Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 5202836, 10 pages
https://doi.org/10.1155/2017/5202836

Research Article

Multiple-Features-Based Semisupervised Clustering DDoS Detection Method

Yonghao Gu,[1] Yongfei Wang,[1] Zhen Yang,[1] Fei Xiong,[2] and Yimu Gao[3]

[1]*Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia, School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*State Grid Information & Telecommunication Branch Company, Beijing 100761, China*
[3]*Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716, USA*

Correspondence should be addressed to Yonghao Gu; guyonghao@bupt.edu.cn

Received 18 May 2017; Accepted 8 October 2017; Published 17 December 2017

Academic Editor: Federica Caselli

DDoS attack stream from different agent host converged at victim host will become very large, which will lead to system halt or network congestion. Therefore, it is necessary to propose an effective method to detect the DDoS attack behavior from the massive data stream. In order to solve the problem that large numbers of labeled data are not provided in supervised learning method, and the relatively low detection accuracy and convergence speed of unsupervised k-means algorithm, this paper presents a semisupervised clustering detection method using multiple features. In this detection method, we firstly select three features according to the characteristics of DDoS attacks to form detection feature vector. Then, Multiple-Features-Based Constrained-K-Means (MF-CKM) algorithm is proposed based on semisupervised clustering. Finally, using MIT Laboratory Scenario (DDoS) 1.0 data set, we verify that the proposed method can improve the convergence speed and accuracy of the algorithm under the condition of using a small amount of labeled data sets.

1. Introduction

A denial-of-service attack (DoS attack) is a cyber-attack where the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the Internet. Denial of service is typically accomplished by flooding the targeted machine or resource with superfluous requests in an attempt to overload systems and prevent some or all legitimate requests from being fulfilled [1]. A distributed denial-of-service (DDoS) is a cyber-attack where the perpetrator uses more than one unique IP address, often thousands of them. The scale of DDoS attacks has continued to rise over recent years, by 2016 exceeding a terabit per second [2]. DDoS attacks are distributed that an attacker initiated this attack by manipulating distributed Internet agent host of different locations at the same time. When the attack stream from different agent host converged, the stream at victim host will become very large and will soon become system halted or network congestion [3].

DDoS attacks can be performed by a large group of cooperating people, a small group of people, or a single person that controls one or more sufficiently powerful botnets. All types of motivations can lead to an organized attack: political and social issues are among the top motivations, but any public or private institution or company can be a victim because small groups or individual criminals usually have specific targets which are chosen based on revenge, competition, or simply the desire to cause damage [4]. Therefore, it is necessary to propose an effective method to detect the DDoS attack in the massive data stream.

In this paper, we propose a novel method for DDoS attack detection, which is based on multiple-features-based semisupervised clustering algorithm, and the provided method uses only small amount of labeled data and relatively large amount of unlabeled data to detect DDoS attack behavior. Compared with previous detection solutions based on supervised learning and unsupervised learning methods, our proposed algorithm has the following advantages:

(i) Compared with supervised learning detection algorithms, our method requires fewer labeled data sets to training detection models.

(ii) Compared with unsupervised learning detection algorithms, our method has higher detection accuracy and can improve the convergence speed of the model (reduce the time complexity of the algorithm).

The remainder of this paper is organized as follows. Section 2 introduces the related works in detecting DDoS attacks and analyzes their shortcomings. Section 3 describes the selected features used in the proposed detection method. In Section 4, our detection model is provided, which is based on semisupervised clustering algorithm. Section 5 shows the experimental details and gives the experimental results and analyses, which is followed by conclusions and future work.

2. Related Works

There are two classes of DDoS detection techniques: misuse detection and anomaly detection [5]. The misuse detection techniques try to detect attack by comparing the current activity of destination network to a database of known attack signatures. These techniques cannot detect unknown attacks, while the anomaly detection techniques try to detect attack by comparing the current activity of destination network to an established normal activity represented as a profile.

In recent years, machine learning algorithms are often used in anomaly detection. Machine learning methods mainly include supervised learning and unsupervised learning [6]. Classification and regression problems belong to the category of supervised learning. Commonly used classification algorithms include decision tree classification [7], naive Bayes classification algorithm [8, 9], Support Vector Machine (SVM) classifier [10], Neural Network method [11], and k-nearest neighbor (kNN) [12]. The problem of association and clustering belongs to the category of unsupervised learning, and k-means algorithm is the most commonly used clustering algorithm [13].

Liao and Vemuri [14] used K-nearest neighbor classifier (KNNC) to categorize process into normal or intrusive class. The KNNC calculates the similarity between the new process and each training process instance and basically assumes that the processes belonging to the same class will cluster together in the vector space. It is excellent in attack detection, but the detector is computationally expensive for real-time implementation when the number of processes simultaneously increases. Support Vector Machine (SVM) is a technique based on machine learning, where data is classified by determining a group of support vectors and characteristics to be quantified are described. As proposed by [15, 16], the Hybrid Intrusion Detection System (HIDS), based on machine learning and specifically the SVM technique, improves the detection rate. More recent study as [11] presents a better classification using an Artificial Neural Network (ANN) to flag detection engine known and unknown attacks from genuine traffic. Ramos et al. [12] use k-NN classifier method and cosine formula based algorithm to detect the DDoS attack, but this method needs some time to train the original packets. Xiao et al. [17] present a detection approach based on CKNN (K-nearest neighbors traffic classification with correlation analysis) to detect DDoS attacks. The approach exploits correlation information of training data to improve the classification accuracy and reduce the overhead caused by the density of training data. Öke et al. [18] used multiple Bayesian classifiers to detect DDoS attacks. However, naive Bayes are based on a very strong independence assumption, which is not always satisfied. Amor et al. [19] compared the performance of naive Bayes with C4.5 decision tree and find the good performance of Bayes with respect to existing best results performed on KDD'99.

