Computer Science and Information Systems 14(3):769–788 https://doi.org/10.2298/CSIS161217028H

A DDoS Attack Detection System Based on Spark Framework

Dezhi Han, Kun Bi, Han Liu, and Jianxin Jia

College of Information Engineering,
Shanghai Maritime University,
Shanghai 201306, China
{dzhan, kunbi}@shmtu.edu.cn
{jmakg23,onlyoneman}@163.com

Abstract. There are many problems in traditional Distributed Denial of Service (DDoS) attack detection such as low accuracy, low detection speed and so on, which is not suitable for the real time detecting and processing of DDoS attacks in big data environment. This paper proposed a novel DDoS attack detection system based on Spark framework including 3 main algorithms. Based on information entropy, the first one can effectively warn all kinds of DDoS attacks in advance according to the information entropy change of data stream source IP address and destination IP address; With the help of designed dynamic sampling K-Means algorithm, this new detection system improves the attack detection accuracy effectively; Through running dynamic sampling K-Means parallelization algorithm, which can quickly and effectively detect a variety of DDoS attacks in big data environment. The experiment results show that this system can not only early warn DDoS attacks effectively, but also can detect all kinds of DDoS attacks in real time, with low false rate.

Keywords: Distributed Denial of Service (DDoS), Early Warn, Attack Detection, Spark framework, K-Means Algorithm.

1. Introduction

With the high-speed development of Internet, majority users has upgrade the bandwidth especially in some large cities, bandwidth of home users has reached 20M or even higher. Besides, with the popularization of 3G networks and gradual application of 4G networks, mobile internet has entered a booming stage. The rapid growth of private network bandwidth and continuously increasing internet users have posed enormous challenges for network security because the impact will be beyond measure once these high bandwidth network users are controlled by hackers and involved in DDoS (distributed denial of service).

It is indicated in the DDoS attack trend report [2] of Incapsula, a globally renowned CDN service provider, published in 2014 that DDoS attacks increased by 240% in 2014 and the traffic exceeded 100G. In addition, it is pointed out in an recently released analysis report [6] by the company that there are at present about tens of thousands or even millions of dedicated SOHO (small office home office) routers that have become part of BotNet and used by hackers to carry out large-scale DDoS attacks in the present. As found by the survey of losses due to DDoS attacks conducted in 2014 Incapsula, 49% of the DDoS attacks would last for 6 to 24 hours and average economic loss per hour is 40,000 dollars

[5]. During the latter half of 2015, Aliyun security team monitored a total of over 100,000 DDoS attacks, an increase of 32% as compared with that in the first half of 2015. Among them, attacks with a traffic exceeding 300Gbps amounted to 66 times, a rise of 127% than that in the first half of 2015 [7].

Network security incidents occurred frequently in the past two years. On January 21, 2014, DNSPod of Tecent got hijacked, resulting in DNS problems for a large number of domestic users. From December 20 to 21, 2014, a game company with its services deployed at Aliyun suffered DDoS attacks and the peak traffic of 453.8G/s made it the worlds biggest victim of DDoS attacks. On March 26, 2015, GitHub, a famous code hosting site, started to suffer a large scale of DDoS attack, it caused interruption of services in certain areas, and the attack was lasted for over 80 hours. On May 11, 2015, NetEase suffered a new DDoS attack, named, LFA (Link Flooding Attack) [5] which resulted in service interruption of 9 hours and loss of RMB 15 million yuan. Consequently, it is important both in theoretical significance and great economic value to research efficiently and promptly detect, warning, and manage large-traffic DDoS attacks.

In a big data background, highly efficient DDoS attack detection involves computation and processing of massive data, while traditional method of single machine takes much time and cannot meet actual demand. The new distributed stream-oriented computing framework (Spark Streaming) adopts the memory-based parallel computing method, which compared with the traditional computing method based on single-machine file system, significantly enhance the processing data quantity and processing data speed in unit time. Application of Spark Streaming to the real-time analysis system of big data flow network can accelerate the speed and accuracy of detection of DDoS attacks in a big data background.

In this paper, a novel DDoS attack system is proposed to detect DDoS attacks in a big data environment based on Spark framework, which includes 3 main algorithms. Based on information entropy, the first one can effectively warn all kinds of DDoS attacks in advance according to the information entropy change of data stream source IP address and destination IP address; With the help of designed dynamic sampling K-Means algorithm, this new detection system improves the attack detection accuracy effectively; Through dynamic sampling K-Means parallelization algorithm, which can quickly and effectively detect a variety of DDoS attacks in big data environment.The experimental results show that good warning results are obtained and the detection accuracy and speed are obviously superior than traditional DDoS attack detection methods.

The rest of this paper is organized as follows: Section 1 presents the working principle of Spark Streaming; Section 2 describes the DDoS attack warning algorithm design in detail. Section 3 presents the detailed design of improving K-Means parallel algorithm based on dynamics of Spark Streaming; Section 4 introduced the structure and major modules of the DDoS attack detection system. Section 5 presents the simulations and results of proposed DDoS attack detection system; Finally, we conclude this paper in Section 6.

2. Spark Streaming Working Principle

Spark [3], proposed in APMLab in University of California Berkeley, formally opened the source in 2010, became an Apache project in 2013 and a top level project of Apache in

2014. Spark offers solution for the problem of slow computation speed due to storage of intermediate results into the disc during calculation of Hadoop [4]. The ecological system of Spark includes batch processing, stream processing, machine learning, diagram calculating, data analyzing, etc. Compared to Hadoop ecosystem, it is a more comprehensive and suitable distributed computing framework used for big data application scenarios.

RDD [11] (Resilient Distribute Data sets) is not only the core of Spark but also the key for Spark to realizing failure recovery and data dependency. With the simple logic of Lineage, RDD can perfectly solve the dependency between data and data, guarantee good fault tolerance. The RDD can also store intermediate results into the memory which significantly improves the computation speed by reducing disc read and write to the minimum. Especially in iterative computation, the speed is increased by one order of magnitude.

Different from MapReduce in Hadoop, MapReduce of Spark is well packaged into RDD. The operation can be conducted with RDD into two types: transformation and action. The Data in RDD do not exist in their original forms but incorporated in RDD in the forms of their locations; then new RDD can be obtained through different transformation of the data in RDD and we can get the final result action when we perform to start the real calculation.

