IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 12, 2019, accepted May 10, 2019, date of publication May 17, 2019, date of current version May 30, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2917532

Semi-Supervised K-Means DDoS Detection Method Using Hybrid Feature Selection Algorithm

YONGHAO GU, KAIYUE LI, ZHENYANG GUO, AND YONGFEI WANG
Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia, School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: Yonghao Gu (guyonghao@bupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant U1536121, Grant 61370195, and Grant 61873040, in part by the China Computer Federation-NSFOCUS (CCF-NSFOCUS) KunPeng Research Fund under Grant 2018004, in part by the Fundamental Research Funds for the Central Universities, in part by the Communication Soft Science Foundation of Ministry of Industry and Information under Grant 2015-R-29, in part by the State Grid Technology Project under Grant SGTYHT/15-JS-191, and in part by the Key Lab of Information Network Security Foundation of Ministry of Public Security.

ABSTRACT Distributed denial of service (DDoS) attack is an attempt to make an online service unavailable by overwhelming it with traffic from multiple sources. Therefore, it is necessary to propose an effective method to detect DDoS attack from massive data traffics. However, the existing schemes have some limitations, including that supervised learning methods, need large numbers of labeled data and unsupervised learning algorithms have relatively low detection rate and high false positive rate. In order to tackle these issues, this paper presents a semi-supervised weighted k-means detection method. Specifically, we firstly present a Hadoop-based hybrid feature selection algorithm to find the most effective feature sets and propose an improved density-based initial cluster centers selection algorithm to solve the problem of outliers and local optimal. Then, we provide the Semi-supervised K-means algorithm using hybrid feature selection (SKM-HFS) to detect attacks. Finally, we exploit DARPA DDoS dataset, CAIDA "DDoS attack 2007" dataset, CICIDS "DDoS attack 2017" dataset and real-world dataset to carry out the verification experiment. The experiment results have demonstrated that the proposed method outperforms the benchmark in the respect of detection performance and technique for order preference by similarity to an ideal solution (TOPSIS) evaluation factor.

INDEX TERMS DDoS attack, semi-supervised k-means, Hadoop-based hybrid feature selection, ratio of average sum of squared errors (SSE) to cluster distance (RSD), TOPSIS.

I. INTRODUCTION

In computing, a denial-of-service attack (DoS attack) is a cyber-attack in which 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 Internet. DoS 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. In a distributed denial-of-service attack (DDoS attack), incoming traffic flooding the victim originates from many different sources. This effectively makes it impossible to stop attack simply by blocking a single source. A DDoS attack is analogous to a group of people crowding the entry door of a shop, making it hard for ordinary customers to enter, disrupting trade. Nowadays DDoS attacks are rising significantly and high-volume attack events have occurred quite frequently, which have a lot to do with the prevalence of Internet of Things (IoT) botnets, such as Mirai botnet.

A. MOTIVATION

To prevent against DDoS attacks, researchers have proposed and implemented various countermeasures, including detection, defense and traceback. Among all these countermeasures, DDoS detection is the first and most important step in fighting against DDoS attacks. There are two classes of DDoS detection techniques: misuse detection and anomaly detection. Misuse detection techniques try to detect attack by comparing the current activity of destination network to a database of known attack signatures. But, these techniques

The associate editor coordinating the review of this manuscript and approving it for publication was Zesong Fei.

are difficult to detect new attacks. So, anomaly detection techniques are introduced to detect unknown attacks by comparing the current activity of destination network to an established normal activity represented as a profile. The basic detection approach is to use machine learning to create a model of trustworthy activity, and then compare new behavior against this model. Machine learning methods mainly include supervised learning and unsupervised learning. Machine learning-based detection methods have the following limitations. Insufficient labeled data of supervised learning methods lead to low detection rate, and unreasonable initialization of unsupervised learning parameters leads to local optimal or poor detection effect. In addition, too many features in learning process cause ''the curse of dimensionality'', and unreasonable feature sets lead to poor detection performance.

