address statistics

Source IP segment	Number
10.20.218.0/24	5036
10.20.222.0/24	6686
10.20.223.0/24	260
10.20.224.0/24	256
10.20.225.0/24	242

the flow entry can be generated according to the set in the paper. Taking the source IP address value (src.ip) as an example generates flow entries that drop the data flow where the source IP is the DDoS attack segment.

If Deep DDoS Defender module detects that the packet is an attack packet, immediately pass through the OpenFlow protocol and send the dropped flow entry to the switch.

ovs-ofctl add-flow br0 nw_src = './home/experiment/source_ip.txt',action = drop

After 1 minute, the packets sent from the Testcenter (C1) port1 are dropped through the OpenFlow switch. As we can see from Figure 11, the DDoS traffic of port1 is different from the network traffic of port2. After 15 seconds, the port2 does not receive any packets. The port2 shields the DDoS attack packets after starting the Deep DDoS Defender module.

6 | CONCLUSIONS

We propose a DDoS attack detection and defense method based on deep learning in this paper; the verification accuracy of the DDoS attack using the ISCX data set is as high as 98% or even 99% in the model training phase. At the same time, we give design process of each function module, define the related functions, use Spirent software to send attack packets, and simulate traffic in a real network environment. The defense architecture is tested for defensive effects by real-time DDoS attacks. As can be seen from the final experimental results, in the DDoS attack detection scheme based on deep learning, if the source address is an attack packet IP address, it is basically detectable. According to other feature fields of DDoS attack data traffic, such as the destination IP address, the source MAC address, the source or destination TCP/UDP port number, and other fields, we can also obtain the correct detection results. The OpenFlow flow entries generated by the DDoS attack detection result can effectively clean the DDoS attack traffic and so can alleviate DDoS attacks in SDN networks. Moreover, the completion of traffic cleaning does not affect the normal forwarding of legitimate data traffic. Finally, the effectiveness of the DDoS attack defense scheme based on deep learning in the SDN network is verified in real-time network environment for detecting and defending DDoS attacks.

According to this experiment, it can be seen that the DDoS attack detection scheme based on deep learning has the advantages of high detection accuracy, little dependence on hardware and software devices, and easy updating of network model, so it makes up for the shortcomings of the existing DDoS attack detection schemes. In short, the advantages of DDoS attack detection scheme based on deep learning can be summarized as follows, not only it improves the accuracy of DDoS attack detection but it also reduces the degree of dependence on the hardware and software environment, and simplifies the real-time update of detection system and the difficulty of upgrading the DDoS attack detection strategy.

ACKNOWLEDGMENTS

This work was supported in part by a grant from the National High Technology Research and Development Program (863) of China (2015AA011901), the National Natural Science Foundation of China (61402408 and 61379120), the Zhejiang Provincial Natural Science Foundation of China (LY18F010006), and the Zhejiang's Key Project of Research and Development Plan (2017C03058).

ORCID


Chuanhuang Li http://orcid.org/0000-0003-0132-4755

REFERENCES

1. Anders N, Ben P, Bob L, et al. ONF TS-020, OpenFlow switch specification version 1.5.0[S]. December 19, 2014
2. Criscuolo PJ. Distributed denial of service: Trin00, tribe flood network, tribe flood network 2000, and stacheldraht ciac-2319. Tech. rep., DTIC Document, 2000.
3. Herberger C, Hoffman Y, Groskop M, et al. Global application & network security report 2015-2016. Tech. rep., 2016.
4. Kaspersky Lab. Kaspersky ddos intelligence report for q4 2015. Tech. rep., 2016.
5. Software-defined networking: the new norm for network. ONF White Paper[OL]. April 13, 2012.https://www.opennetworking.org/images/stories/downloads/whitepapers/wp-sdn-newnorm.pdf.
6. Mckeown N, Anderson T, Peterson L, et al. OpenFlow: enabling innovation in campus networks[J]. *ACM SIGCOMM Computer Communication Review*. 2008;38(2):69-74.