Spark Streaming [10] is a framework in Spark ecological system used for real-time calculation and its core is also based on RDD. Therefore, it can realize seamless connection with Spark to fuse historical data and real-time data perfectly. The features of Spark Streaming are as follows:

(1) Spark Streaming can realize complex processing logic with short simple codes. Its principle is to divide streaming data into small time intervals (e.g. several seconds), namely, to make the data discrete and transform them into data sets (RDD), then process the RDD in batches and conduct calculation on the RDD, thereby finishing the complex streaming data processing.

(2) Good fault tolerance: Spark Streaming has inherited the fault tolerance feature of RDD. If certain partitions of RDD is lost, computation can be restored based on the lineage information.

(3) Good universality: thanks to the design of RDD, Spark Streaming can realize seamless integration with other modules data of the Spark platform and combine real-time processing and batch processing.

(4) Spark Streaming has external data sources of various types which can be classified into the following two major categories: external file system data (such as HDFS data) and network system streaming data (such as streaming data collected by Kafka, ZeroMQ and Flume). The above features of Spark Streaming make it quite suitable for real-time data analysis against the background of big data.

3. DDoS Attack Early-Warning

It is of great significance to study the DDoS attack early-warning algorithm and early-warning, for they can process the early-warning of DDoS attack, especially in big data environment before DDoS attack do harm to the system, and they will save time for system by eliminating damages to the system caused by large-scale DDoS attack. In this paper,

DDoS attack is early-warned based on abnormal changes of source IP and destination IP information entropy of network data stream.

3.1. Traffic Information Entropy Feature

Entropy is an indicator of diversity and uniformity of the microscopic state which reflects the probability distribution of the system in the microscopic state. It can be seen from the perspective of communication that random interference in a system is unavoidable. Therefore, statistical methods can be adopted to describe characteristics of the communication system. To be specific, take the information source as a collection of random events whose probability of occurrence is similar to uncertainty in the microscopic state in thermodynamics; Calculating probability of occurrence in each information source in the information system to simulate the uncertainty of the system in thermodynamics, thus forming information entropy [12]. Information entropy has similar meaning to entropy in thermodynamics and it is an uncertainty indicator of the information system, which may indicate the amount of information in an information system.

Based on the network traffic information, entropy is defined as shown in Equation (1).

$$H(X) = E[-\log p_i] = -\sum_{i=1}^{n} p_i \log p_i \tag{1}$$

In above Equation (1), X represents an information source symbol which has n values: $X_1 ... X_i ... X_n$, each value corresponding probabilities are: $P_1 ... P_i ... P_n$, since each source symbol appears independent of each other, so there comes to the equation:

$$\sum_{i=1}^{n} p_i = 1 \tag{2}$$

When DDoS attacks are launched, hundreds of bottled machines will send large streams of data packets to the target and the attacker, in order to hide its position, will randomly produce fake source IP addresses for the attacking packets or adopt more advanced reply flood DDoS attacks. In this case, the amount of requests for source IP addresses monitored by the server will drastically increases and the distribution will be more dispersed. Moreover, there will be a large amount of request flow flocking into certain service ports at the server side, and at the same time, the requests distribution for destination IP addresses which monitored by the server and the destination ports will become concentrated increasingly. When it occurs to the DDoS attacking, the information entropy of destination IP and source IP of the data flow that arrived the attacked server, which can reflect the uncertainty of system by calculating information entropy of destination IP and source IP, that also can be used for the DDoS attack warning in large-scale network traffic.

Fig. 1 and Fig. 2 are shown as the experimental and test conditions of the public server for the authors school network center. In the beginning of the first 100 seconds test time, the public servers to be tested will be attacked by traffic DDoS 30GB, which are issued by multiple clients in the laboratory. From the detecting results of the gateway to connect the public server, DDoS attack flow occurred in 100th seconds and it is detected by the system that the information entropy based on the destination IP and source IP occurs significant changes. The information entropy based on destination IP decreases rapidly, while the information entropy based on source IP increases rapidly. The result may certify that when the information entropy can better reflect the DDoS attack, the server receives

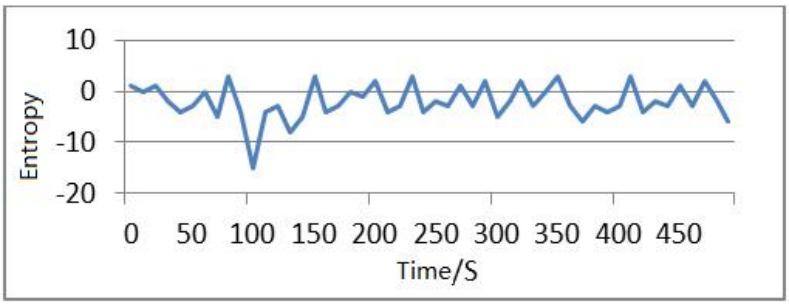

Fig. 1. Information entropy change based on the destination IP.

the uncertainty of the request change range, can be used for the early-warning of DDoS attacks.

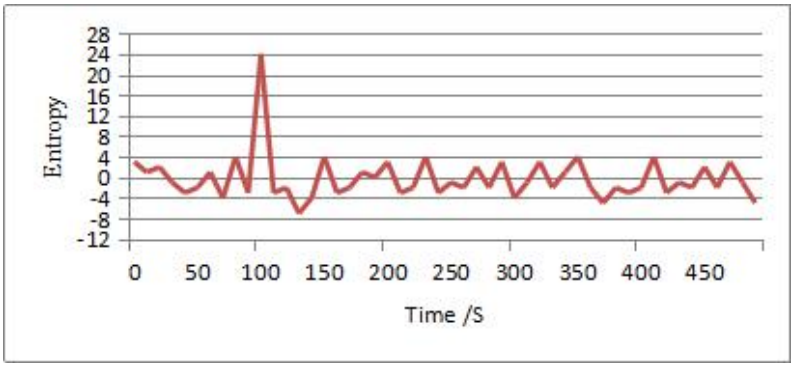

Fig. 2. Information entropy change based on the source IP.