In order to overcome the above limitations, this paper proposes a semi-supervised clustering detection method using hybrid feature selection algorithm, and the provided method uses only small amount of labeled data and relatively large amount of unlabeled data to detect DDoS attack behavior.

B. CONTRIBUTIONS

The main contributions of this paper can be summarized as follows:

1) It presents a hadoop-based hybrid feature selection method combined with subsequent learning process to find the most effective feature set. (Section III)
2) It proposes a semi-supervised weighted k-means method using hybrid feature selection algorithm (SKM-HFS) to achieve better detection performance, and this method requires fewer labeled data sets for training. (Section IV)
3) It provides an improved density-based initial cluster centers selection method to solve the problem of outliers and local optimal of k-means clustering. (Section IV)
4) It exploits DARPA DDoS dataset, CAIDA ''DDoS attack 2007'' dataset, CICIDS ''DDoS attack 2017'' dataset and real-world dataset to verify that the proposed method outperforms the benchmark in the respect of detection performance and TOPSIS evaluation factor. (Section V)

The remainder of this paper is organized as follows. In Section II, we review the related works about DDoS detection methods and feature selection methods in DDoS detection. Section III provides the hybrid feature selection method and Section IV proposes the semi-supervised k-means DDoS detection algorithm using hybrid feature selection method, and Section V shows the experiment details and gives the experiment results and analyses. Finally, conclusions and future work are provided in Section VI.

II. RELATED WORKS

A. DETECTION METHODS

There are many DDoS attack detection methods, while this paper mainly focuses on machine learning based detection methods. Machine learning methods mainly include unsupervised learning and supervised learning.

Unsupervised learning techniques deal with learning tasks with unlabeled or untagged data, and clustering is the most popular unsupervised learning technique. K-means algorithm, as a clustering method, has been successfully used to detect anomalies [1] and DDoS [2], and some modified k-means methods [3], [4] are provided to improve detection efficiency. Besides, there are some other unsupervised learning methods to detect DDoS attacks [5].

Meanwhile, many supervised learning algorithms are used for DDoS detection [6]. Nguyen *et al.* [7] uses k-NN classifier method and cosine formula based algorithm to detect DDoS attacks. Xiao*et al.* [8] presents a CKNN (KNN with Correlation analysis) detection method, which exploits correlation information of training data to improve classification accuracy. Vijayasarathy *et al.* [9] uses multiple Bayesian classifiers to detect DDoS attacks. However, naive Bayes is based on a very strong independence assumption, which is not always satisfied. Bouzida *et al.* [10] proposes a decision tree-based detection method and exploits KDD99 dataset for model training and testing to obtain 93% detection rate. But, this model requires a large number of labeled data for effective training. Li *et al.* [11] demonstrates a DDoS detection system based on LVQ neural network to improve accuracy. Cheng *et al.* [12] proposes a flow correlation degree feature and applies a random forest detection model, which has a 98.57% detection rate and a 2.72% false positive rate. With increase of labeled samples, detect accuracy is improved. Khundrakpam *et al.* [13] detects DDoS attack using a multilayer perceptron (MLP) classification method with genetic algorithm.

In summary, supervised learning method has better accuracy. But, one limitation of it is the need of large-scale labeled data to train the classifier, which is not easy to get. Unsupervised learning has the advantage of detecting new samples better than supervised learning. However, the manually assignment of cluster numbers and other parameters could result in relatively low accuracy.

In addition to the above machine learning methods, chaos theory is used in DDoS detection in recent years [14]–[16] and has a nice detection performance. This paper mainly deals with the limitations of supervised learning and unsupervised learning detection methods, and also compares the detection performance between the proposed method and the chaos theory based method.

B. FEATURE SELECTION METHODS

The research on machine learning based DDoS detection method not only focuses on detection models, but also includes the feature selection methods. Related papers of feature selection methods in DDoS detection are shown as follows.