Clustering algorithm mainly includes two categories: hierarchical and partitioning [20]. Partitioning method is inappropriate for our case because the number of clusters should be predetermined in partitioning. Therefore, the paper adopts a hierarchical method. This method is often used to classify plants and animals and is expected to be adequate for classifying the phases of the DDoS attack by the use of their features. In clustering, the learning algorithm finds similarities among instances to build the clusters (i.e., group of instances). Instances that belong to the same cluster are assumed to have similar characteristics or properties and then are assembled into the same class. K-means algorithm belongs to partitioning one, which has been successfully used to detect anomalies [13] and DDoS [21], using clustering methodologies to formulate the normal patterns, since one of the advantages of clustering methods over statistical methods is that they do not rely on any prior known data distribution. But machine learning based techniques require a lengthy learning period and hence currently these methods cannot operate in real-time [22].

Many advantages and disadvantages related to the above machine learning algorithms with anomaly detection have been reported by many researchers [23, 24]. Supervised learning has the advantage to achieve better accuracy to classify similar examples. But, one shortcoming of supervised learning is the need for large scale labeled instances. This raises ambiguity about the performance of supervised learning, since it requires a sufficient amount of labeled data to train the classifier [25]. Unsupervised learning techniques deal with the learning tasks with unlabeled or untagged data, and clustering is the most popular unsupervised learning technique [26]. They have the advantage of detecting new examples better than supervised learning techniques and are considered to be more robust in IDSs. However, the disadvantage of unsupervised learning is the manual assignment of cluster numbers, which results in relatively low accuracy in predictions. In case of unsupervised learning, large amount of uncertainty is associated with modeling the data set. In addition, for the typical unsupervised learning algorithm k-means, the selection of k value and the initial clustering centers have great influence on the clustering accuracy and the convergence speed of the algorithm.

In order to integrate the advantages of supervised and unsupervised learning methods, and considering the actual application scenes which have small amount of labeled data and relatively large amount of unlabeled data, this paper provides a semisupervised clustering method to detect DDoS attacks.

3. Feature Selection

DDoS attack has the following characteristics. (1) Attack flow distribution is obvious. Traffic flows are always from distributed sources to one target or limited targets. (2) The proportion of unsuccessful two-way connections is large. Under normal circumstances, the proportion of successful two-way connection is high. When DDoS attack occurs, both sides of communication cannot establish a normal connection because the attackers use spoofed IP addresses and the proportion of unsuccessful connections increases rapidly.

3.1. Why Use One-Way Connection Density? In a given time window, the proportion of one-way connection packets to all data packets is called the one-way connection density (OWCD) [27]. In the IP data stream, if both request packets and reply packets from each side are received, this connection is described as a two-way connection (TWC). If no reply packet is received, we call it one-way connection (OWC).

Three most common DDoS attacks are TCP flood, UDP flood, and ICMP flood. TCP flood, also known as SYN flood, is a form of denial-of-service (DoS) attack in which an attacker sends a succession of SYN requests to a target's system in an attempt to consume enough server resources to make the system unresponsive to legitimate traffic [28]. A UDP flood attack is a denial-of-service (DoS) attack using the User Datagram Protocol (UDP), initiated by sending a large number of UDP packets to random ports on a remote host. Thus, for a large number of UDP packets, the victimized system will be forced into sending many response packets, eventually causing it to be unreachable by other normal clients [29]. ICMP flood is one kind of DoS flood, by using ICMP packets to consume enough of computing resource to slow down the target system. Ping flood is a typical ICMP flood attack where the attacker overwhelms the victim with ICMP "echo request" (ping) packets.

When the TCP flood occurs, there are large numbers of OWC requests, which could not complete three-way handshake. During UDP flood and ICMP flood, data request packets are captured only, and few data reply packets for the corresponding requests to the normal clients are found. Therefore, when DDoS attacks occur, the OWC request packets increase rapidly and OWCD increases obviously. In this paper, OWCD is defined as

$$\mathrm{OWCD} = \frac{\sum \mathrm{Packets}_{\mathrm{OWC}}}{\sum \mathrm{Packets}_{\mathrm{IP}}} \times 100\%. \tag{1}$$

$\mathrm{Packets}_{\mathrm{OWC}}$ and $\mathrm{Packets}_{\mathrm{IP}}$ stand for the number of OWC packets and IP packets separately.

3.2. Why Use Entropy of Traffic Packets Fields? Lakhina et al. [30] found that each kind of anomalies affects the distribution of certain traffic features. In one case, some feature distributions become more dispersed (e.g., source IP address in DDoS), while other feature distributions become concentrated (e.g., destination IP address in DDoS) on a small set of values. We need to find some statistic metrics to quantify the distribution of traffic features.

Generally, entropy refers to disorder or uncertainty, and the definition of entropy used in information theory is directly analogous to the definition used in statistical thermodynamics. The concept of information entropy was introduced by Shannon in his 1948 paper "A Mathematical Theory of Communication" [31]. Information entropy, which is used in this paper as entropy, could be used as such a metric to detect DDoS attacks effectively, which represents the random feature of network traffic. It describes the degree of concentration and dispersal characteristic of traffic.

In most normal cases, source IPs and destination IPs have nearly the same entropy values at the same time interval, which means the traffic flow between source hosts and destination hosts is simply point to point communications, while network anomalies will cause some changes in the distribution of packet fields in traffic. As DDoS attack happens, some feature distributions will become more dispersed, as when source addresses are spoofed, and other feature distributions will become more concentrated, as when destination addresses are overwhelmed. Entropy could capture this dispersal or concentration phenomenon.