3.2. Design of Network Traffic Model and Early-Warning Algorithm

When DDoS attack happens, the entropy of destination IP and source IP will change largely. Based on the characteristics, the network traffic model was defined. We can analysis the destination IP and sources IP feature on a given time windows based the model. So a DDoS attack early-warning algorithm that based on the information entropy is designed.

First, define a network traffic model, as shown in Fig.3. The traffic model mentioned in this paper includes two kinds of traffic entities, namely Normal (normal request flow) and DDoS (attack flow) under normal circumstances,The detection system collects all the traffic data at a certain timeΔt,and calculated the information entropy of the flow of Δt. Calculate the mean value of the formal flow information entropy of the first $n-1\Delta t$. Calculate the maximum information entropy and mean the difference between the values as an early-warning threshold. When DDoS attacks occur, In theΔt time, the information entropy will change greatly, when the difference of information entropy and the mean value

exceeds the early-warning threshold value, the systems may encounter DDoS attacks and send out the alarm.

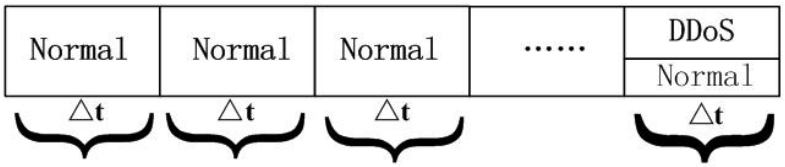

Fig. 3. The flow model.

DDoS attack early-warning algorithm is as follows:

Step 1: StatisticΔt time of all requests, n kinds of different purposes IP (source IP) recorded as X, the number of times per X appears as N.

Step 2: Calculate the probability P of the X emergence.

$$p_i = \frac{N_i}{\sum_{j=1}^{n} N_j} \tag{3}$$

Step 3: Calculate Δt time information entropy $H(X)$.

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{4}$$

Step 4: Calculate the mean value of information entropy of the first $(n-1)\Delta t$.

$$A = \frac{1}{n-1} \sum_{i=1}^{n-1} H(X_i) \tag{5}$$

Step 5: Calculated threshold V, k is the amplification factor, different network environment K value is different, the value is greater than or equal to 1.5.

$$V = (\mathrm{Max}[H(X)] - A) \times k \tag{6}$$

Step 6: Calculate the difference value in Δt, between information entropy and mean.

$$S = H(X) - A \tag{7}$$

Step 7: if $S >= V$, this means issue a DDoS attack alert, and the detection system will start calling DDoS detection module; if $S < V$, this means the entropy change in the normal range, and the network traffic is normal.

There are two key parameters in the network traffic model and early-warning algorithm:

(1) Δt settings, according to the characteristics of DDoS attacks, Δt can be set between 1–10 seconds. The smaller time requests the greater calculating amount when the attacks are detecting, and the detection and treatment effect of DDoS attacks are better;

(2) The calculation of early-warning threshold V, when calculating the V, the network flow and the peak period of network traffic should be fully considered. The key is to set the amplification factor K which can set the value between 1.5 and 2.2. According to experience, the value will automatically set to 2.

4. DDoS Attack Detection

Under the big data environment, traditional single-machine processing methods are not competent to solving the high-speed DDoS attacks because it will cost a great deal of time. The unique RDD internal access mechanism in Spark platform and support provided from Spark Streaming modules for real-time processing will effectively solve the attacking problems caused by DDoS attacks with huge and real-time data flow. Therefore, this paper utilizes K-Means clustering algorithm belonging to category of machine learning and data mining. Besides, improvements to K-Means proposed in this paper will make it suitable for dynamic sampling and parallelization environment, and make it be able to merge sufficiently with Spark Streaming modules of Spark platform. Thus it can adapt to detecting various high-speed DDoS attacks under the big data circumstance.

4.1. Data Preprocessing and Feature Extraction

Faced with a large amount of requested data, DDoS detecting system cannot perform machine learning determination. The data flow texts are produced from diverse networking protocols. However, the detecting algorithm, based on machine learning, requires entering feature vectors including fields with special meanings. Since the proper values must express features of relative requests efficiently and accurately, it is required to carry out pretreatment to request flows. In dimensions of time, space and protocol type, quantification of data flow can make machine recognize and process data. Because data flow of DDoS attack presents strong dependency, certain features describing total flow can be obtained by analyzing existing relationship between current link and before links. On the basis of the features, the thesis will adopt K-Means clustering algorithm to build detecting model of DDoS attack and design related algorithms. According to the features of data flow, the feature extract can be carried out from two parts. The first part is statistics analysis of links during past period t which have the same destination host as current link; the second part is statistics analysis of links during past period t which have same services as current link.

The traffic statistics based on the time are just statistics in the $T1$ time period of the connection, of which relationship refers to the relationship between the other connections in this period and the current connection. In the actual DDoS attack, attackers sometimes use slow attack methods to scan IP and ports.When slow attack scanning frequency is greater than t, the method of time-based traffic statistics cannot get contact between requests.

In this paper, we use a time window to statistics that, in the time window N a current connection with the previous N connection information and set connection information as a feature. According to the characteristics of the specific set of 10 characteristic variables, these characteristic value variables include as follows:

(1) $x1$ represents the number of the current connection with N connection with the same target host, and the value ranges from 0 to 255.

(2) $x2$ represents the number of the same services for the current connection and previous N connections with the same target host, and the value ranges from 0 to 255.

(3) $x3$ represents the ratio of the same service to the current connection and before the N connection has the same target host, and the value ranges from 0 to 1.

(4) $x4$ represents the ratio of the current connection to the previous N connection with the same target host different services, and the value ranges from 0 to 1.

(5) $x5$ represents the ratio of the current connection to the same source port of the previous N connection with the same target host, and the value ranges from 0 to 1.

(6) $x6$ represents the ratio of the same service to the same service as the previous N connection, which is the same as the host, and the value ranges from 0 to 1.

(7) $x7$ represents the ratio of SYN error in links with same as the destination host the same service between current links and the former N links, and the value ranges from 0 to 1.

(8) $x8$ represents the ratio of SYN error in links with same destination host between current links and the previous N links, and the value ranges from 0 to 1.