Yusof *et al.* [17] combines the consistency subset evaluation (CSE) and DDoS characteristic features (DCF) for feature selection, which is superior to traditional features selection method such as Information Gain, Gain Ratio, Correlated features selection(CFS), but this paper does not tackle feature redundancy. Balkanli *et al.* [18] employs the Chi-Square and Symmetrical Uncertainty with Decision Tree classifier to detect backscatter DDoS behaviors, which needs large number of labeled features. Zi *et al.* [19] provides the linear correlation coefficient for feature ranking and Modified Global K-means algorithm(MGKM) to detect attacks. As the number of top-ranked features decreases, a point where the cluster function value drops heavily will be chosen and the final selected features will be identified. However, this method could not capture correlations between features that are not liner in nature. Jiang *et al.* [20] proposes the filter algorithm GAIG for feature selection by combing genetic algorithm as the search strategy and information gain as the evaluation function, which could reduce the noisy features. However, the provided genetic search strategy has a high time complexity. Osanaiye *et al.* [21] presents an Ensemble-based Multi-Filter Feature Selection (EMFFS) method combining Information Gain, Gain Ratio, Chi-square and ReliefF, which has a high computational complexity.

From the above analysis, it is necessary to provide a reasonable feature selection algorithm before model training to achieve effective attack detection.

III. HYBRID FEATURE SELECTION METHOD

As we know, there are many features used for DDoS detection. While, too many features in the learning process cause ‘‘the curse of dimensionality’’, and unreasonable feature sets lead to poor detection performance. This paper proposes a hybrid feature selection method for DDoS detection. The method includes three steps, namely data normalization, feature ranking and feature subset searching. The input of this method is the candidate feature set and the output is the selected feature set for detection model.

A. CANDIDATE FEATURE SET

Through investigation of related works, we find that detection features mainly include: entropy [24]–[27], conditional entropy [28], Renyi entropy [29], φ-entropy of source ip (destination ip, protocol) [30], occurrence rate of TCP packet (UDP packet, ICMP packet) [25], percent of packets with the port number 80, variance of the numbers of packets to each destination ip, average of payloads, probability of occurrence of IP [31], mean time intervals (*MTI*), TTL, time stamp, ACK value, SYN value [32], variation index of source IPs [33], answer resource record, authority resource record, average packet size [34] and etc. Among the above 38 features, the most widely used features are the following 13 ones: entropy of source ip ($H(Sip)$), entropy of destination ip ($H(Dip)$), entropy of source port ($H(Sport)$), entropy of destination port ($H(Dport)$), conditional entropy of source ip given destination ip ($H(Sip \mid Dip)$), conditional entropy of source ip given destination port ($H(Sip \mid Dport)$), conditional entropy of destination port given destination ip ($H(Dport \mid Dip)$), One-Way Connection Density (*OWCD*), entropy of packet type ($H(PacType)$), occurrence rate of TCP packet (*TCPRate*), occurrence rate of UDP packet (*UDPRate*) and occurrence rate of ICMP packet (*ICMPRate*), time interval of packets (*PckTimeInt*). Many other features are variants of the above 13 features, or derived from them. Therefore, it is practical to compare the detection effects between these 13 features as candidate features and select the final feature sets from them.

In order to verify the performance of each feature while reducing the number of candidate features, we compare the changes of each feature value before and under DDoS attacks. The DARPA DDoS dataset [35] is used in comparison experiments. The variation of each feature is shown in the following figures (Fig. 1 and Fig. 2). The first half (left of the dotted line) of each figure shows the feature value fluctuation without attack, and the second half (right of the dotted line) shows the feature value fluctuation under attack.

As we can see from these figures, the larger the relative difference of feature values between left part and right part is, the more obvious the distinction between normal traffic and attack traffic is, the more effective the feature is. Fig. 1 shows the values of 9 features with better performance, and Fig. 2 shows the other 4 feature values with relatively poor performance.

