RESEARCH ON DDOS ATTACK DETECTION BASED ON ISOLATION FOREST AND K-MEANS ALGORITHM IN SDN

Zhaohui Ma and Zhuojie Li*

School of Information Science and Technology, Guangdong University of Foreign Studies
Guangzhou, China

*Corresponding author

ABSTRACT

Compared with the traditional network, as a new generation of network architecture, Software Defined Network (SDN) has the advantages of separation of control and forwarding, centralized control, and is more suitable for today's complex and changeable network environment. Due to these characteristics that the controller is vulnerable to DDoS attack in SDN, so it is necessary to study the security of SDN. This paper proposes a DDoS attack detection method based on the combination of isolated forest and K-means algorithm in software defined networks. By collecting the information of flow table in SDN network, extracting the characteristics of flow table, using the model to classify and predict the feature vector of flow, the potential attack flow is screened out, thus DDoS attack detection can be realized. The experimental results show that the algorithm can achieve 99% accuracy on a variety of test sets.

KEYWORDS

Software Defined Network, Isolation Forest, K-Means, Distribution Denial of Service Attack Detection

1. INTRODUCTION

1.1. Research Background and Significance

With the arrival of the era of big data and artificial intelligence, traditional network equipment has been increasingly unable to adapt to the needs of the network, so software defined network (SDN) as a new network architecture emerged. Its core ideas are decoupling of control and forwarding, abstraction and programming. Compared with the traditional network, the control function of SDN is realized by the controller. The network transmission equipment is only used for high-speed forwarding of data abstracted into flow table, which is more flexible, agile and programmable. Therefore, it is widely used in the Internet of things and data center network as one of the basic technologies for 5G network deployment[1-2].

With the increasing attention to network security, the research on traffic detection of DDoS attack on traditional network has been a hot research direction for many years, and obvious achievements have been made. However, in the field of SDN, the research on this issue is still in the preliminary stage of exploration and no enterprise-level research has been conducted. In

addition, due to the centralized control of SDN, compared with traditional network, SDN controller is more vulnerable to DDoS attack.

According to the attack situation report of DDoS in 2019[3], the average peak of DDoS attacks in 2019 reached 42.9Gbps, and the number of attacks increased by 30.2% compared with that in 2018. In terms of attack scale, small-scale attacks of 1-5gbps increased significantly, and large-scale attacks over 300Gbps increased by more than 200 times compared with 2018. The main attack types are UDP Flood, SYN Flood and ACK Flood, accounting for 82% of the attack times, among which SYN Flood occupies a dominant position in the mass traffic attack. With the development of DDoS attack, it is more critical to enrich the dimension of attack traffic identification and replace the artificial intelligence to improve the response speed and reduce the interruption time.

To sum up, although the SDN network architecture has the advantages of centralized control, separation of control and forwarding, the traditional network architecture suffers more serious DDoS attacks, so it is urgent to study the detection of DDoS attacks in SDN network. When the SDN controller is attacked by DDoS, it should give early warning in time, so that the system and the operation and maintenance management personnel can take corresponding security measures to reduce the harm to the system.

1.2. Research Status

In traditional network DDoS attack detection, Niu[4] proposed an improved clustering algorithm MFCBR based on clustering and a local outlier mining algorithm LOGD based on grid query. Chen[5] proposed network traffic anomaly detection based on logistic regression and decision tree and distributed denial of service attack detection based on genetic algorithm and gradient promotion tree.

In recent years, the research on DDoS attack detection in SDN has also made some progress. Xu and Sun[6] conducted a comprehensive study on abnormal flow detection of SDN, and summarized the potential network security problems of the control and data forwarding layer in the three-tier architecture of SDN. This paper introduces and analyzes the abnormal flow detection framework in SDN three-tier architecture. It also points out the possible research direction of abnormal traffic monitoring in SDN architecture in the future. But this is more about theory than practical application. Li et al.[7] proposed a research on DDoS attack detection based on SVM algorithm. This method has a higher detection rate than traditional detection methods, but it is greatly affected by the selection of initial data sets and initial kernel functions. Li[8], Zhu et al.[9] proposed DDoS attack detection methods based on XGBoot and DBN respectively, but the detection rate of these methods still needs to be improved, and the single algorithm is also their shortcoming. Meng[10] proposed entropy value combined with CNN detection method for detection. Ma[11] proposed a model combining entropy value detection, SVM and k-means algorithm to detect DDoS attacks. Although these methods have a low CPU utilization rate, they have a low detection rate and a high false alarm rate, which are also inadequate for detection of DDoS attacks.

To sum up, this paper proposes a DDoS attack detection algorithm based on the combination of isolated forest and k-means, which has a high detection rate and a very low false alarm rate.

1.3. Research Content and Article Structure

With the development of SDN network technology, SDN architecture plays an important role in 5G, the Internet of things and other fields, and more and more scholars and enterprises pay more and more attention to the security of SDN.

By analyzing the principle of DDoS attack and combining with the three-layer architecture mode of SDN, this paper investigated the plane protection method of SDN vulnerable to attack according to the existing literature, and determined to protect the control layer and the southward interface. The DDoS detection model of isolated forest and k-means algorithm was proposed. Combining with the characteristics of SDN flow, the module was refined, and the combined model was optimized. At the same time, the detection model with higher detection rate and better performance was selected through horizontal comparison with other models.

The content of this paper can be divided into the following parts:

1) Introduce the background knowledge of SDN. By elaborating the southbound interface protocol OpenFlow, readers can preliminarily master the relevant protocols and data processing modes under SDN.

2) Based on the weakness that SDN controller and southbound interface are vulnerable to DDoS attack, combining with the traffic characteristics of SDN network, this paper introduces the principle of DDoS attack and the types of DDoS attack under SDN, so that readers can know how to carry out DDoS attack under SDN network.

3) As the SDN controller is responsible for the centralized management of the network and the forwarding function of the flow table within the whole network in the SDN network, the performance of the controller is crucial to the performance evaluation of the system. So by RYU controller programming experiments to select suitable model to test the DDoS attacks, implementation flow chart characteristic collection, flow processing, model training, and other functions, and through to the k-means algorithm are isolated forests and combined with the optimization algorithm model, improve the detection effect, improve system performance and achieve the real time detecting DDoS attacks.

2. RELATED KNOWLEDGE

2.1. DDoS Attacks

Now DDoS attack has become the primary threat of network security, DDoS attack or distributed denial of service attack, refers to the attacker from different position control by means of a one-to-many host to form "botnets", using forged out of a large number of IP address and the target site or system to establish connection requests in a short time, makes a firewall or other equipment, unable to identify which server network bandwidth and CPU resources and

etc. The server paralyzed and unable to run normally, and provide the normal request for legitimate users.In traditional networks, attackers select different attack types according to different attack targets[12]. Common DDoS attacks include traffic type attacks, such as ICMP Flood and UDP Flood. Vulnerability attack, such as SYN Flood; Reflective attacks, such as Smurf attacks, NTP reflection attacks, etc.