Entropy is the measure of information and uncertainty of a random variable. The entropy of variable X can be defined as

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log_2 \left(P(x_i)\right). \tag{2}$$

X means a variable of one network traffic feature, which has n values x_i $(i = 1, \ldots, n)$, and $P(x_i)$ represents the probability of each value, satisfying $\sum_{i=1}^{n} P(x_i) = 1$. If we use the entropy as a detection feature, we can distinguish between normal and abnormal behavior by getting all entropy values of the traffic feature during a period of time.

In this paper, we propose a feature vector to detect DDoS attack. This vector is based on the characteristics of the one-way connection density and traffic distribution. This feature vector is (OWCD, E_{SIP}, E_{DIP}), in which E_{SIP} and E_{DIP} stand for source IP entropy and destination IP entropy separately. The contribution of each feature in the vector to the accuracy of the detection method is represented by the weight coefficient w_m $(i = 1, \ldots, 3)$, satisfying $\sum_{m=1}^{3} w_m = 1$.

4. Proposed Methods

4.1. Semisupervised k-Means Clustering. K-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. K-means algorithm is based on iterative relocation that partition a data set into k clusters, locally minimizing the average squared distance between the cluster data points and the cluster centers.

For a set of data points $X = \{x_1, \ldots, x_n\}$, the k-means algorithm creates a k-partitioning $\{X_i\}_{i=1}^{k}$ of X so that if c_i $(i = 1, \ldots, k)$ represent the ith partition center, then the following objective function is locally minimized:

$$O_{k\text{-means}} = \sum_{i=1}^{k} \sum_{x_j \in X_i} \left\| x_j - c_i \right\|^2. \tag{3}$$

K-means clustering has the following disadvantages: (1) the selection of k value is very difficult to estimate; (2) the initial clustering centers of the algorithm are randomly selected, and the selection of the center has a great influence on the clustering results; (3) the algorithm needs to continuously adjust the sample classification until convergence of the objective function, so that when the amount of data is very large, the time complexity of the algorithm is large.

Generally, in the research area of intrusion detection, there are four main categories of attacks. They are DoS (denial-of-service), R2L (unauthorized access from a remote machine, e.g., guessing password), U2R (unauthorized access to local root privileges, e.g., various "buffer overflow" attacks), and probing (surveillance and other probing, e.g., port scanning). There are two general forms of DoS attacks: those that crash services and those that flood services. The most serious attacks are distributed DoS (DDoS), and this paper focuses on DDoS attack. In order to detect DDoS attack from normal traffic, we use a semisupervised k-means clustering method, called constrained-k-means algorithm, which is evolved from the k-means algorithm. K-means algorithm has a disadvantage; that is, the selection of k value is very difficult to estimate, in which k stands for the clustering number. The paper only needs to distinguish two traffics, normal and DDoS attack. Besides, we use two categories of data sets to train our model, which are Inside Sniffer, Phase 5, of Lincoln Laboratory Scenarios (DDoS) 1.0 as attack data set and Week 1 of 1999 training data (attack free) as normal data set. So, the clustering number k is set to two, which will be confirmed correct by the experimental results in Section 5. Therefore, the first disadvantage of k-means algorithm described above can be ignored.

In order to solve the last two disadvantages of K-means algorithm, we use semisupervised clustering algorithm, which uses a small amount of labeled data to constrain the selection of the initial center points and improve the convergence speed and classification accuracy of algorithm.

Basu et al. [32] introduce two semisupervised clustering algorithms based on k-means, which are seeded-k-means and constrained-k-means. Both algorithms use the labeled seed set to initialize the center of k clusters. Seeded-k-means algorithm seeds the k-means algorithm with the labeled data set and labeling of the seed data may be changed in the following clustering step, while constrained-k-means algorithm does not change the labeling of the seed data in the whole algorithm. Therefore, if the initial labeling of seed set is noise-free, constrained-k-means algorithm is appropriate; otherwise, seeded-k-means algorithm is a better choice.

In this paper, we use Lincoln Laboratory Scenarios (DDoS) 1.0 as labeled data set and DARPAR Intrusion Detection Evaluation Data Set [33] as unlabeled training data set and test data set. Then, we use labeled data set as the initial seed (or labeled) data to determine the initial clustering centers and use unlabeled data set to implement the subsequent clustering procedure. We consider the labeled data set as noise-free, so constrained-k-means algorithm is chosen.

In the following, we combine constrained-k-means and features vector to provide a semisupervised clustering detection algorithm, which is named as Multiple-Features-Based Constrained-K-Means (MF-CKM) algorithm. So, the object function (3) is changed as follows:

$$O_{\text{MF-CKM}} = \sum_{i=1}^{k} \sum_{x_{mj} \in X_i} \left(\sum_{m=1}^{3} w_m \left\| x_{mj} - c_{mi} \right\| \right)^2. \quad (4)$$

Values of the selected detection features in Section 3 may have different scales. Before the above weighted object function is computed, these features need to be standardized as follows:

$$x_{mj} = \frac{x_{mj} - \min_{x_{mj} \in X_i} x_{mj}}{\max_{x_{mj} \in X_i} x_{mj} - \min_{x_{mj} \in X_i} x_{mj}}, \quad (5)$$

in which $\max_{x_{mj} \in X_i} x_{mj}$ and $\min_{x_{mj} \in X_i} x_{mj}$, respectively, stand for the maximum and minimum value of the mth feature. As a result, new value of feature x_{mj} is in range $[0, 1]$.

4.2. MF-CKM Detection Method. In K-means clustering method, the initial clustering centers of the algorithm are randomly selected, and the selection of the center has a great influence on the clustering results. In this paper, we use the small amount of labeled data to guide the selection of initial clustering centers and use other unlabeled data to train and form clusters.

The detection system based on MF-CKM algorithm is divided into three parts: feature extraction, model training, and model testing.