In addition, this paper focuses on feature selection method rather than extracted features. If other good features are found, they can be added to the corresponding candidate feature set to participate in the proposed feature selection process. Therefore, this paper chooses these 9 features in Tab. 1 as the candidate feature set.

TABLE 1. Main features used in DDoS detection methods.

No.	Feature	Feature Description
1	*H*(*Sip*)	Entropy of source ip
2	*H*(*Dip*)	Entropy of destination ip
3	*H*(*Sport*)	Entropy of source port
4	*H*(*Dport*)	Entropy of destination port
5	*H*(*PacType*)	Entropy of packet type
6	*OWCD*	One-Way Connection Density
7	*H*(*Sip*|*Dip*)	Conditional Entropy of source ip given destination ip
8	*H*(*Sip*|*Dport*)	Conditional Entropy of source ip given destination port
9	*H*(*Dport*|*Dip*)	Conditional Entropy of destination port given destination ip

Why use entropy of traffic packets field as detection feature in many papers? Lakhina *et al.* [36] 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

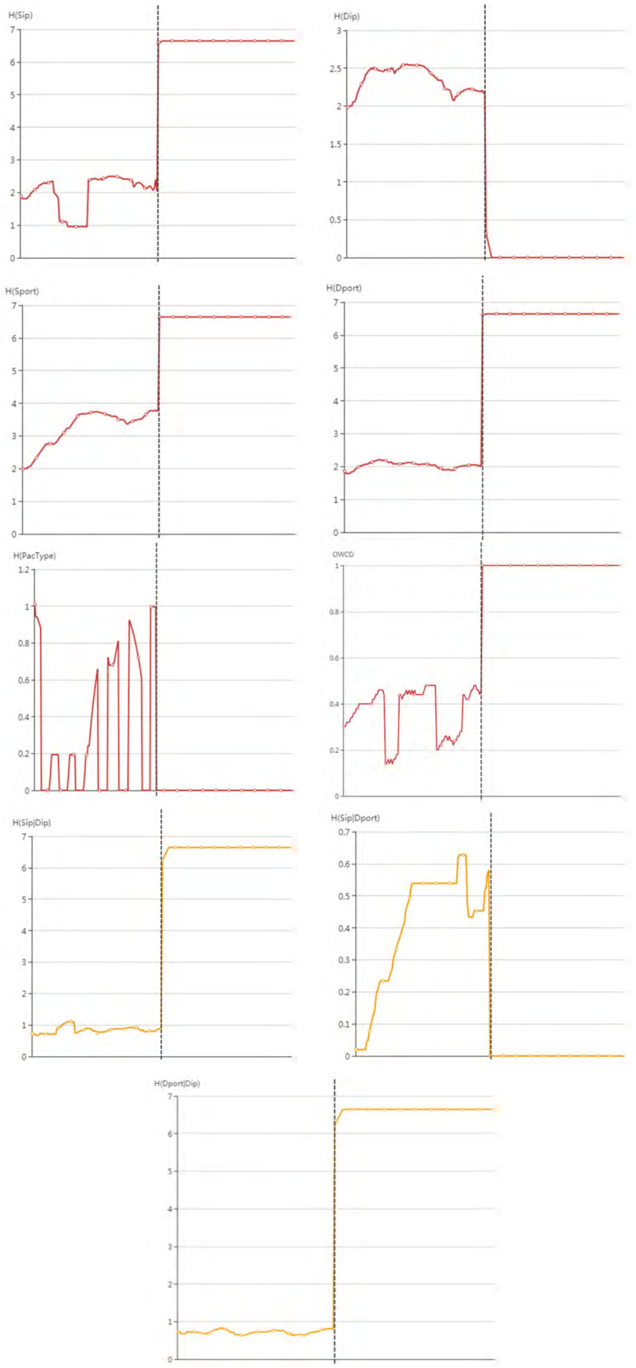


FIGURE 1. Nine features with better detection effect.