2.2. DDoS Attack Under SDN

In SDN network, different attack types are different for different planes, so DDoS attacks under SDN can be classified according to the plane type[13]. (1)Apply plane attack. Due to the openness of the northbound interface, malicious code implantation may become the attack method selected by the attacker. (2)Control plane attack. Control layer is the core of the controller, it is because of the characteristic of the controller centralized control, the source IP address of the attacker sends a large number of small false invalid packets, make the switch frequently send Packet - in a message to a controller, led to the controller cannot be distributed to each Packet forwarding strategy in time, so the controller resource is occupied by a large amount of invalid data packets, affect the performance of the system, create a single point of collapse; (3)Forwarding plane attack. For the switch, due to the limited buffer space of the switch flow table, there are a large number of invalid packets in the network, which will constantly consume the switch's cache. Therefore, when each flow arrives, it is impossible to match and forward normal packets quickly and consume link bandwidth. In order to attack the network host, the attacker takes advantage of the protocol vulnerability, initiates a large number of connection requests, consumes a large amount of network resources of the target system and CPU resources of the target host, etc., making it unable to provide services for normal users' normal requests.

2.3. K-Means

K-means is an unsupervised clustering algorithm that can divide unlabeled data. At the beginning, K points in the data set are randomly selected as the clustering center. The specific algorithm steps are as follows:

① K pieces of data were randomly selected from the initial data set as the clustering center of the cluster.
②Traverse all data P, calculate the distance from P to each cluster center, and divide the data into the set of the nearest cluster center.
③Traverse K sets and calculate the central position of each cluster as the new cluster center.
④Repeat 2 and 3 until the cluster center no longer changes.
In this paper, Euclidean distance is used to measure the distance between cluster centers. The Euclidean distance is calculated as Formula (1) :

$$d = \sqrt{\Sigma_{i=1}^{n}\left(x_i - c_j\right)^2} \quad (1)$$

Where, x_i represents each data and c_j represents each cluster center.

The advantage of the k-means algorithm lies in its fast convergence speed and easy implementation. The disadvantage lies in the improper selection of the initial K value, which will affect the classification results and be sensitive to outliers.

2.4. Isolated Forest

Isolation Forest algorithm was proposed by Fei Tony Liu, Kai Ming Ting and Zhou Zhihua [14-15], and jointly determined the basis of isolated Forest algorithm.

Algorithm principle:

From N selection data set out the ψ sample of data as one of the training samples of the iTree tree, a feature from samples selected randomly to two poor division of sample data, which is based on the comparison of the characteristic value of the data set other data division, under the same feature is less than the value of data classification to the left, to the right is greater than the value of data classification.

So to this value to get the split conditions in the two data sets, and then respectively in the two sides of the data sets the value of repeated selection process, divided into until termination condition:(1) Data cannot be divided, may contain only a sample or samples of remaining the same, or (2) The height of the tree to $\log_2(\psi)$. Generally, iTree internal algorithm limits the height of the tree considering the execution efficiency of the algorithm.

Due to the low reliability of a tree, a sufficient number of iTree trees need to be built before the algorithm starts to predict the data. Will test data from the root node in the iTree tree, according to the classification conditions along the path of the corresponding divided down, until you reach a leaf node, and record the data in the process of dividing path length h (x), the data from the root node to leaf node by the number of edges (path length), such as Formula (2), which factor is the euler's constant.

$$h(x) - ln(x) + \xi \tag{2}$$

The average path length data range of binary search tree fluctuates greatly, so it is necessary to normalize the data, as shown in Formula (3).

$$C(n) = 2H(n-1) - \left(\frac{2(n-1)}{n}\right) \tag{3}$$

Finally, the Anomaly Score of each piece of data to be tested is calculated, as shown in Formula (4).

$$s(x,n) = 2^{\left(-\frac{E(h(x))}{C(n)}\right)} \tag{4}$$

Where E(h(x)) represents the expected average path length of all iTree trees in the forest.

3. SYSTEM MODEL

3.1. Model Evaluation Indicators

In order to easily distinguish the categories after the traffic prediction, a confusion matrix is constructed according to the possible prediction results, as shown in Table 1.

Table 1. The confusion matrix of classification results

Flow type	Attack flow	Normal flow
Attack flow	TP	FN
Normal flow	FP	TN

Where, TP represents the number of test samples whose attack traffic is correctly identified; FP represents the number of test samples in which normal traffic is wrongly identified as attack traffic. TN represents the number of test samples correctly identified for normal flow; FN represents the number of test samples in which the attack traffic was incorrectly identified as normal traffic.

Accuracy: Accuracy is the proportion of the number of samples correctly identified for normal flow and attack flow in the sample to the total number of samples. Its accuracy is related to the system error. The higher the accuracy is, the more accurate the flow classification is. The calculation method is shown in Formula (5).

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

False alarm rate: false alarm rate is the proportion of the number of samples in the sample where normal traffic is wrongly identified as attack traffic. The algorithm itself and configuration will affect the false alarm rate. The lower the false alarm rate is, the less influence it has on the system, and the higher the classification detection rate is. The calculation method is shown in Formula (6).

$$FPR = \frac{FP}{TN + FP} \tag{6}$$

Error reporting rate: error reporting rate is the proportion of the number of samples in which the attack flow is wrongly identified as normal flow in the sample. The lower the error reporting rate is, the stronger the identification of the attack flow is and the more reliable the system is. The calculation method is shown in Formula (7).

$$ER = \frac{FN}{TP + FN} \tag{7}$$

CPU utilization rate: since system resources will be heavily used when the system is attacked by DDoS, the lower the CPU utilization rate of the model is, the less the model consumes system performance.

In this paper, there are two evaluation criteria for the model. The first one is the evaluation of accuracy, false alarm rate and error rate. Only isolated forest is used, and only k-mean is used. The second is the evaluation of system resource utilization. In this experiment, the system CPU utilization is selected as the reference. If the prediction results of the final model have higher accuracy, lower error rate and lower CPU utilization, the model is considered to be better.

3.2. Feasibility Analysis

There are usually two ways to detect abnormal traffic. The first way is to compare the new traffic data with the normal traffic data model. The second method compares the new traffic data with the known attack methods[16]. This paper adopts the first method and adopts a combined machine learning algorithm to conduct DDoS attack detection. Each machine learning algorithm has its own advantages. Although it has a good effect on some specific training sets, the training effect may be completely different after the data set is replaced.

Based on the characteristics of isolated forest and k-means algorithm, the isolated forest algorithm can obtain obvious abnormal detection effect when the number of abnormal samples is small. The system memory consumption in the initialization process of the algorithm is not high, and the model processing speed is fast after the prediction tree is built, which has a good effect on the system with high real-time performance. The error rate of the isolated forest algorithm is low, but the accuracy is insufficient[17].