In the feature extraction phase, traffic features such as source IP and destination IP are extracted from the original data sets or real-time traffic. Then, the values of OWCD and the above feature entropies are computed during a period of time to form the feature vector set (OWCD, E_{SIP}, E_{DIP}).

In the model training phase, the MF-CKM algorithm is provided for clustering. For a 3-dimensional feature vector data set, the initial clustering center is calculated according to the labeled data and formula (4) is used to calculate the similarity between other unlabeled data sets and the initial clustering center until the algorithm converges (MF-CKM objective function is locally optimized). The clusters in the training process are built.

In the detection phase, by capturing new data packets and extracting features, the distances between the feature values of new data and each clustering center are calculated. Then, the new data is assigned to the closest cluster, which is based on the distances.

Multiple-Features-Based Constrained-K-Means (MF-CKM) algorithm is described as Algorithm 1.

5. Experimental Results and Analysis

5.1. Data Set and Data Preprocessing. In this paper, we use Inside Sniffer, Phase 5, of Lincoln Laboratory Scenarios (DDoS) 1.0 as labeled data set, Tcpdump data of Four-Hour Subset of Training Data (DARPA Intrusion Detection Evaluation Data Set) as unlabeled data set, and Monday's Tcpdump data (First Week of Test Data) as test data. This DDoS attack

Input: Set of data points $X = \{x_1, \ldots, x_n\}$, in which x_j $(j = 1, \ldots, n)$ is a vector with three detection features (the weight of each feature x_{mj} is expressed as w_m $(m = 1, \ldots, 3)$), number of clusters k, and set $S = \bigcup_{i=1}^{k} S_i$ of labeled data as seeds for selection of initial clustering center $c_i^{(0)}$, which satisfies $S \subset X$.

Output: Disjoint k partitioning $\{X_i\}_{i=1}^{k}$ of **X** such that the MF-CKM objective function is optimized.

Method:

(1) Selection of initialize clustering centers. $c_i^{(t)} = (1/|S_i|) \sum_{x_j \in S_i} x_j$, (for $i = 1, \ldots, k$; $t = 0$), in which $c_i^{(t)}$ is a vector with three elements expressed as $c_{mi}^{(t)}$ $(m = 1, 2, 3)$.

(2) Repeat until algorithm convergence

(2a) assign_cluster: For $x_j \in S$, if $x_j \in S_i$ assign x_j to the cluster i (i.e., set $X_i^{(t+1)}$). For $x_j \notin S$, assign x_j to the cluster i^* (i.e., set $X_{i^*}^{(t+1)}$), for $i^* = \arg\min_i (\sum_{m=1}^{3} \|x_{mj} - c_{mi}^{(t)}\| \cdot w_m)^2$

(2b) estimate_means: $c_i^{(t+1)} = (1/|X_i^{(t+1)}|) \sum_{x_j \in X_i^{(t+1)}} x_j$

(2c) $t = t + 1$

Algorithm 1: Multiple-Features-Based Constrained-K-Means (MF-CKM) detection algorithm.

Table 1: Comparison of clustering centers obtained by different methods using different features.

Feature	Method					
	Unsupervised K-means center		MF-CKM initial center		MF-CKM final center	
	Attack	Normal	Attack	Normal	Attack	Normal
Clustering center of E_{SIP}	6.468	1.687	6.610	1.871	6.530	1.754
Clustering center of E_{DIP}	0.029	1.667	0.008	1.874	0.019	1.748
Clustering center of OWCD	0.988	0.130	0.999	0.055	0.999	0.103

scenario is carried out over multiple network and audit sessions. These sessions have been grouped into 5 attack phases over the course of which the adversary probes, finds system vulnerabilities, breaks-in, installs trojan mstream DDoS software, and launches a DDoS attack against an off-site server. In this experiment, we need to detect the DDoS attacking process, so we only use the data of Phase 5 as labeled data sets.

The above data sets are stored as binary file, and, in order to facilitate data processing, we use Wireshark tool to convert the binary file to the XML file, which is shown as Box 1.

Besides, we use stream processing to read the data set XML files. The data processing is based on the principle of sliding window, and every 100 records of data sets are read in each time window, so that these training and testing data sets are transformed into flow data sets. In each time window, we could separately get the entropy of source IP and destination IP using formula (2). We can also calculate the OWCD by counting the number of OWC packets and IP packets using formula (1). In this paper, the proposed detection method is implemented on Eclipse 4.5.0 software by using Java language, WEKA, and MATLAB.

5.2. Results and Performance Analysis

5.2.1. Clustering Centers Using Different Detection Methods. As we know, the initial clustering centers of k-means algorithm are randomly selected, and MF-CKM algorithm uses labeled data to guide the selection of initial centers so that it could obtain relatively accurate results. In both algorithms, k is set to 2. Table 1 shows the comparison of clustering centers obtained by different methods using different features.

5.2.2. Clustering Results Using Different Detecting Methods with Single Feature. In this section, we firstly use single detection feature (such as source IP entropy, destination IP entropy, and OWCD) to compare the clustering results, respectively, achieved by the k-means algorithm and the MF-CKM algorithm.

Figure 1 uses source IP entropy as the detection feature to distinguish DDoS attack traffic and normal traffic as two clusters. In addition, this figure shows the different clustering centers of E_{SIP}, respectively, generated by unsupervised k-means algorithm and semisupervised MF-CKM algorithm.

Figure 2 uses destination IP entropy as the detection feature to distinguish abnormal traffic and normal traffic as two clusters. Moreover, this figure shows the different clustering centers of E_{DIP}, respectively, generated by k-means algorithm and the provided semisupervised algorithm (MF-CKM).