theory is directly analogous to the definition used in statistical thermodynamics. 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. Entropy is the measure of information and uncertainty of a random variable. The entropy of variable X can be defined as:

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

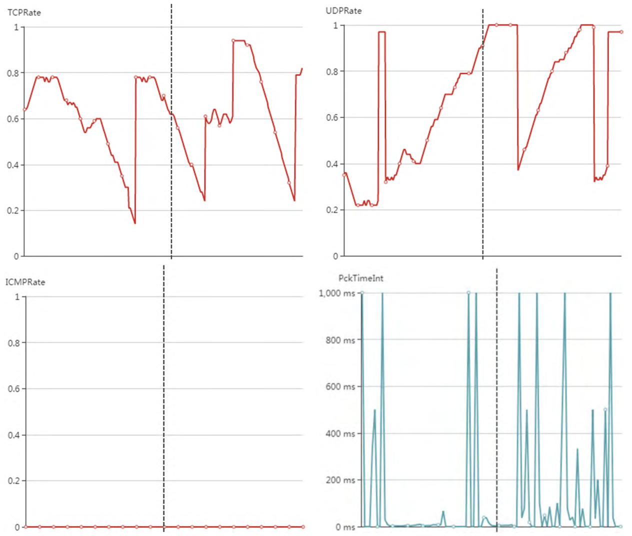


FIGURE 2. Four features with poor detection effect.

X means the 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 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. The calculation method of *OWCD* can be seen in our previous work [22].

B. DATA NORMALIZATION

Before making feature selection, data normalization is an essential process, which scales the value of each feature into a well-proportioned range, so that the bias in favor of features with greater values is eliminated from the dataset. The normalization transforms each feature value linearly scaled to the one in the range of [0], [1] using the equation (2):

$$x_{mj} = \frac{x_{mj} - \min\limits_{x_{mj}\in X_i} x_{mj}}{\max\limits_{x_{mj}\in X_i} x_{mj} - \min\limits_{x_{mj}\in X_i} x_{mj}}, \tag{2}$$

in which $\max\limits_{x_{mj}\in X_i} x_{mj}$ and $\min\limits_{x_{mj}\in X_i} x_{mj}$ respectively stands for the maximum and minimum value of the m^{th} feature. The process of data normalization is described as follows.

1) Collect the necessary metadata from the actual dataset, including data record number, time, protocol type, source IP address, destination IP address, source port number, and destination port number, which are stored in the *txt* or *csv* file.
2) The above file is processed by sliding window principle.
3) Use the metadata of each sliding window unit in the above file to calculate the feature values in Tab. 1 and form the value set for each feature.
4) The value of each feature is normalized by equation (2).

C. FEATURE RANKING

After feature normalization, all candidate features will be ranked by filter models. Traditional filter methods analyze features independent of the classifier and use "goodness" metric to decide which features should be kept. They sort each feature through specific evaluation index, and "poor" ranking index may not have ideal accuracy in attack detection. In this paper, the filter process is combined with subsequent learning algorithm, and a novel ranking index is proposed by using the objective function *SSE* (Sum of Squared Errors) of k-means method. The definition of the provided ranking index is as follows:

Definition 1: Ratio of average *SSE* to cluster Distance (*RSD*)

The datasets $X = \{x_1, \ldots, x_n\}$ are clustered by k-means algorithm based on each feature $f_t (t = 1, 2, ..., l)$, and the center of each cluster $C_i\ (i = 1, 2, \cdots, k)$ is expressed as c_i. So the ratio of average *SSE* to the sum of the distances between each pair of centers is obtained, which is shorted as $RSD\,(f_t)$

$$RSD\,(f_t) = \frac{SSE}{n \cdot \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \left\| c_i - c_j \right\|^2}, \tag{3}$$

and *SSE* is computed using the following equation (4)

$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} \|c_i - x\|^2. \tag{4}$$

Why we use*RSD*as the feature ranking index?