Principle of k-means algorithm is simple, easy to understand, convenient algorithm implementation and improvement, even under the condition of the clustering samples less also can classify samples, and the algorithm process of constant iterative update class cluster center, can to iterative optimization of data set, with the increase of sample can remove a small amount of sample is not accuracy, low time complexity. The K mean is more accurate than the isolated forest in the prediction rate, but the error rate is much higher than the isolated forest.

Therefore, in order to reduce the influence of the data set on the prediction effect, the method of combining machine learning algorithm is adopted to output the prediction result after combining the classification results of various machine learning algorithms, which is conducive to reducing the chance of classification results. If the above two algorithms are combined, obvious detection effect can be achieved in the initial stage, and the errors of isolated forest prediction samples can be corrected by k-means algorithm, so as to improve the final detection rate and accuracy rate and reduce the error reporting rate.

3.3. System Architecture

In order to improve the system detection efficiency, the system model is divided into flow table collection, feature extraction and anomaly detection modules based on the characteristics of flow table in SDN network. The flow table collection module is designed to process the flow table information of the switch and carry out flow injection. The feature extraction module is designed to extract the flow characteristics and model features from the flow table. The anomaly detection module is responsible for monitoring the system for real-time flow detection. The system architecture is shown in Figure 1:

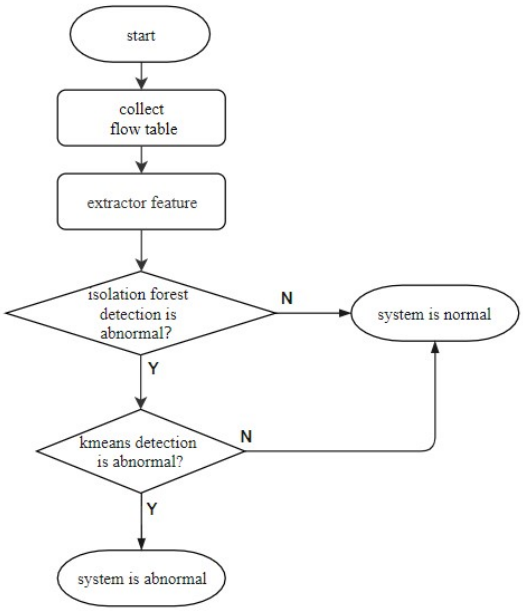

Figure 1. The system model process

3.4. Flow Table Collection Module

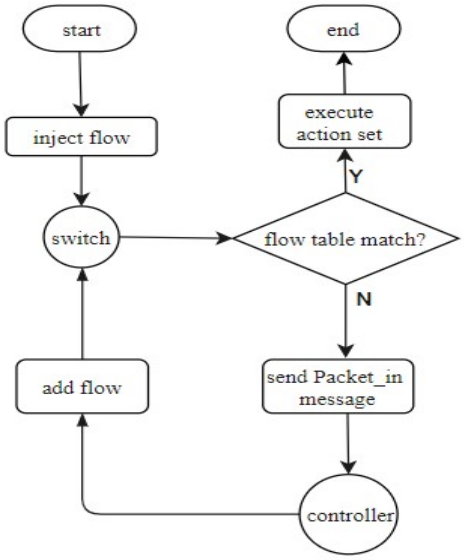

Figure 2. Flow table extraction module

The flow table collection module is responsible for the learning of the switch, the processing of the flow table and the flow injection of the controller. Under the openflow protocol, the Controller periodically sends controller-to-switch messages to the Switch to obtain the flow table information of the Switch according to the EventOFPSwitchFeatures event.

When there is traffic injection in the network, the switch extracts the protocol type, source, destination IP, MAC address and other information for matching. If the matching is successful, it will process according to the instruction set of the flow table. If the match fails, Packet_in message is sent to the controller. After receiving the message via EventOFPPacketIn event, the controller extracts the protocol version, inbound port, datapath and other information from the flow table. After processing according to different protocols, the flow table is issued and installed to the corresponding switch. If it is an ARP message, record the incoming port of the source IP; If it is ICMP message, extract the source IP and destination IP address; If it is a TCP message, extract the source, destination IP and MAC addresses; If it is a UDP message, extract the source and destination MAC addresses. The module flow chart is shown in Figure 2.

3.5. Feature Extraction Module

The feature extraction module is responsible for extracting feature values from the traffic injected into the network and recording the flow table features into the features.csv file. DDoS attacks in the SYN Flood attack is the attacker using TCP protocol defects, through forged IP addresses to send a large number of TCP connection request, the three-way handshake phase, puppet machine sends a connection request message, enter half connection status, and no longer confirm confirm message returned by the server, the server sends or timeout, bandwidth or the resources of the target host. When a large number of such cases occur, the server will consume a lot of resources to maintain such half-connection requests and cannot normally process normal requests from users. In order to hit the server with the minimum cost, the attacker will forge a large number of IP and port addresses, reduce the packet size, and increase the number of packets sent. As a result, the number of source IP addresses and ports in the network increases dramatically. Under a normal network, these data changes are relatively stable. Therefore, abnormal flow rate is detected in SDN by analyzing the characteristic changes of flow table within unit time.

3.5.1. Track of Eigenvalues

According to the characteristics of the SDN flow chart and the characteristics of the DDoS attack, according to protocol and flow feature extraction the characteristic values of seven kinds of basic characteristics and aggregation are: flow packet average number, average bit stream packet number, average flow package request rate, average transmission rate, flow port growth, the source IP, flow rate, various characteristic value concept and calculation method is as follows:

1) Average Number of Packets in per flow table (ANP)

In SDN networks, under normal circumstances, the number of packets sent is relatively stable, the flow of each flow table entry number is relatively small, but when a DDoS attack, the attacker can produce a large number of requests, to exhaust the switch caching, and switches the flow of the match, there is no corresponding flow in the table so constantly Packet_In messages

sent to the controller, the controller issued flow table to the switch. As a result, the number of stream entries in the same flow table in a switch increases rapidly, and the average number of stream packets decreases accordingly. Its expression is shown in Formula (8).

$$\mathrm{ANP} = \frac{\sum_{i=1}^{flow_{num}} packet_{num_i}}{flow_{num}} \tag{8}$$

Where $flow_{num}$ represents the number of streams received per unit time, and $packet\_num_i$ represents the number of packets contained in stream i.

2) Average Number of Bytes in per flow table (ANB)

In SDN, under normal circumstances, the size of packets sent and the number of stream entries are stable. However, when a DDoS attack occurs, the attacker will forge a large number of small and meaningless packets in order to generate a large number of streams so as to reduce the attack cost. The packet size changes little, but it constantly triggers the mechanism of the switch sending Packet_In messages to the controller, which makes it difficult for the controller to handle a large number of flow requests at the same time, resulting in the controller performance degradation. In the same flow table, the packet size changes little and the number of stream entries increases rapidly, so the average number of stream bits decreases. Its expression is shown in Formula (9).

$$\mathrm{ANB} = \frac{\sum_{i=1}^{flow_{num}} bytes_i}{\mathrm{flow_{num}}} \tag{9}$$

Where, $bytes_i$ represents the total number of bits in stream i.