7. Peng T, Leckie C, Ramamohanarao K. Survey of network-based defense mechanisms countering the DoS and DDoS problems[J]. *Acm Computing Surveys*. 2007;39(1):3
8. Mirkovic J, Martin J, Reiher P. A taxonomy of DDoS attacks and DDoS defense mechanisms[J]. *AcmSigcomm Computer Communication Review*. 2001;34(2):39-53.
9. Zhou H, Leung VCM, Zhu C, Xu S, Fan J. Predicting temporal social contact patterns for data forwarding in opportunistic mobile networks[J]. IEEE Transactions on Vehicular Technology, 2017.
10. Braga R, Mota E, Passito A. Lightweight DDoS flooding attack detection using NOX/OpenFlow[J]. *J Nucl Cardiol*. 2010;8(Suppl1):408-415.
11. Kamijo K, Tanigawa T. Stock price pattern recognition-a recurrent neural network approach[C]//neural networks, 1990., 1990 IJCNN International Joint Conference on. IEEE, 1990: 215-221.
12. Hochreiter S, Schmidhuber J. Long short-term memory[J]. *Neural Comput*. 1997;9(8):1735-1780.
13. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks[J]. *Advances in neural information processing systems*. 2012;25(2):1097-1105.
14. Tamaru A, Gilham F, Jagannathan R, et al. *A real-time intrusion-detection expert system(IDES)*. Computer Science Laboratory: SRI International; 1992.
15. Ferguson P, Senie D. Network ingress filtering: defeating denial of service attacks which employ IP source address spoofing[J]. RFC, 1998.
16. Gil TM, Poletto M. MULTOPS: a data-structure for bandwidth attack detection[C]// conference on Usenix security symposium. *USENIX Association*. 2001;3-3.
17. Fonseca P, Bennesby R, Mota E, Passito A. A replication component for resilient OpenFlow-based networking[C]. IEEE Network Operations & Management Symposium. IEEE, 2012:933-939.
18. Kim MS, Kang HJ, Hang SC, Chung SH, Hang JW. A flow-based method for abnormal network traffic detection.[C]// network operations and management symposium, 2004. NOMS 2004.IEEE/IFIP. IEEE, 2004:599-612.
19. Wang X. Huawei SDN technology into the Unicom cloud cleaning business from the source to reject DDoS attacks [J]. *Communications World*. 2014;27:20-20.
20. Gao S, Peng Z, Xiao B, Hu A, Ren K. FloodDefender: protecting data and control plane resources under SDN-aimed DoSAttacks[C]. INFOCOM 2017-IEEE Conference on Computer Communications, IEEE, 2017: 1-9.
21. Niyaz Q, Sun W, Javaid AY. A deep learning based DDoS detection system in software-defined networking (SDN)[J]. arXiv preprint arXiv:1611.07400, 2016.
22. Review T. Google's deep learning machine learns to synthesize real world images[J]. 2015.
23. Metz C. Facebook's 'Deep Learning' guru reveals the future of AI[OL]. [2017-2-12]. https://www.wired.com/2013/12/facebook-yann-lecun-qa/.
24. Li D. Recent advances in deep learning at Microsoft: a selected overview[OL]. [2017-2-12]. http://research.microsoft.com/jump/241484/
25. Yu K. Large-scale deep learning at Baidu[C]. ACM international conference on Information & knowledge management. *ACM*. 2013;2211-2212.
26. LeCun Y, Bengio Y, Hinton G. Deep learning[J]. *Nature*. 2015;521(7553):436-444.
27. LeCun Y, Bottou L, Bengio Y, Patrick H. Gradient-based learning applied to document recognition[J]. *Proc IEEE*. 1998;86(11):2278-2324.
28. Zhang Q, Yang LT, Chen Z, Li P, Deen MJ. Privacy-preserving double-projection deep computation model with crowdsourcing on cloud for big data feature learning[J]. *IEEE Internet of Things Journal*. 2017;
29. Dey R, Salem FM. Gate-variants of gated recurrent unit (GRU) neural networks[J]. 2017.
30. UNB ISCX intrusion detection evaluation DataSet[OL]. http://www.unb.ca/research/iscx/dataset/iscx-IDS-dataset.html.
31. Zhang Q, Yang LT, Chen Z, Li P. An improved deep computation model based on canonical polyadic decomposition[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2017.
32. Zhang Q, Yang LT, Liu X, Chen Z, Li P. A Tucker deep computation model for mobile multimedia feature learning[J]. *ACM Transactions on Multimedia Computing, Communications, and Applications*. 2017;13(3s):39:1-39:18.
33. Ketkar N. Introduction to Keras[M]. Deep Learning with Python. Apress, 2017.

How to cite this article: Li C, Wu Y, Yuan X, et al. Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN. *Int J Commun Syst*. 2018;31:e3497. https://doi.org/10.1002/dac.3497