The evaluation criterion of good clustering model is having both high intra-cluster similarity and low inter-cluster similarity. Intuitively, the proposed ranking index should be proportional to the intra-cluster distance and inversely proportional to the inter-cluster distance.

For the intra-cluster similarity, *SSE* in k-means algorithm characterizes the clustering degree of intra-cluster data. The dataset X is clustered by k-means algorithm, and the center of each cluster C_i is expressed as c_i. The mean intra-cluster distance of X in the k clusters is

$$d_{\text{intra-cluster}} = \frac{1}{n} SSE = \frac{1}{n} \sum_{i=1}^{k} \sum_{x \in C_i} \|c_i - x\|^2.$$

The smaller the $d_{\text{intra-cluster}}$ is, the higher the intra-cluster similarity is.

For the inter-cluster similarity, the distance between any two clusters C_i and C_j is expressed by the distance between the centers of the two clusters as

$$d_{\text{inter-cluster}}(C_i, C_j) = \left\| c_i - c_j \right\|^2.$$

So, the inter-cluster similarity of k clusters is expressed by the sum of all the cluster-cluster distances as

$$d_{\text{inter-cluster}} = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} d_{\text{inter-cluster}}(C_i, C_j)$$

$$= \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \left\| c_i - c_j \right\|^2.$$

The larger the $d_{\text{inter-cluster}}$ is, the lower the inter-cluster similarity is.

Since the *RSD* is proportional to the intra-cluster distance and inversely proportional to the inter-cluster distance, the *RSD* of each feature f_t is computed as

$$RSD\,(f_t) = \frac{SSE}{n \cdot \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \left\| c_i - c_j \right\|^2}.$$

The smaller the $RSD\,(f_t)$ is, the better the clustering result is by using the feature f_t, which means the feature is much better for detecting attacks.

So, the candidate feature set is sorted by $RSD\,(f_t)$ and features whose *RSD* values are above a given threshold are discarded. And finally, the initial feature subset is obtained as $F^{'} = \{f_1, f_2, \cdots, f_{l'}\}$ from the candidate feature set $F = \{f_1, f_2, \cdots, f_l\}$. The process of feature ranking is described as follows.

1) The normalized candidate feature set is clustered by k-means method.
2) According to the clustering results, k clustering centers $\{c_i\ (i = 1, 2, \cdots, k)\}$ of each feature value and *SSE* are obtained.
3) Calculate the *RSD* values of each feature using equation (3).
4) The *RSD* values are arranged in ascending order, and the first l' features are selected as initial feature set or the corresponding feature set whose *RSD* values are less than a given threshold are selected as the initial feature set $F^{'} = \{f_1, f_2, \cdots, f_{l'}\}$.

D. FEATURE SUBSET SEARCHING

After getting the initial feature set $F^{'}$, we need a proper search strategy to find the best feature subset for detection model. In order to reduce the time complexity, we use the Sequential Forward Selection (SFS) algorithm. SFS is a search method that starts with an empty set of features and adds a single feature from $F^{'}$ in each iteration with the fitness function $f\,(x)$ monotonously increasing or decreasing. The SFS algorithm has following steps:

1) Initialize the empty feature subset $F^{''} : F^{''} = \{\emptyset\}$;
2) Select a feature f_i from $F^{'}$ and add it to $F^{''}$, satisfying fitness function

$$f\left(F_k^{''}\right) = \min\left(f\left(F_{k-1}^{''}, f_1\right), f\left(F_{k-1}^{''}, f_2\right), \cdots, f\left(F_{k-1}^{''}, f_{l'-k+1}\right)\right),$$

in which $F_k^{''}$ stands for the current selected feature subset after the k^{th} iteration;
3) If $f\,(F_k^{''}) > f\,(F_{k-1}^{''})$, the algorithm is stopped, and $F_{k-1}^{''}$ is the final selected feature subset; Else if $k = l^{'}$,

the algorithm is stopped, and F_k'' is the final selected feature subset; Else return 2.