3) Average Request Rate of Packet in per flow table (ARR)

Under normal conditions, the duration of the normal flow is much less than that of the attack flow. When a DDoS attack, the attacker will usually produce a large number of entries, and only in very small amounts of flow set small invalid packets, actually effective flow entry number very few, and within each flow contains a data packet and flow duration will be longer, less so when network to attack, the index will be significantly reduced. The expression is shown in Formula (10).

$$ARR = \frac{\sum_{i=1}^{flow_{num}} packet_{num_i}}{duration} \tag{10}$$

Where duration represents the duration of the flow.

4) Average Transmission Rate of bytes in per flow table (ATR)

In the case of a DDoS attack, the attacker will usually generate a large number of stream entries to establish a large number of invalid connections with the server and maintain them for a long

time to consume its resources, but the amount of data to be sent is small or even zero. Therefore, the data transmission rate will be significantly reduced compared with normal traffic. Its expression is shown in Formula (11).

$$ATR = \frac{\sum_{i=1}^{flow_{num}} bytes_i}{duration} \tag{11}$$

5) Port Generating Speed (PGS)

In a normal network, the connections between processes are relatively smooth and the number of ports is relatively stable. However, when a DDoS attack occurs, the attacker will generate a large number of processes to establish a connection and randomly select any port among them to launch the attack. Therefore, in the same flow table, the number of ports generated increases rapidly and the port growth rate increases. Its expression is shown in Formula (12).

$$\mathrm{PGS} = \frac{port_{num}}{time} \tag{12}$$

Where, $port_{num}$ represents the total number of ports corresponding to each source IP address, and time represents the time consumed to extract the characteristics of the flow table after the controller sends the statistical flow information and receives the reply message from the switch.

6) Flow Generating Speed (FGS)

When a DDoS attack occurs, because there are no corresponding matching flow table items in the switch for a large number of packets forged by the attacker, flow requests are constantly sent to the controller and the controller is asked to issue a forwarding strategy. Therefore, the number of flow items in the flow table increases rapidly, and the growth rate of flow increases rapidly, as shown in Formula (13).

$$\mathrm{FGS} = \frac{flow_{num}}{time} \tag{13}$$

7) Source IP Generating speed (SIG)

In a normal network, the IP address of the packet source changes little because the communication between the host and the server is relatively stable after the connection is established. However, when a DDoS attack occurs, the attacker can control a large number of "zombie hosts" to communicate and randomly generate a large number of source IP addresses. The number of source IP addresses increases rapidly during the controller's processing of the flow table, so the growth rate of source IP addresses will increase. Its expression is shown in Formula (14).

$$\mathrm{SIG} = \frac{src_{ip_{num}}}{time} \tag{14}$$

Where src_ip_num represents the number of source IP addresses per unit time.

From the above data, the seven-element feature vector D_i=<ANP，ANB，ARR，ATR，PGS，FGS，SIG> is formed, which represents the flow characteristics of the flow table in the same event. At the same time, labels are marked in the extraction of normal traffic and attack traffic, so as to optimize the model.

3.5.2. Module Workflow

Controller to monitor switches, through periodic send switch flow request message and online status through EventOFPStateChange event access switches (Main/Dead) to update the link

state, after get flow chart of information through EventOFPFlowStatsReply events, extraction flow, the number of bits, the number of packets, IP and MAC address and other network basic information, to deal with the data flow and calculate the corresponding aggregation characteristics of ANP, ANB, ARR, ATR, PGS, FGS, SIG, It is recorded for use by the abnormal flow module. The process of feature extraction module is shown in Figure 3.

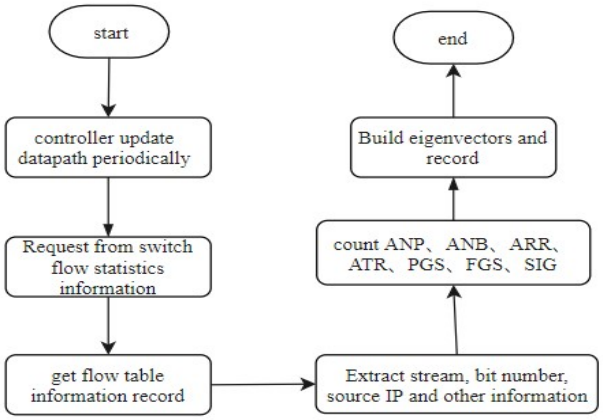

Figure 3. Feature extraction module process

The feature extraction module implements traffic collection through the feature_extractor_normal.py and feature_extractor_attack.py script files. Their functions are similar, except for the difference of tag collection. The main function is to realize the switch to send request events to the controller and collect flow table item information. Method functions and roles are shown in Table 2:

Table 2. Feature_extractor.py main function definition

Function name	Definition
_monitor	The controller monitors the switches through coroutines and sends request message to the switches to get the flow table periodically.
_records	Process and record the flow features information into the collect file, and ignore the flow that the features are all zero during system initialization.
_switch_features_handler	Get switch feature information and reset current flow table.
_state_change_handler	When the switch status changes, modify the corresponding datapath list. Add if it is in main state, and delete if it is in dead state
_request_stats	Send flow table request message to switch
_flow_stats_reply_handler	Process flow entry status request information, extract flow entry information, and process according to different protocols. If it is TCP, extract source, destination IP and port address; if it is UDP, extract source and destination port address.

3.6. Exception Detection Module

3.6.1. Module Detection Process

The anomaly detection module is responsible for the recognition of the flow, which is divided into the model training stage and the real-time flow detection stage. In the model training phase, the pandas library is used to improve the file reading speed for model training. The isolated forest algorithm is used to score the traffic training set. If the score result is positive, the traffic will be considered as normal. If the score result is negative, the test samples will be transferred to the k-means module for final classification and the prediction model will be obtained. In the real-time flow detection stage, the controller obtains the characteristics of the real-time flow through the feature extraction module, USES the prediction model to detect the real-time flow, and outputs the prediction results of the flow. The working process of this module is shown in Figure 4.