(9) $x9$ represents the ratio of REJ error in links with same destination host between current links and the previous N links, and the value ranges from 0 to 1.

(10) $x10$ represents the ratio of REJ error in links with same as the destination host the same service between current links and the previous N links, and the value ranges from 0 to 1. By pretreating and extracting of the characteristic value of the normal network data, it can be trained to detect K-Means clustering model and design K-Means clustering algorithms of DDoS attack detection.

4.2. K-Means Clustering

The detecting objective of DDoS attack is to distinguish normal access request flow from abnormal attack flow; in nature, it is a kind of cluster. K-Means is a classic type of objective function clustering algorithm of LAN prototype, which belongs to category of unsupervised learning. In 1967, it was firstly put forward by James MacQueen and then it was popularized in various machine learning fields. The core idea of the algorithm is as follows: firstly, to select k objects at random and every initial object shows the center or average value of a cluster. After successive traversal, distances from the surplus objects to centers of all clusters will be calculated. Then by the comparison of the distances, they will be distributed to center with the smallest distance and calculations of all centers will be performed again. Next repeat the process until the convergence of clustering criterion function. The algorithm flow chart is shown in Fig. 4. The detailed description of the algorithm is as follows:

Input: K, D (Initial sample data)

Output: K clustering centers

Step1: Data set D as the initial sample, the n-dimensional of each point: $d_j = \{x_1, x_2, x_3, ..., x_n\}$. Each one dimension represents a feature vector. Random selection of K objects as initial cluster centers from data set D, the cluster center set is denoted as K.

Step2: Calculate the distance from each point in the D to the K cluster center, according to the minimum, assign the point to the corresponding category, cluster centers corresponding data is denoted .Using Equation (8) to calculate the Euclidean distance.

$$D(k,d) = \sqrt{\sum\nolimits_{i=1}^{n} (x_{ki} - x_{di})^2}, k \in K, d \in D \tag{8}$$

Step3: Cluster center of updated cluster.

$$k = \frac{1}{n} \sum\nolimits_{i=1}^{n} c_i, c_i \in C, k_i \in K, n = \text{Size}(C_k) \tag{9}$$

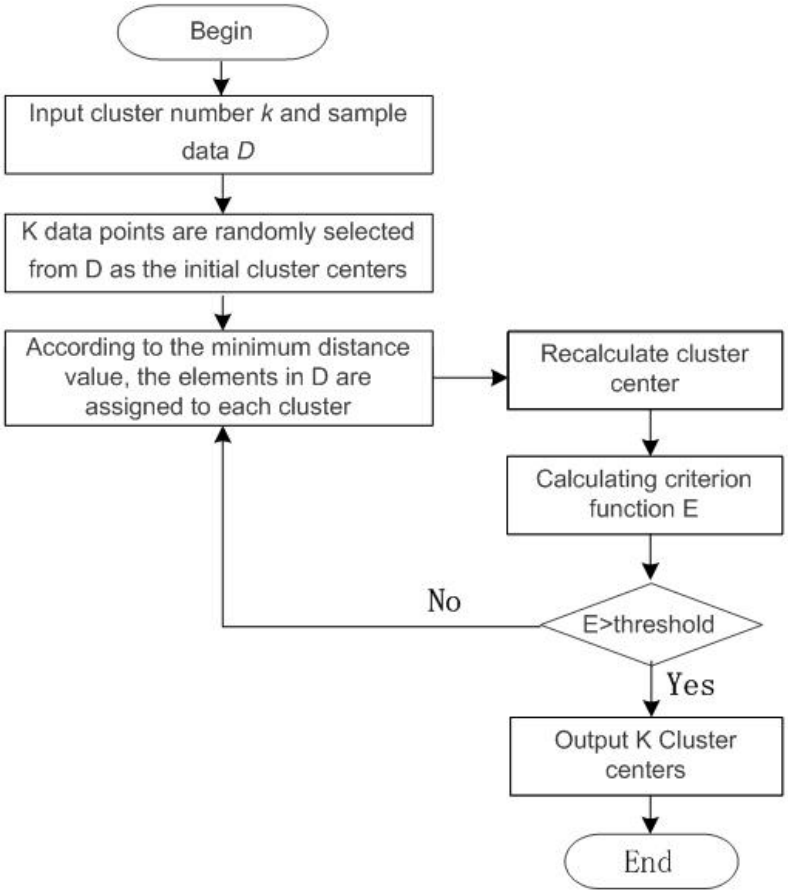

Fig. 4. K-Means flow

Step 4: Calculation criterion function.

$$E = \sum_{i=1}^{k} \sum_{c_j \in C_i} \left(c_j - k_i\right)^2, c_i \in C, k_i \in K \tag{10}$$

Step5: Meet the threshold criterion function exit, otherwise return to step 2.

4.3. Dynamic Sampling K-Means Algorithm

In common clustering algorithm, k points will be selected randomly as the center and the value of k as well as the selection of initial center will have direct influence on consequence of final cluster. If k is selected inappropriately, K-Means algorithm would converge on locally optimal solution, with the result that the correct result would not be obtained. In the DDoS attack detection system, K-Means requires to process a great deal of data mixing with attack flow, which leads to great difficulty to the selection of initial center. In order to solve the problem, dynamic sampling K-Means cluster will be employed to improve the algorithm to meet the demands of DDoS attack detecting system. The algorithm is shown Fig.5.

The main way to improve K-Means algorithm is to select only one point in advance as clustering center to build scale function. The function represents quadratic sum of distance between data point and its clustering center. Then, the clustering results will be converged by continuous iteration of minimum function value. The main theory for the improvement of K-Means algorithm is as follows: firstly, select a point from data set as initial clustering center and add it into dynamic sampling set C, which can be calculated by scale function, then perform circulation N times; secondly, select m points during each circulation and calculate sampling probability $P(X)$ .The meaning of the probability shows that clustering center is easy to be another center when it is more far away from original center because it is relatively disperse. In other words, the selected points should be far away from current clustering center. After iteration, the function value should be calculated again and it is required to update sampling probability for the next time. Afterwards, the overlaps between central point set C of sampling cluster and original sampling set C will act as new sampling set. After the N circulation, a new sampling set C will be produced which has several data. The scale of current data set is far smaller than that of original X and the data are relatively centralized due to the reason that they are filtered. Finally, the common K-Means algorithm of C will be performed and the process will be extremely fast because C is obtained after processing in advance. Meanwhile, the algorithm is improved in time complexity. It adopts the method of iteration replacing convergence threshold and reduces times of iteration, which is important to inspect DDoS attack under the environment of big data by machine learning method.