Figure 3 uses OWCD as the detection feature to distinguish DDoS attack traffic and normal traffic as two clusters. Furthermore, this figure shows the different clustering centers of OWCD respectively generated by unsupervised k-means algorithm and the proposed MF-CKM algorithm.

```
<packet>
<section>20</section>                                   //Packet ID
<section>202.77.162.213</section>                       //Source IP
<section>49212</section>                                //Source Port
<section>172.16.115.20</section>                        //Destination IP
<section>23</section>                                   //Destination Port
<section>49212 &gt; telnet [PSH, ACK] Seq=124 Ack=89 Win=33580 Len=1 TSV=190937
TSER=9429952</section>                                  //Payload
</packet>
```

Box 1: XML file structure.

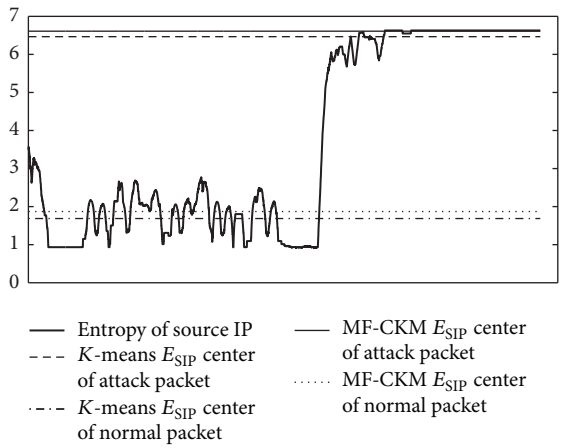

Figure 1: Comparison of source IP entropy center obtained by K-means and MF-CKM algorithm.

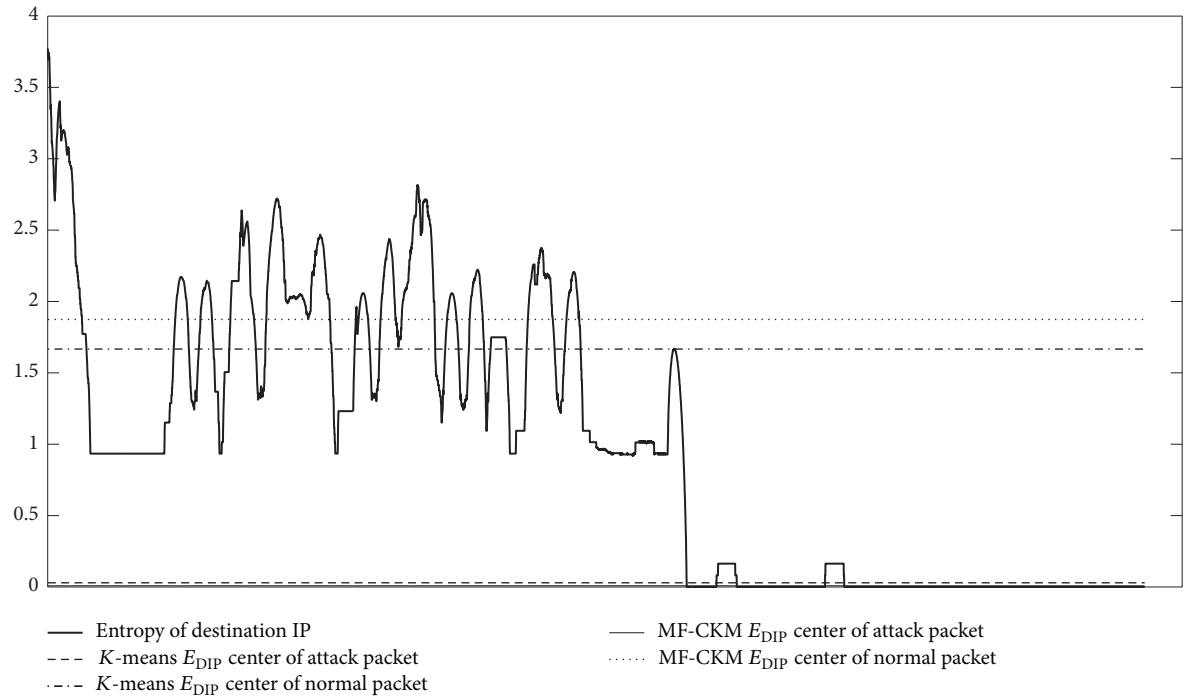

Figure 2: Comparison of destination IP entropy center obtained by K-means and MF-CKM algorithm.

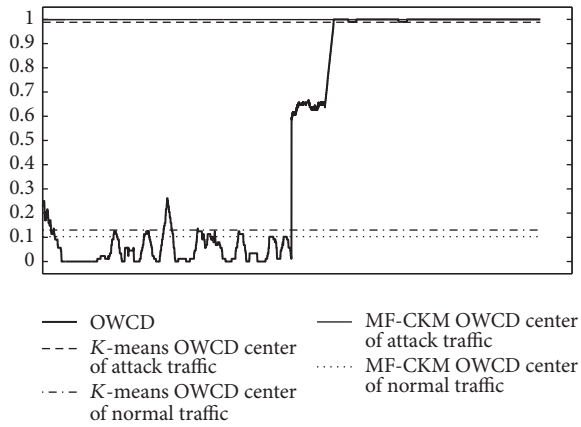

Figure 3: Comparison of OWCD center obtained by K-means and MF-CKM algorithm.

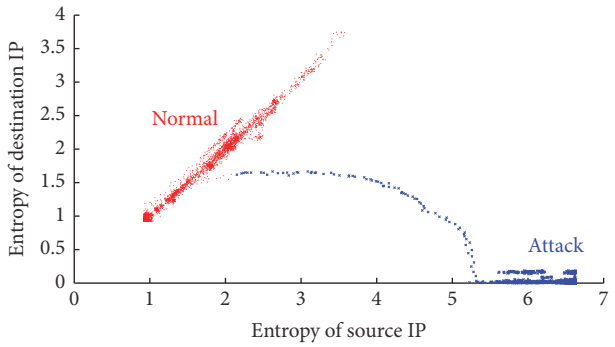

Figure 4: Clustering results using two-dimensional feature vector (E_{SIP}, E_{DIP}).