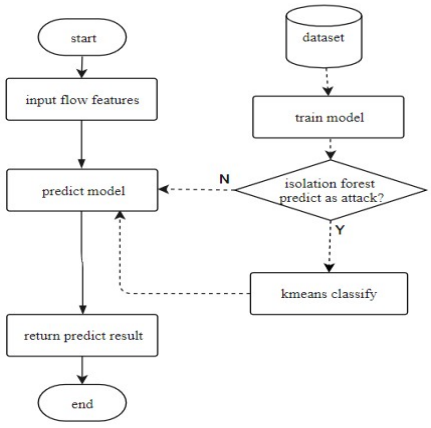

Figure 4. Anomaly detection module process

3.6.2. Detection Algorithm

DDoS attack detection algorithm steps:

①Input of flow characteristics
②The isolated forest model was trained, and the predicted results were divided into normal flow and attack flow
③If the predicted result is a normal flow, normal class labels will be added to the data directly
④If the predicted result is an attack stream, all the data whose predicted result is an attack stream will be transferred to the k-means module for training
⑤Divide K clusters and calculate the center of each cluster
⑥Traverse the samples in each step ④and divide them into the nearest cluster

⑦Repeat steps ⑤and ⑥ until the center is no longer changing
⑧If the samples are divided into normal clusters, normal class labels are added to the data; If the sample is divided into attack clusters, the data is tagged with an exception class
⑨Output classification result model
⑩Real-time detection of traffic data and prediction of results

3.6.3. Function Description

The model training process was realized by the train.py script, and the real-time detection stage was realized by the monitor. Py file. The main function of the model training stage is to train the data sets collected in the feature extraction stage, and to obtain the prediction model by combining the isolated forest and k-means algorithm. The main function of the real-time detection stage is that the controller periodically monitors the switch in real time, obtains the request information of the switch, collects the results of the characteristic processing module, and then processes the response message data to judge whether the traffic in the system is abnormal traffic.

The function description and functions of the method in the real-time detection stage are shown in Table 3.

Table 3 Monitor main function definition

Function name	definition
_init	Start the controller, collect stream table information, and initialize isolated forests and K-means models
_monitor	The controller get the switches status information periodically through coroutines and add flow to the switches, record the flow features to classify and predict.
classifyFeatures	Classify and predict the flow according to its features. If the predict result of isolation forest is normal, return True. But if the predict result of isolation forest is attack, use k-means to classify again. If the result of k-means is normal, return True, or return False.

_record	Process and record flow features information, then call the classifyFeatures function and log flow features and classification results to the _rcd file as a log.

3.7. Model Optimization

3.7.1. Data Normalization

Because the difference of data magnitude will interfere with the classification results, the original data should be normalized. In this paper, minimum and maximum normalization is adopted to ensure that all data are within the ideal interval. The normalized formula is calculated as Formula (15).

$$S = \frac{n - min(x)}{max(x) - min(x)} \tag{15}$$

3.7.2. K Value Selection

Due to the k-means algorithm in the selection of K value on classification results will produce great influence, therefore this article through the elbow method [18] to select the best K value, the elbow method is a kind of using the error sum of squares (SSE) and the K value of diagram to confirm the optimal values of K, principle is with the increasing of cluster number K, data gathering is more and more finely divided, error square value will decrease gradually. When K value is less than the real clustering number, the variation of error square value will be obvious with the increase of K value. When K value reaches the real clustering number, the average error will flatten with the increase of K value, and finally take the shape of elbow. The formula for calculating the sum of squares of errors is as follows:

$$SSE = \sum_{i=1}^{k} \sum_{p \in k_i} |P - C_i|^2 \tag{16}$$

Where, P is the center of mass of each sample point and Ci is the center of mass of the cluster center.

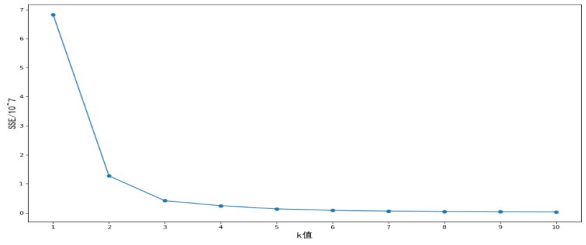

Figure 5. K-value selection

After clustering the samples, a polyline graph of the corresponding relationship is drawn according to K value and SSE value, as shown in Figure 5. Finally, when K=3, the model is optimized.

3.7.3. Improvement of K-means Algorithm

Since the points initially selected by the k-means algorithm will also have a serious impact on the classification results, this model adopts KMeans++ algorithm to eliminate the impact of the initial points[19]. KMenas++ improves the selection of initial points in the k-means algorithm. Different from the random selection of initial points by the k-means algorithm, the further the distance between the sample points in the selection of the initial clustering center, the greater the probability that KMeans++ will become a new clustering center. The basic description of the algorithm is as follows:

① A point in the data set is randomly selected as the first cluster center
② Traverse each point in the data set and calculate its distance D from the nearest cluster center
③The point with a larger D value in the data set is more likely to be selected as the new clustering center
④Repeat 2 and 3 until k initial clustering centers are selected,
⑤Start to implement the k-means clustering algorithm

3.7.4. Parameter Adjustment of Isolated Forest Algorithm

The isolated forest algorithm is greatly influenced by the number of predicted trees and the number of samples[20]. The experiment shows that when the number of iTree is 100, the path length has been covered completely, and the more the number of iTree trees, the lower the prediction effect may be. So set the verbose parameter to 100.

Since the collected flow characteristics are high-dimensional data, the detection effect of the isolated forest algorithm for high-dimensional data is insufficient, so max_features is set to 3, and three features are randomly selected for sampling every time itree is built.

Different from other models, the more data the isolated forest algorithm samples, the lower its ability to identify abnormal data. Moreover, when there are fewer abnormal points in the sample, the prediction effect is better, even if there are no abnormal points in the sample, the prediction can be carried out. The training effect can be improved by adjusting the sample size. The experiment shows that the algorithm has the best detection effect when the sample number is 256. So set the max_sample parameter to 256.

3.7.5. Model Sequence

Since the model can use K means for classification and then use isolated forest for prediction, or use isolated forest for initial prediction and then use K means for further judgment, different model sequences will bring different results. This paper measures the accuracy rate, false alarm rate and error rate, and tests 10,000 pieces of data collected in SDN network. In the first model, the model detection accuracy and false alarm rate of the isolated forest algorithm are lower after the k-means algorithm is used first. In the second model, the detection accuracy is higher when the isolated forest is used first and then the k-means algorithm is used. The error reporting rate

of the first model is significantly higher than that of the second model, as shown in Figure 6. Due to the serious harm of DDoS attack stream to the system, each attack stream should be identified as far as possible. Therefore, due to the high accuracy and low error rate of the system, the second model is adopted in this system, that is, the isolated forest is initially identified, and then the k-means model is used for further classification.