The specific algorithm is defined as follows:

Definition 1: The scale function $V(X)$ is defined as the formula (11). Where the $D^2(X, C)$ represents the square sum of the distance from the point in the X to the cluster center.

$$V(X) = \sqrt{\sum_{i=1}^{n} D^2(X, C)} = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{d} \left(x_j - x_c\right)^2} \tag{11}$$

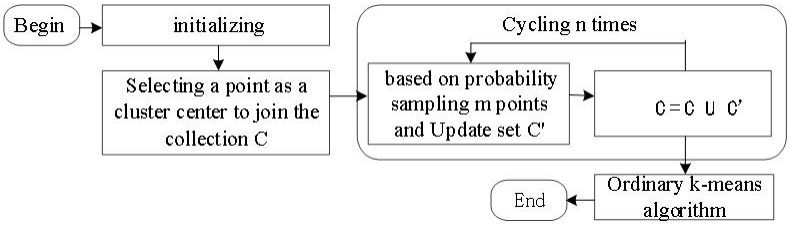

Fig. 5. Dynamic sampling improved K-Means algorithm flow.

Definition 2: The dynamic sampling probability function $P(X)$ is defined as the formula (12).

$$P(X) = \frac{D^2_{\min}(X, C)}{V(X)} \tag{12}$$

Definition 3: initial limited scale function value V, initial sample number $m < k$.

Specific algorithm is as follows:

Input: data set, K

Output: K clustering centers

Step 1: Randomly select one point from the set X to join in the set C.

Step 2: According to formula (11) to calculate the initial limited scale function value of the C, denoted as V.

Step 3: Cycle $\log V = N$, calculate the dynamic sampling probability $P(X)$ according to equation (12), recorded as P. Take out m points from the set X in accordance with the probability of P to join the collection C', calculate $C \bigcup C'$ and denoted by C, end of the cycle.

Step 4: Calculate the clustering center of the set C by using a common K-Means algorithm.

4.4. An Improved K-Means Algorithm for Dynamic Sampling Based on Spark

Ordinary DDoS attack detection algorithm cannot run directly on the Spark platform, according to the principle of Spark, the design of dynamic sampling and improved K-Means algorithm. The specific process is as follows:

(1) Algorithm begins, Master node program obtain the initial data set from the data input source, which is a predefined interface that can obtain data through a variety of ways, such as InputStream, HDFS, local files, etc., this design is convenient for the test of the algorithm. After obtaining the data, the system will convert the data to RDD1, and call the cache method to load the RDD1 to memory, the RDD will act as the data to be processed.

(2) Carry on the segmentation of data, to prepare for the parallelization. The system takes the block as a unit (64MB) to divide the RDD1 into several sub blocks. Then the master node calls the map method, and the large data blocks are allocated to multiple Worker nodes. When worker node receives the data blocks and executes the map instruction of Master, processing the data block. After this step, the String text of the original data set will be converted to DenseVector objects, which are the data that the program can

use directly; the distribution of the data is calculated on each Worker node. When the map method is finished, the RDD1 generates a new RDD2.

(3) Randomly select the initial cluster centers. The program calls the takeSample method, selects one of the RDD2 as the clustering center vector, and creates the RDD3 object.

(4) Begin to enter the cycle process, the program according to the 4.3 section of the implementation of the specific algorithm step 3 to carry out an iterative calculation. In each cycle, according to the definition 1 and definition 2 to recalculate the current sampling probability function P, and then call the takeSample method according to the probability P select the new RDD vector as the center point. After a cycle, the sampling total vectors are $1 + m$, and generate RDD4. Then the system calls the union method, the RDD3 and RDD4 merged into RDD5.

(5) After $\log V$ times will end the cycle, at this time, the number of vectors in the RDD5 is not more than $1 + m * \log V$. This amount is far less than the amount of initial data.

(6) The system will output RDD5 as a result.

The RDD conversion process of the entire sampling phase is shown in Fig.6. In Fig.6, the rounded rectangle frame represents the RDD; the straight rectangular box in the RDD represents the data fragmentation in the RDD, which is spread on a different Worker node; the direction of the arrow indicates the process of the RDD conversion.

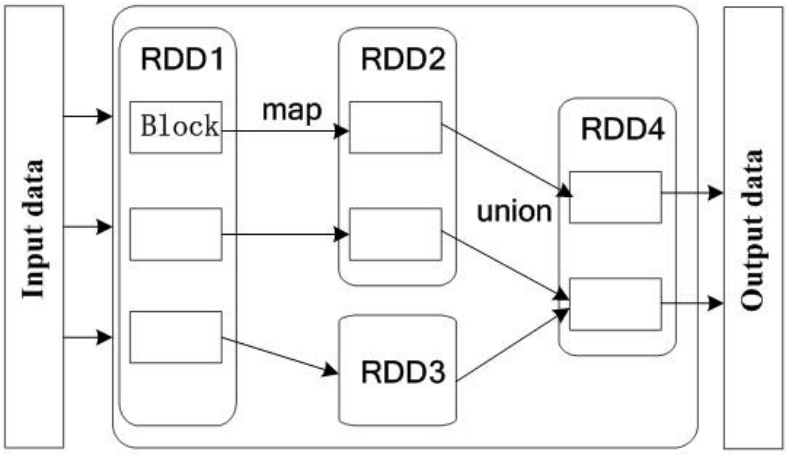

Fig. 6. RDD conversion process of sampling phase.

5. DDoS Attack Detection System

The software structure of DDoS attack detection system based on Spark is shown in Fig.7. The whole system is divided into four modules. These four modules are running on the nodes of Spark cluster and work together to complete the DDoS attack detection.

The detailed design of DDoS attack detection system is shown in Fig. 8. The whole system is running on a distributed cluster. It can not only make full use of Spark technology in management of distributed computing, but also improves the reliability and the processing speed of the system.