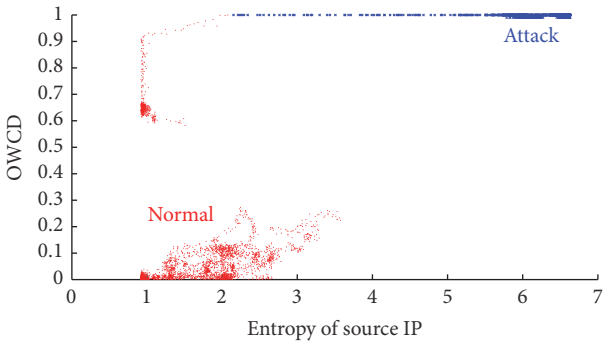

Figure 5: Clustering results using two-dimensional feature vector (E_{SIP}, OWCD).

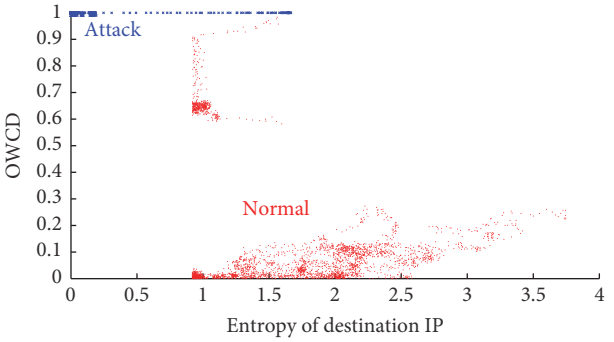

Figure 6: Clustering results using two-dimensional feature vector (E_{DIP}, OWCD).

The three figures show the gradual change of the three detection feature values during the process of network traffic. As we can see, these values are gradually moving away from the normal range until they are stabilized around another center, which is the center of DDoS attack.

5.2.3. Clustering Results Using MF-CKM Algorithm with Two-Dimensional Feature Vector. In this experiment, Weka software is used and two-dimensional feature vectors of source IP entropy and destination IP entropy are used as the equally weighted detection features to form a two-dimensional clustering map as Figure 4. In Figure 4, the cluster shown as blue points stands for the cluster of attack traffic, and the cluster shown as red points represents the cluster of normal traffic. As we can see from the figure, MF-CKM algorithm divides the data set into two distinct clusters. The intermediate points between two clusters are due to the gradual progress of network traffic from normal traffic to the beginning of the attack flow and then to the maximum flow of attack traffic.

And then, two-dimensional feature vectors of source IP entropy and OWCD are used as the equally weighted detection features to form a two-dimensional clustering map as Figure 5. In Figure 6, two-dimensional feature vectors of destination IP entropy and OWCD are used as the equally weighted detection features to form a two-dimensional clustering map. In these two figures, the cluster shown as blue points stands for the cluster of attack traffic, and the cluster shown as red points represents the cluster of normal traffic. As we can see from these two figures, MF-CKM algorithm also divides the data set into two distinct clusters. Obviously, there are some transition points between two clusters, and these intermediate points also show the gradual progress of network traffic from normal traffic to the beginning of the attack flow and then to the maximum flow of attack traffic.

5.2.4. Comparison of Convergence Time between K-Means Algorithm and MF-CKM Algorithm. The convergence condition of the two clustering algorithm is that the difference of clustering centers obtained by the continuous two cycles in each algorithm is less than a given threshold value, which is set to zero in this experiment.

Because execution time of the algorithm is not a fixed value, the convergence time of the algorithm is calculated as the average value of the time running for 10 times. The average convergence time of the two algorithms using different detection features is shown in Table 2. It can be seen

Table 2: Comparison of average time of two algorithms using different detection features.

Detection feature	Algorithm	
	K-means	MF-CKM
E_{SIP}	16.1 ms	14.5 ms
E_{DIP}	15.5 ms	15.4 ms
OWCD	14.6 ms	14.0 ms

Remarks. E_{SIP}: entropy of source IP; E_{DIP}: entropy of destination IP.

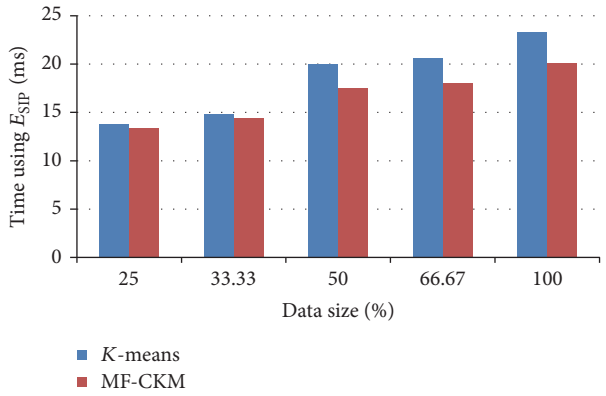

Figure 7: Running time histogram of K-means and MF-CKM using E_{SIP}.

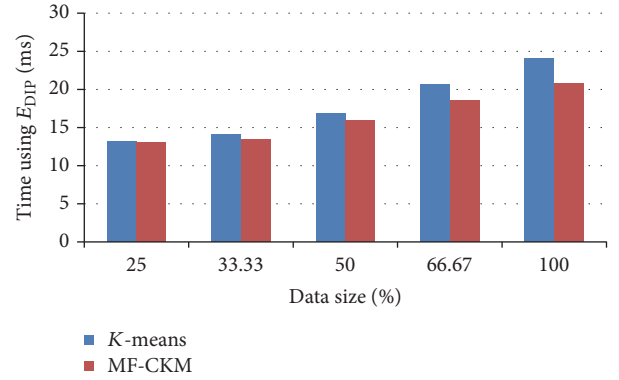

Figure 8: Running time histogram of K-means and MF-CKM using E_{DIP}.