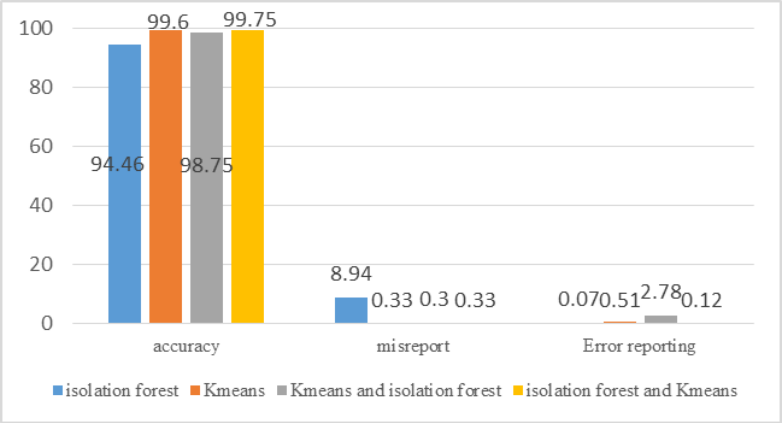

Figure 6. Comparison of model effects

3.7.6. Optimization Results

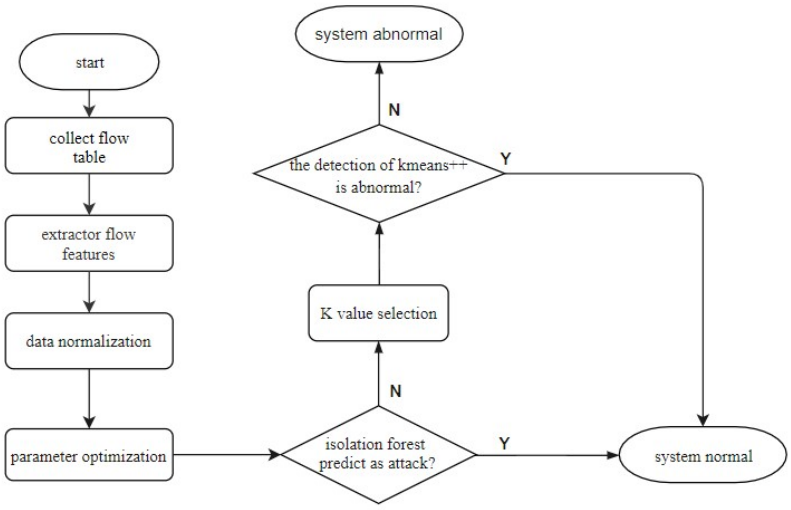

Figure 7. Optimize system model

After collecting the flow table and extracting the aggregation features corresponding to the flow, the controller USES the minimum and maximum normalization to normalize the data in order to prevent the influence of large data fluctuation range between features on the predicted results. By optimizing the parameter properties of the isolated forest, the prediction effect and

performance of the isolated forest algorithm are improved. Start running algorithm to predict isolated forests, if isolated forest algorithm to predict the results as normal flow, the judge system is normal, or transfer the data stream characteristics to KMeans++ module to select suitable K value again classified, if KMeans++ module classification results of the sample for attack cluster, argues that the traffic as attack traffic, judgment system is under attack, for normal cluster, argues that the flow rate to normal flow. The flow chart after model optimization is shown in Figure 7.

4. EXPERIMENTAL ANALYSIS

4.1. Experimental Environment

4.1.1. System Environment

Model optimization was carried out in the Windows environment, and the model was migrated to the SDN network environment. Under the Linux system of Ubuntu16.04 virtual machine, the flow feature extraction and real-time detection of unknown traffic were carried out. The detection function was realized through Ryu controller programming, and the network topology built with mininet was centrally managed through the controller. Scapy library is used to construct data packet simulation to generate normal traffic, and Trafgen plug-in is used to simulate attack traffic and conduct DDoS attack.

Trafgen is a tool for simulating DDoS attacks in the Linux environment. It can generate different types of DDoS attacks and modify the parameters and types of attacks through configuration files and script commands.

4.1.2. Topological Environment

The experiment USES python script to build the topology structure. The topology structure is shown in Figure 8:

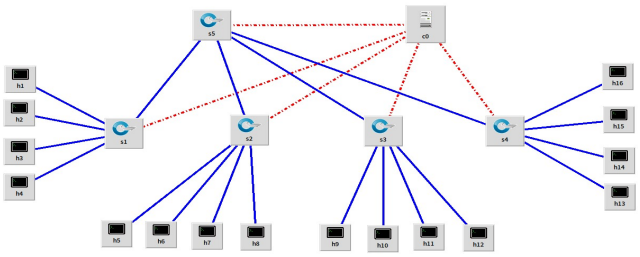

Figure 8. Experimental topology

The network topology structure is shown in figure 10. Five OVS switches are deployed on mininet, among which s1-s4 as the edge switch is responsible for providing users with access to the controller and information collection function in the local network segment. S5 as the convergence layer switch handles all traffic from the edge switch and provides the uplink to the core layer controller. A reliable connection is established between the controller and each switch through a secure transmission channel (red dotted line in figure 10). The blue solid line between

switch and switch and between switch and host is a normal network link, and the bandwidth of all links is set to 100Mbps.

The configuration of controller, switch and host is shown in Table 4 below.

Table 4. Device configuration table

device	IP address and port
Controller c0	127.0.0.1:6653
Switch s1-s5	192.168.[1-5].250
Host h1-h4	192.168.1.[1-4]
Host h5-h8	192.168.2.[1-4]
Host h9-h12	192.168.3.[1-4]
Host h13-h16	192.168.4.[1-4]

4.2. Experimental Process

4.2.1. Flow Table Collection

A) Set up topology: enter the corresponding directory and enter the command sudo python topo.py to set up topology. After starting the controller, check to see if the topology is connected. Go back to the first terminal and enter pingall (test all hosts) or pingpair (test any two hosts) to test connectivity. Successful connection is shown in Figure 9.

```
leo@leo-notebook:~/test$ sudo python topo.py
*** Adding controller
*** Add switches
*** Add hosts
*** Add links
(100.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit) (1
00.00Mbit) (100.00Mbit) *** Starting network
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16
*** Starting controllers
*** Starting switches
(100.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit) (1
00.00Mbit) (100.00Mbit) *** Post configure switches and hosts
*** Starting CLI:
mininet> pingpair
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet>
```

Figure 9. Topological Connectivity Test

B) The experiment USES the scapy library to construct packets to randomly simulate normal data for normal flow injection, and then carries out flow injection on the switch s1-s4 through the script file Rs[1-4]. Py. The functions of the Rs[1-4].py function are similar except that it is injected into different switches and the size of the data sent is different. The rs1.py function is defined in Table 5.

Table 5. Rs1.py function

Function name	definition
port_loader、getPort	Get host port
getSrcAddress、getDstAddress	Get the source and destination IP addresses of the host in series with the switch
sendicmp、sendtcp、sendudp	Send ICMP、TCP、UDP message
generate_random_str	Add any length of data information content to the message
_send	Send different types of reports proportionally, TCP accounts for 85%，UDP for 10%，ICMP for 5%

Collect normal flow: re-open a terminal and enter the command sudo python run.py normal_train to start the controller. If the network is connected, start injecting traffic into the SDN environment, open the new terminal, and enter the command sudo python Rs[1-4].py.