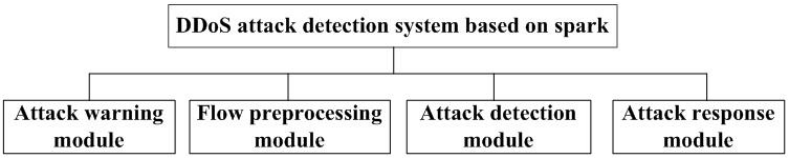

Fig. 7. Structure of DDoS attack detection system based on Spark.

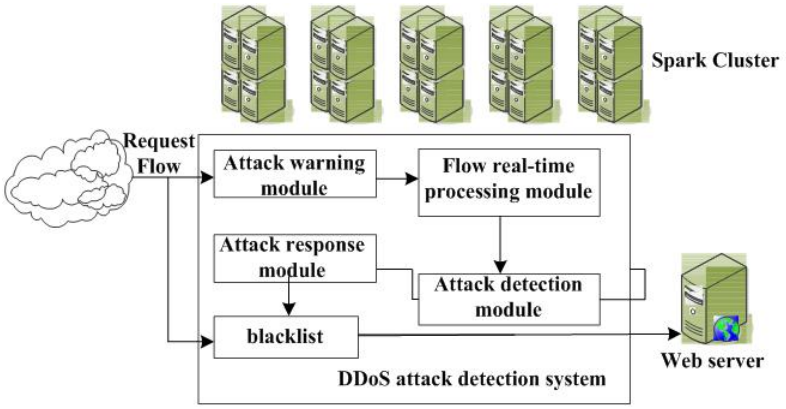

Fig. 8. Frame of DDoS attack detection system.

The system is comprised by attack warning, flow preprocessing, inspection and attack response modules. Attack warning module adopts the early-warning algorithm based on flow information entropy; flow real-time processing module mainly processes warning data flow section by section upon real-time processing framework called Spark Streaming, picking up relative characteristics and outputting the characteristic data to modules of attack detection in order to detect DDoS attacks; attack detecting module adopts DDoS detection algorithm similar to Spark, mainly receiving data from flow processing module, recognizing DDoS attacks according to clustering results and outputting results to module of attack response; the attack response module adds original IP address of DDoS flow detected by attack detecting module into blacklist, then the detecting system will filter attack flow in the list.

6. Experimental Results and Analysis

6.1. DDoS Attack Warning Algorithm Test

The test based on DDoS alarming algorithm of information entropy in the thesis, will be implemented on an e-commerce website. The website uses 3 nodes and each machine employs 8 core CPU with16GB internal storage and 1TB hard disk drive. In terms of software configuration, Spark 1.5.2 version is used to process big data and Java 1.8 version to compile program. The website has a quite large amount of information about access behaviors of users and daily records of access behavior reach over 16 million. The warning test period lasts a week from 8:00 am to 20:00 pm. DDoS attack uses software Autocrat [1] to perform SYN, LAND, FakePing and Furious Ping attack respectively. The test result is that the average warning rate reaches 98.5% while the average error warning rate is only 1.6%. For network server being in peak period, the error warning is mainly caused by various interferences.

6.2. DDoS Attack Detection Algorithm Test

In order to test the improved K-Means algorithm for dynamic sampling, a set of contrast tests under the situation of single-machine operation is designed in the thesis. Firstly, Java language programming is used to verify K-Means algorithm. Meanwhile, the algorithm is adopted to perform clustering analysis of test data and work out the required time for calculation and accuracy of clustering. Secondly, Java language programming is employed to verify the improved K-Means algorithm for dynamic sampling. Besides, the same test data is used to carry out clustering calculation to count the required time and accuracy. The test data is selected from training set with intensive kddcup-99 [8] data. Whats more, the data is also filtered and eight classical properties from the original properties are selected as properties of test data. Totally, 5 groups of data is selected and their data is respectively 10000 for group 1, 50000 for group 2, 100000 for group 3, 200000 for group 4 and 500000 group 5. Data for each group is different in figure but similar in distribution, which is used to perform a contrast test. The results of the test are shown in Fig. 9 and Fig.10.

From Fig.9 and Fig.10, we can see that when the amount of data is less, the dynamic sampling of the improved K-Means algorithm and the common K-Means algorithm is very close to the time. With the increase of the training set size, the advantage of the improved K-Means algorithm of dynamic sampling becomes increasingly distinct. In the case of 500 thousand data sets, the improved algorithm is obviously superior to the ordinary algorithm in time complexity. In terms of accurate rates, the improved K-Means algorithm of dynamic sampling is relatively close to ordinary K-Means algorithm. And the accurate rates in different test sets fluctuate but the fluctuation maintains in a relatively stable range.

In order to test the detection speed and accuracy of the proposed detection algorithm on the Spark cluster, the following experiments are designed: Using the KDD99 data set of the training works (5 million data) as the experimental samples, respectively, 5 groups of data are selected. These data are respectively 1 ten thousand for group 1, 50 ten thousand for group 2, 1 million for group 3, 2 million for group 4 and 5 million for group 5. And

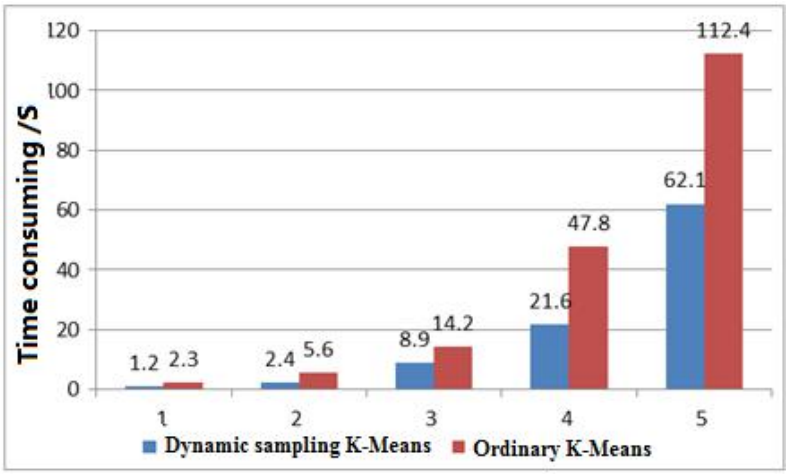

Fig. 9. Comparison of the time-consumption of two algorithms in five experiments.