Furthermore, we need to find a fitness function $f(x)$ with monotonous decreasing. As we know, DDoS detection effect is mainly evaluated by two indexes: True Positive Rate (*TPR*) and False Positive Rate (*FPR*). *TPR* also called the Recall Rate (*RR*) in some fields, measures the proportion of actual positives (DDoS attacks) that are correctly identified as such. *FPR* is calculated as the ratio between the numbers of negative events (normal traffics) wrongly categorized as positives and the total number of actual negative events. The fitness function $f(x)$ of feature selection in DDoS attacks should be related to these two evaluating indexes *RR* and *FPR*. So, we need to find a method or function to consider these two evaluation indexes simultaneously, which is a multi-criteria decision-making problem. TOPSIS [37] is exactly a multi-criteria decision-making approach. This paper uses TOPSIS method, whose value is expressed by *T(RR, FPR)* as the evaluation factor. *T(RR,FPR)* is set to 1-*RR*+*FPR* to represent the fitness function of the SFS process. The smaller the *T(RR,FPR)* value is, the selected feature subset is much better for detecting attacks.

E. HYBRID FEATURE SELECTION ALGORITHM

This paper exploits parallelism to provide a hadoop-based feature selection method shown in Fig. 3, and the corresponding algorithm is shown as Algorithm 1. In this algorithm, the parameter *key* in the step 1 and step 5.a represents the label, whose values include 0 (labeled attack data), 1 (labeled normal data), 2 (unlabeled data) and 3 (test data). The parameter *value* represents the feature value corresponding to each *key*.

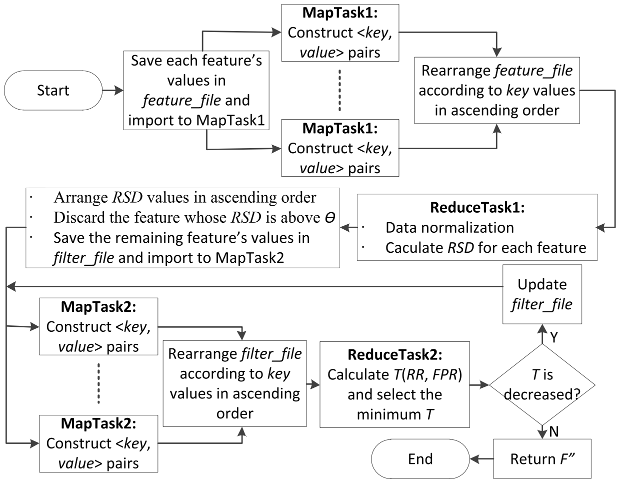

Start
Save each feature's values in feature_file and import to MapTask1
MapTask1: Construct <key, value> pairs
MapTask1: Construct <key, value> pairs
Rearrange feature_file according to key values in ascending order
ReduceTask1:
· Data normalization
· Caculate RSD for each feature
· Arrange RSD values in ascending order
· Discard the feature whose RSD is above Θ
· Save the remaining feature's values in filter_file and import to MapTask2
Update filter_file
MapTask2: Construct <key, value> pairs
MapTask2: Construct <key, value> pairs
Rearrange filter_file according to key values in ascending order
ReduceTask2: Calculate T(RR, FPR) and select the minimum T
T is decreased?
Y
N
Return F"
End