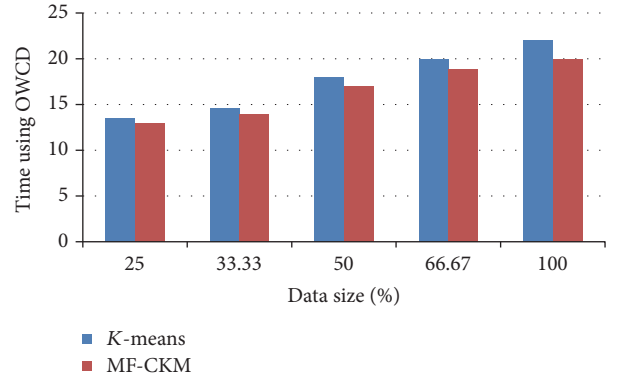

Figure 9: Running time histogram of K-means and MF-CKM using OWCD.

from Table 2 that the MF-CKM algorithm consumes less time compared with the K-means algorithm.

Besides, in order to verify the time-consuming difference between the two algorithms in different size of data sets, this paper increases the amount of data sets and adds *1998 Training Data, Week 1 of DARPA Intrusion Detection Evaluation Data Set* into the two previous used data sets (*Inside Sniffer, Phase 5, of Lincoln Laboratory Scenarios 1.0* and *Tcpdump data of Four-Hour Subset of Training Data*) to train the model. After the preprocessing used in Section 5.1, the aforementioned three data sets have about 30000 records. We extract 25%, 33.33%, 50%, 66.66%, and 100% records from the total records and calculate the running time of the two algorithms, respectively, and the results were compared as follows in Table 3. The running time histograms of the two algorithms using three different features are shown as follows from Figures 7–9. As the amount of data sets increases, the time-consuming difference between the running times of two algorithms is also increasing.

6. Conclusions and Future Work

In order to integrate the advantages of supervised and unsupervised learning methods and considering the actual application scenes which have small amount of labeled data and large amount of unlabeled data, this paper provides a semisupervised clustering method MF-CKM algorithm to detect DDoS attacks. The provided algorithm uses the feature vector as the feature detection to reduce the problem of low detection efficiency caused by using single feature. In the same time, MF-CKM uses the labeled data to guide the selection of the initial clustering center to improve the convergence speed.

In the subsequent experiments, a larger data set will be used to verify the advantages of the provided algorithm in terms of convergence time. At the same time, we will consider using DDoS attack tools in our lab's local area network to generate real-time attack traffic to further verify the proposed algorithm.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (no. 61173017, no. 61370195), Communication Soft Science Foundation of Ministry of Industry and Information (no. 2014-R-42, no. 2015-R-29), Key Lab of Information Network Security Foundation of Ministry

Table 3: Comparison of two algorithms running times using different detection features with different data size.

Feature	Data size									
	25%		33.33%		50%		66.67%		100%	
	Algorithm									
	K-means	MF-CKM	K-means	MF-CKM	K-means	MF-CKM	K-means	MF-CKM	K-means	MF-CKM
E_{SIP}	13.8 ms	13.4 ms	14.8 ms	14.4 ms	20.0 ms	17.5 ms	20.6 ms	18.0 ms	23.3 ms	20.1 ms
E_{DIP}	13.2 ms	13.0 ms	14.1 ms	13.5 ms	16.9 ms	16.0 ms	20.6 ms	18.5 ms	24.1 ms	20.8 ms
OWCD	13.5 ms	13.0 ms	14.6 ms	14.0 ms	18.0 ms	17.0 ms	20.0 ms	18.8 ms	22.0 ms	20.0 ms

Remarks. E_{SIP}: entropy of source IP; E_{DIP}: entropy of destination IP.

of Public Security (no. C14613), and State Grid Technology Project (no. SGTYHT/15-JS-191).

References

[1] O. Osanaiye, K.-K. R. Choo, and M. Dlodlo, "Distributed denial of service (DDoS) resilience in cloud: review and conceptual cloud DDoS mitigation framework," *Journal of Network and Computer Applications*, vol. 67, pp. 147–165, 2016.

[2] G. Somani, M. S. Gaur, D. Sanghi, M. Conti, M. Rajarajan, and R. Buyya, "Combating DDoS attacks in the cloud: requirements, trends, and future directions," *IEEE Cloud Computing*, vol. 4, no. 1, pp. 22–32, 2017.

[3] N. Hoque, D. K. Bhattacharyya, and J. K. Kalita, "Botnet in DDoS attacks: trends and challenges," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2242–2270, 2015.

[4] A. Shameli-Sendi, M. Pourzandi, M. Fekih-Ahmed, and M. Cheriet, "Taxonomy of distributed denial of service mitigation approaches for cloud computing," *Journal of Network and Computer Applications*, vol. 58, pp. 165–179, 2015.

[5] K. Saravanan, R. Asokan, and K. Venkatachalam, "Detection mechanism for distributed denial of service (DDoS) attacks for anomaly detection system," *Journal of Theoretical and Applied Information Technology*, vol. 60, no. 1, pp. 174–178, 2014.

[6] A. Kulkarni, Y. Pino, M. French, and T. Mohsenin, "Real-time anomaly detection framework for many-core router through machine-learning techniques," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 13, no. 1, article no. 10, pp. 1–22, 2016.

[7] J. Song, Z. Zhu, P. Scully, and C. Price, "Selecting features for anomaly intrusion detection: a novel method using fuzzy C means and decision tree classification," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 8300, pp. 299–307, 2013.