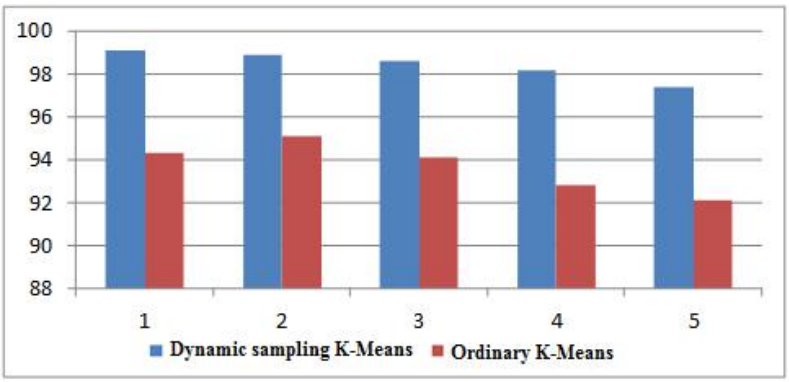

Fig. 10. Comparison of the accuracy rates of two algorithms in five experiments.

Spark cluster adopts 1.5.2 Spark version for the configuration of the software, while Java 1.8 version is used for the preparation of the Spark program.

Three experimental groups are designed in the experiment with the first experimental group using a single algorithm, serial processing of data samples; second experimental group using ordinary K-Means algorithm, parallel processing of data samples and the third experimental group using the improved K-Means algorithm.

Analysis is made on the time consumption, the average time of each round of iteration, and the correct rate in the three experimental groups. The final results are shown in Fig. 11 and Fig. 12.

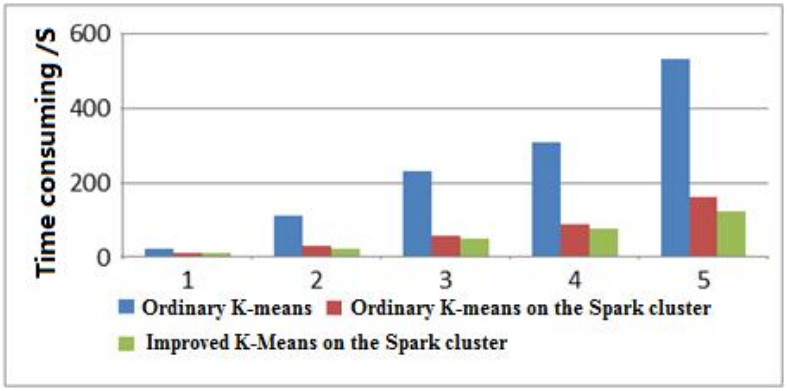

Fig. 11. Comparison of time-consumption for three experimental groups in five experiments.

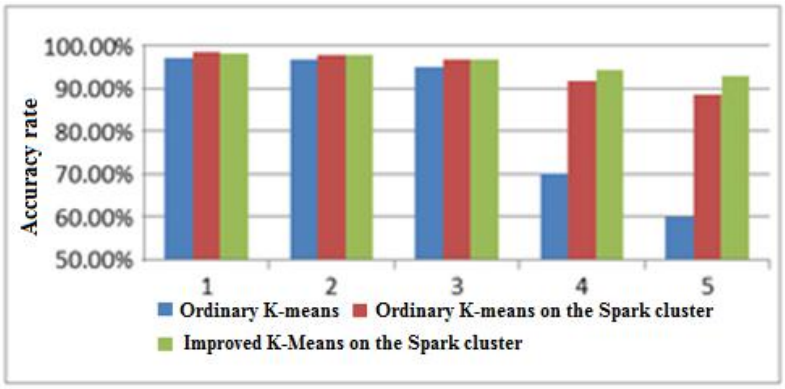

Fig. 12. Comparison of accurate rates for three experimental groups in five experiments.

Through the comparison of the above results, in the case of a small amount of data, it is found that the time difference between the three algorithms is not very large and the ac-

curacy of single machine operation is relatively high, and the advantage of Spark parallel computation is not obvious. When the amount of data is greater than 1million, the time to run a single machine increases dramatically while the accuracy of the data decreases rapidly. At the same time, the advantage of Spark parallel computing is very significant. Compared to the ordinary K-Means algorithm implemented on the Spark cluster, the improved K-Means algorithm has better accuracy and efficiency. This experiment can better reflect the advantage of parallel computing based Spark dynamic sampling platform to achieve an improved K-Means algorithm.

In order to test the system's ability to deal with DDoS attacks, this article through the open source software simulates the large data traffic DDoS attack [1], and starts the detection system to detect and address it. The Experimental design is to launch the attacks on the Web Service that has set up the DDoS attack detection system and the web service that did not build the DDoS attack detection system respectively. The actual impact of DDoS attacks on the server is determined by calculating the Web Service real-time throughput and CPU utilization rate. The final experimental statistics are shown in Fig. 13 and Fig. 14.

As is shown in Fig. 13 and Fig. 14, the throughput of the server increases rapidly after the DDoS attack within 100 seconds, after which, the throughput of the server in experimental group 1 without DDoS detection system falls sharply with CPU occupancy rate close to 100% whereas that of the server in experimental group 2 with DDoS detection system remains at normal level. Thus, it is proved that Web Service without detection system cannot continue to provide the normal service while the one with the detection system still can operate normally when confronted with DDoS attacks.

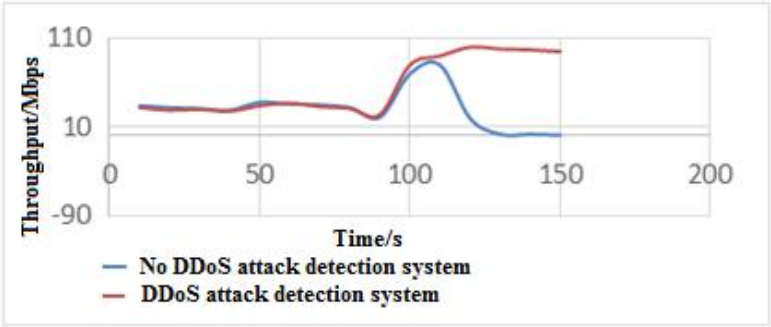

Fig. 13. Throughput Comparison.