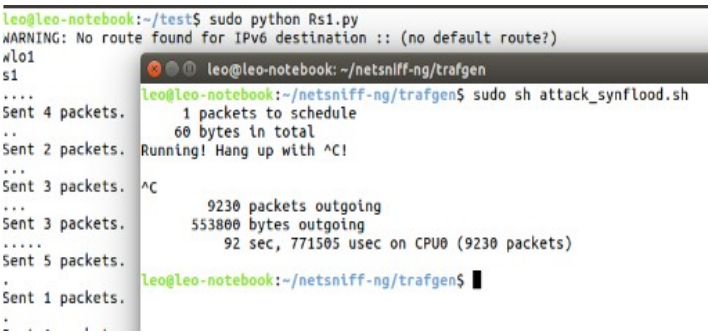

Figure 10. Inject normal and attack flow

C) Useing trafgen tool to simulate DDoS attack for abnormal flow injection.Collect abnormal flow: re-open a terminal, enter the command: sudo python run.py attack_train, start the controller, open the new terminal, enter the trafgen installation directory, enter the command: sudo sh attack_synflood.sh, and the results of normal and abnormal flow injection are shown in Figure 10. After data collection, the features of normal and attack traffic are recorded in the features. CSV file.

4.2.2. Model Training

The data set was trained by isolated forest and k-means algorithm model. The isolated forest was used to predict the training set. If the predicted result was an attack stream, the data whose predicted result was an attack stream was separated from the original data set for k-means training, and the cluster class center of the two clusters of normal stream and attack stream was finally obtained. Enter the command sudo python model.py into the terminal to get the train file after the isolated forest and k-mean model training.

4.2.3. System Resource Occupation

Open the new terminal and enter the top command to view the system resource occupation, as shown in Figure 11. By using the command top -b -p PID >> top.log, the CPU consumption results are output to the top.log file for easy statistics. Where PID is the instruction to run the controller. In this experiment, python3 is used to run the controller. Therefore, the CPU occupancy rate is obtained through the PID of the process corresponding to python3.

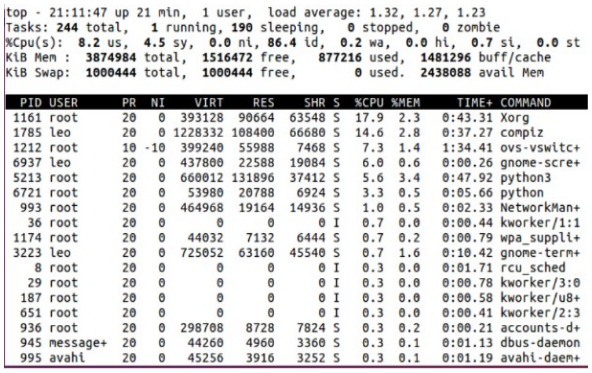

Figure 11. View CPU utilization

4.2.4. Abnormal Detection

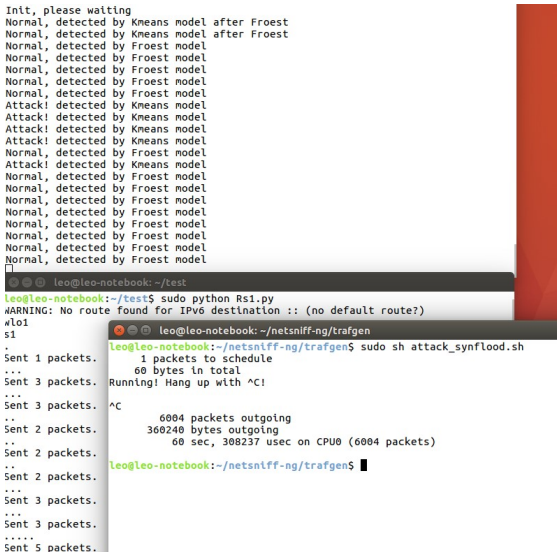

Figure 12. Real-time flow detection

Open the terminal input command: sudo python run.py fk_check starts the controller and selects the model. By changing the parameter fk_check to f_check or k_check, you can switch to the isolated forest and k-mean model for detection, and then open the new terminal to inject normal and attack traffic.

As can be seen in Figure 12, at the initial stage of the system, the user is prompted to send no data within the current network segment. Then, normal flow was injected, but the isolated forest algorithm misreported it, and the k-means were used to detect it. It was identified as normal flow, and the system had no false alarm. The isolated forest then identified the normal flow correctly. After 7 normal flows were sent, the attack flow was injected. Considering the injection attack time, a total of 6 attack flows were injected, of which 5 were correctly identified by the k-means, but 1 was reported as the correct flow by the isolated forest. After the injection of attack traffic, the isolated forest correctly identified the normal traffic sent.

4.3. Experimental Results

The experimental data set was collected into the traffic injected into the SDN network, and 1000 pieces of data were collected from the traffic under different attack intensity. The evaluation criterion was the evaluation index in section 3.1 of this paper, and the combination model of isolated forest, k-mean and isolated forest combined with k-mean was compared horizontally.

4.3.1. Test Accuracy Analysis

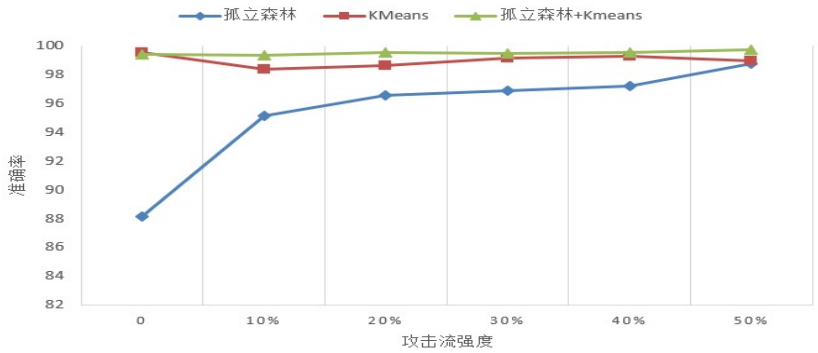

Figure 13. Comparison of test results

The accuracy of three models was tested by setting different intensity of attack flow. The Figure 13 shows that isolated forests algorithm detection accuracy compared with other algorithms from a k-means algorithm flow intensity is lower the attack detection effect than isolated forest combined k-means algorithm, but with the strength of the attack flow increase, combination algorithm can keep high detection rate, and k-means detection effect is on the decline. The model adopted in this paper has a high detection rate of abnormal traffic. One of the reasons is that the network environment in which the experiment is conducted is not a real network environment, and the network traffic characteristics are different from those in the real environment. In the real network environment, the types of DDoS attacks chosen by attackers are more abundant, the attack range is larger and the attack time is longer, the traffic characteristics also have more distinguishable dimensions, the data set collected is more complex and abstract, and the correlation between features is more fuzzy, which finally leads to the decline of classification accuracy.