[8] D. M. Farid and M. Z. Rahman, "Anomaly network intrusion detection based on improved self adaptive Bayesian algorithm," *Journal of Computers*, vol. 5, no. 1, pp. 23–31, 2010.

[9] L. Koc, T. A. Mazzuchi, and S. Sarkani, "A network intrusion detection system based on a hidden Naïve Bayes multiclass classifier," *Expert Systems with Applications*, vol. 39, no. 18, pp. 13492–13500, 2012.

[10] C. A. Catania, F. Bromberg, and C. G. Garino, "An autonomous labeling approach to support vector machines algorithms for network traffic anomaly detection," *Expert Systems with Applications*, vol. 39, no. 2, pp. 1822–1829, 2012.

[11] A. Saied, R. E. Overill, and T. Radzik, "Artificial neural networks in the detection of known and unknown DDoS attacks: proof-of-concept," *Communications in Computer and Information Science*, vol. 430, pp. 300–320, 2014.

[12] E. R. Ramos, S. Chae, M. Kim, and M. Choi, "The optimistic schemes of cluster analysis and k-NN classifier method in detecting and counteracting learned DDoS attack," in *Proceedings of the New Technologies, Mobility and Security Conference and Workshops, NTMS 2008*, Morocco, November 2008.

[13] G. Münz, S. Li, and G. Carle, "Traffic anomaly detection using k-means clustering," In GI/ITG Workshop MMBnet, 2007.

[14] Y. Liao and V. R. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection," *Computers & Security*, vol. 21, no. 5, pp. 439–448, 2002.

[15] S. Seufert and D. O'brien, "Machine learning for automatic defence against distributed denial of service attacks," in *Proceedings of the 2007 IEEE International Conference on Communications, ICC'07*, pp. 1217–1222, UK, June 2007.

[16] T. Subbulakshmi, S. M. Shalinie, V. Ganapathisubramanian, K. Balakrishnan, D. Anandkumar, and K. Kannathal, "Detection of DDoS attacks using enhanced support vector machines with real time generated dataset," in *Proceedings of the 3rd International Conference on Advanced Computing (ICoAC '11)*, pp. 17–22, IEEE, Chennai, India, December 2011.

[17] P. Xiao, W. Y. Qu, H. Qi, and Z. Y. Li, "Detecting DDoS attacks against data center with correlation analysis," *Computer Communications*, vol. 67, pp. 66–74, 2015.

[18] G. Öke, G. Loukas, and E. Gelenbe, "Detecting denial of service attacks with Bayesian classifiers and the random neural network," in *Proceedings of the 2007 IEEE International Conference on Fuzzy Systems, FUZZY*, UK, July 2007.

[19] N. B. Amor, S. Benferhat, and Z. Elouedi, "Naive Bayes vs decision trees in intrusion detection systems," in *Proceedings of the ACM Symposium on Applied Computing*, pp. 420–424, ACM, March 2004.

[20] L. Kaufman and P. J. Rousseeuw, *An introduction to cluster analysis*, Wiley series in probability and mathematical statistics, John Wiley and Sons, Inc, 1990.

[21] J. Yu, Z. Li, H. Chen, and X. Chen, "A detection and offense mechanism to defend against application layer DDoS attacks," in *Proceedings of the 3rd International Conference on Networking and Services, ICNS 2007*, Greece, June 2007.

[22] M. Suresh and R. Anitha, "Evaluating machine learning algorithms for detecting DDoS attacks," *Communications in Computer and Information Science*, vol. 196, pp. 441–452, 2011.

[23] K. L. Li, W. Zhang, X. Ma, Z. Cao, and C. Zhang, "A novel semi-supervised SVM based on tri-training," in *Proceedings of the 2008 2nd International Symposium on Intelligent Information*

Technology Application, IITA 2008, pp. 47–51, China, December 2008.

[24] Y. Li, Z. Li, and R. Wang, "Intrusion detection algorithm based on semi-supervised learning," in *Proceedings of the 2011 International Conference on Information Technology, Computer Engineering and Management Sciences, ICM 2011*, pp. 153–156, China, September 2011.

[25] P. Laskov, P. Düssel, C. Schäfer, and K. Rieck, "Learning intrusion detection: supervised or unsupervised?" in *Image Analysis and Processing—ICIAP 2005: 13th International Conference, Cagliari, Italy, September 6–8, 2005. Proceedings*, vol. 3617 of *Lecture Notes in Computer Science*, pp. 50–57, Springer, Berlin, Germany, 2005.

[26] M. Luo, L. Wang, H. Zhang, and J. Chen, "A Research on intrusion detection based on unsupervised clustering and support vector machine," in *Proceedings of Information and Communications Security: 5th International Conference, ICICS*, vol. 2836, pp. 325–336, Huhehaote , China, 2003.

[27] U. X. Tu and H. E. Da-Ke, *Features analysis and detection of DDoS attack*, Computer Engineering Applications, 2006.

[28] "TCP SYN flooding and IP spoofing attacks," *Network Security*, vol. 1996, no. 10, p. 2, 1996.

[29] "UDP port denial-of-service attack," *Network Security*, vol. 1996, no. 2, pp. 2-3, 1996.

[30] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *Proceedings of the the 2005 conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp. 217–228, Philadelphia, Pa, USA, August 2005.

[31] C. E. Shannon, "A mathematical theory of communication," *Bell Labs Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.

[32] S. Basu, A. Banerjee, and R. J. Mooney, "Semi-supervised clustering by seeding," in *Proceedings of the Nineteenth International Conference on Machine Learning*, pp. 27–34, Morgan Kaufmann, 2002.

[33] Lincoln Laboratory Scenario (DDoS) 1.0 of DARPA Intrusion Detection Evaluation Data Set, http://www.ll.mit.edu/ideval/data/2000/LLS_DDOS_1.0.html, 2000.