6.3. Comparison with the classical DDoS detection method

In order to effectively analyze the performance of the proposed method, the simulation experiment is also used in the training of KDD 99 data sets (5 million data) as the experimental samples. 5 groups of data are selected and their data is respectively 1 ten thousand for group 1, 50 ten thousand for group 2, 1 million for group 3, 2 million for group 4 and 5 million for group 5. Three classical DDoS detection methods are selected after the

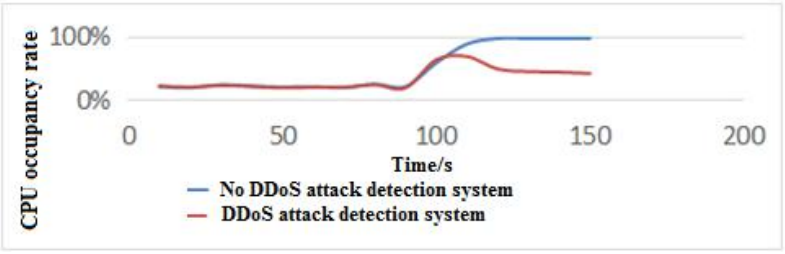

Fig. 14. Occupancy rate comparison.

experiment comparison [9], namely, DDoS attack detection method based on Hurst parameter(indicated by "DC1"), DDoS attack detection method based on nonlinear network flow analysis (DC2) and Wavelet analysis method based on adaptive detection of DDoS attacks ((indicated by "DC3"). By comparing these three classical algorithms with the dynamic improved K-Means method based on Spark in this paper (by "DC4") concerning the average response time, the average recognition rate, the average false rate, the results of the four methods are demonstrated in table 1. According to table 1, the method proposed in this paper is superior to the classical DC1, DC2 and DC3 methods in terms of average response time, average recognition rate, and average false rate.

Table 1. The performance comparison of DDoS detection algorithms.

Comparison Algorithm	DC1	DC2	DC3	DC4
Average response time	6.61	2.21	1.83	0.62
Average recognition rate	87.23 Apache15	93.12	91.64	98.3
Average false rate	3.52	2.13	2.25	1.5

7. Conclusions

In the big data environment, DDoS attacks are becoming one of the biggest threats to network security. Based on the existing researches, this paper designs a DDoS detection system based on Spark, to ensure accuracy in detection. In the meanwhile, the time for detecting DDoS attacks is reduced and the detection efficiency is improved significantly with the advantage of Spark technology.

In the future research work, the following aspects need to be improved:

(1) Spark Framework version iteration is very fast and each version will have new content and more powerful features. In the future research work, we should use the new features of the Spark framework flexibly to improve the efficiency of the system.

(2) For distributed systems, parameter setting is essential. In the future research work, we should do in-depth research in parameter tuning of the Spark framework to improve DDoS attack detection efficiency in big data condition.

(3) The limitation of this research is that it does not study much on tracking attackers in the DDoS detection. In order to prevent DDoS attacks more effectively, the method of investigating the legal liability of the attacker through internet forensics will be studied.

Acknowledgments. This work has been supported by the National Natural Science Foundation of China (No. 61373028 and No. 61672338).

References

1. Ddos attack using common tools (2013), [Online]. Available: http://www.bingdun.com/news/bingdun/8576.htm
2. Incapsula.report:2014 ddos trends-botnet activity is up by 240% (2014), [Online]. Available: https://w-ww.incapsula.com/blog/ddos-threat-landscape-report-2014.html
3. Apache software foundation. apache spark-lightning-fast cluster computing (2015), [Online]. Available: http://spark.apache.org/
4. Apache software foundation. welcome to apache hadoop (2015), [Online]. Available: http://hadoop.apache.org/
5. Incapsula.ddos impact survey reveals the actual cost of ddos attacks (2015), [Online]. Available: htt-ps://www.incapsula.com/blog/ddos-impact-cost-of-ddos-attack.html
6. Incapsula.lax security opens the door for mass-scale abuse of soho routers (2015), [Online]. Available: https://www.incapsula.com/blog/ddos-botnet-soho-router.html
7. Cloud shield internet ddos state and trend report in the second half of 2015 (2016), [Online]. Available: http://wenku.baidu.com/view/747d352f0c22590103029d6f
8. Hettich, S., Bay, S.D.: Kdd cup 1999 data (1999), [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup-99/kddcup99.html
9. Peng, Y., Yan, L.: Approach of ddos attacks prediction and detection with heqpso-svm algorithm based on date center network. Journal of Chinese Computer Systems 36(1), 150–163 (2015)
10. Zaharia, M., Das, T., Li, H., Shenker, S.: Discretized streams: An efficient and fault-tolerant model for stream processing on large clusters. In: HotCloud. pp. 141–146. ACM (2012)
11. Zaharia, M.: Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing. in-memory cluster computing. In: USENIX Symposium on Networked Systems Design and Implementation (NSDI), pp. 141–146 (2011)
12. Zhao, H.: DDOS anomaly detection technology research based on the information entropy clustering. Ph.D. thesis, Central South University, Changsha, China (2010)

Dezhi Han (corresponding author) received the Ph.D. degree from Huazhong University of Science and Technology. He is currently a professor of computer science and engineering at Shanghai Maritime University. His research interests include cloud computing, mobile networking and cloud security.

Kun Bi received the Ph.D. degree from University of Science and Technology of China. He is currently a lecture of computer science and engineering at Shanghai Maritime University. His research interests include network security, big data and cloud security.

Han Liu received the M.S. degree from Shanghai Maritime University of computer science and engineering. He is currently a Ph.D. candidate at Shanghai Maritime University. His research interests include cloud computing and cloud security.

Jianxin Jia received the M.S. degree from Shanghai Maritime University of computer science and engineering. He is currently pursuing the Ph.D. degree at Shanghai Maritime University. His main research interests include mobile networking, underwater acoustic communication technology and cloud security.

Received: December 17, 2016; Accepted: August 20, 2017.