4.3.2. Resource Occupancy Analysis

As shown in Figure 14, at the controller startup stage, the model algorithm has just been initialized, so the CPU resource utilization rate is relatively high, among which the k-mean algorithm has a high CPU utilization rate. When the system receives normal traffic, the CPU utilization of the three models is at a low level. At 30s, when DDoS attack stream was injected, the system resource occupancy rate was significantly increased. When attacked, the occupancy rate of isolated forest algorithm was between 39% and 55%, while the resource occupancy rate of k-means algorithm was between 40% and 60%, and the resource occupancy rate of isolated forest combined with k-means algorithm was between 36% and 45%. After 1990s, the attack ended, and the system gradually returned to a stable state. It can be seen that the model used in this paper can maintain good performance when subjected to DDoS attacks and has more advantages in average CPU utilization.

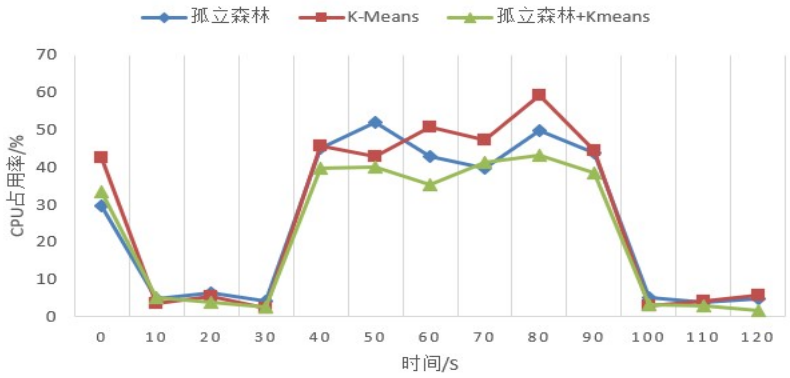

Figure 14. Comparison of CPU utilization

5. CONCLUSIONS

This article through studies SDN DDoS attack detection method in the network, analysis the advantages and disadvantages of machine learning algorithm, k-means clustering speed, but is sensitive to outliers, prone to error, isolated forests to detect fast but lack accuracy slightly, the characteristics of combination of machine learning methods have been put forward the two algorithms together for SDN network under DDoS attack detection. By analyzing the characteristics of the flow table in SDN network, the seven-element aggregation features are extracted from the flow table to identify the flow type and give early warning to the controller in time. In the experimental design, the appropriate algorithm detection model was selected by adjusting K value and model parameter tuning. Meanwhile, the experimental results proved that the detection method combining isolated forest and k-means had good detection effect, with high accuracy and low false alarm rate, and reduced the average CPU utilization rate.

For example, the injection of real traffic is generated randomly by tools. The collected traffic data may be different from the normal data due to network or configuration problems, which may affect the training results of the model. At the same time, the experiment is carried out in a simple virtual network, which may have errors with the actual complex network. In addition,

how to improve the algorithm to reduce the error rate of the system is also worthy of further study. On this basis, real traffic data will be obtained for testing in the future, and the feasibility of this method will be tested in the actual environment, and improvements will be made based on the specific results.

This paper provides some ideas for scholars who are interested in the research of SDN security field. It is believed that in the future, with the research of more scholars, the detection of DDoS attack under SDN network can make a significant breakthrough.

REFERENCES

[1] Yi Zhiling, Cui Chunfeng, Han Shuangfeng, Pan Chengkang, Chen Yami. Analysis of Key Technologies of 5G-Oriented Cellular Internet of Things[J]. Journal of Beijing University of Posts and Telecommunications, 2018, 41(05): 20-25.

[2] Xu Weida. Application Research of Software Defined Network SDN in Operator IP Metropolitan Area Network[D]. Jilin: Jilin University, 2019.

[3] NSFOCUS.《2019 DDoS attack landscape》[EB/OL]. (2019-12-17) [2020-4-2] http://blog.nsfocus.net/wp-content/uploads/2019/12/2019-DDoS.pdf

[4] Niu Shaozhang. Research on Intrusion Detection Based on Clustering and Outlier Detection[D]. Guangdong: Guangdong University of Technology, 2019.

[5] Chen Yu. Research on network traffic anomaly detection method based on combination learning[D]. Hebei: Yanshan University, 2019.

[6] Xu Yuhua, Sun Zhixin. Research Development of Abnormal Traffic Detection in Software Defined Networking[J]. Journal of Software, 2020, 31(01): 183-207.

[7] Li Hefei, Huang Xinli, Zheng Zhengqi. Detection Method of DDoS Attack Based on Software Defined Network and Its Application[J]. Computer Engineering, 2016, 42(2): 118-123.

[8] Li Dong, Zhou Qizhao. Research on DDoS Real-Time Monitoring and Mitigating in SDN Network[J]. Computer Science and Appication, 2019, 9(4): 721-730.

[9] Zhu Jing, Wu Zhongdong, Ding Longbin, Wang Yang. DDoS Attack Detection Based on DBN in SDN Environment[J/OL]. Computer Engineering, 2020, 46(04): 157-161+182.

[10] Meng Qingyue. Research and Implementation of SDN Southbound Security Protection System [D]. Beijing: Beijing University of Posts and Telecommunications, 2019.

[11] Ma Lele, Shu yongan. A DDoS Attack Detection Model Based on Machine Learning Algorithm in SDN Environment[J]. Microelectronics and Computer, 2018, 35(05): 15-20.

[12] He Jiantao. Researches on DDoS Attack Detection and Defense Methods Using Machine Learning in SDN[D]. Anhui: Anhui University, 2019.

[13] Gao Xiaonan. A survey of DDoS attacks in software defined networks[J]. Electronic Technology and Software Engineering, 2019, 155(09): 230-232.

[14] Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou. “Isolation forest.” Data Mining, 2008. ICDM’08. Eighth IEEE International Conference on. IEEE, 2008.

[15] Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou. “Isolation-based anomaly detection.” ACM Transactions on Knowledge Discovery from Data (TKDD) 6.1 (2012): 3.

[16] Zhang Ping, Tang Xinmei. The DDoS Attacking Detection Model of SDN based on Deep Belief Network[J]. Journal of Guangxi University for Nationalities(Natural Science Edition), 2019, 25(01): 80-82.

[17] Yuan Yifang, Li Yan, Chen Xu, Gao Yonglong, Xi Xin. Research on Mobile Police Network Traffic Monitoring Method Based on Isolated Forest Algorithm[J]. Computer Engineering and Software, 2019, 40(12): 229-232.

[18] Wu Guangjian, Zhang Jianlin, Yuan Ding. Automatically Obtaining K Value Based on K-means Elbow Method[J]. Computer Engineering and Software, 2019, 40(05): 167-170.

[19] Zhong Xi, Sun Xiange. Research on Naive Bayes Ensemble Method Based on Kmeans++ Clustering[J]. Computer Science,2019, 46(S1): 439-451.

[20] Chen Jia, OuYang Jinyuan, Feng Anqi, Wu Yuan, Qian Liping. DoS Anomaly Detection Based on Isolation Forest Algorithm Under Edge Computing Framework[J].Computer Science, 2020, 47(02): 287-293.