FIGURE 3. **The flow chart of hadoop-based hybrid feature selection method.**

Algorithm 1 Hadoop-Based Hybrid Feature Selection Algorithm.

```
Input: Threshold $\theta$, set of data points $X = \{x_1, x_2, \cdots, x_n\}$, set of labeled data $L \subset X$, candidate feature set $F = \{f_1, f_2, \cdots, f_l\}$, and each feature's values and labels are stored in each feature_file
Output: Selected feature subset $F'' = \{f_1, f_2, \cdots, f_{l''}\}$
Method:
    1. MapTask1:
        Context. write(key, value) //construct
        <key, value> pairs for the feature_file
    2. ReduceTask1:
      (2.a) Normalize each data point in $L$ and get dataset $L'$
      (2.b) Calculate $RSD(f_t)$ for each $f_t (t = 1,2,\ldots,l)$ in dataset $L'$
    3. Arrange RSD values in the ascending order
    4. if $RSD(f_t) < \theta$ (for $t = 1,2,\ldots,l$)
        add $f_t$ to $F'$
      else
        discard $f_t$
    5.Repeat
        (5.a) MapTask 2:
        Context. write(key, value) //construct <key, value> pairs for the filter_file
        (5.b) ReduceTask2:
        Select a feature $f_i$ from $F'$ and add it to $F''$,
        satisfying the value of $T(RR,FPR)$
        is minimum for the set $X$
        (5.c) if the value of $T(RR,FPR)$
        is decreased after adding the new feature $f_i$ in 5.a

              update filter_file
              return 5.a
            else
              end Repeat
      return $F''$
```

IV. SEMI-SUPERVISED K-MEANS DETECTION ALGORITHM USING HYBRID FEATURE SELECTION METHOD

By using the aforementioned hybrid feature selection method, the framework of our proposed DDoS detection prototype is depicted in Fig. 4. The detection framework consists of four major phases: (1) data preparation, where labeled and unlabeled training data and test data are prepared, (2) data preprocessing, where training and test data are preprocessed and important features are selected using proposed hybrid feature selection method, (3) model training, where the model is training using proposed SKM-HFS, and (4) attack detection and evaluation, where the trained model is used to detect DDoS attacks on test data and several indexes are used to evaluate detection performance. The contributions of this paper are shown as red font in Fig. 4. The proposed hybrid feature selection method is shown in Section III. This section will focus on the proposed detection model, which is based on the semi-supervised weighted k-means method using hybrid feature selection algorithm (SKM-HFS).

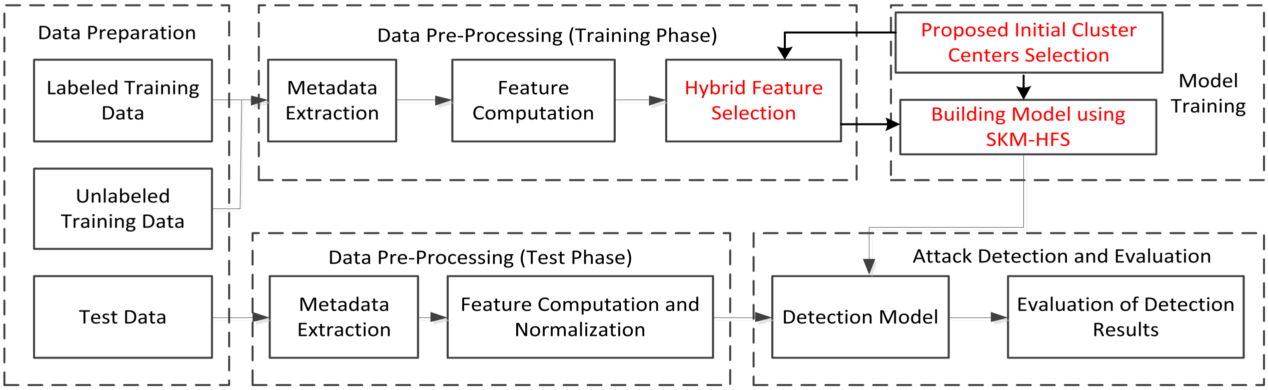

Data Preparation
Labeled Training Data
Unlabeled Training Data
Test Data
Data Pre-Processing (Training Phase)
Metadata Extraction
Feature Computation
Hybrid Feature Selection
Proposed Initial Cluster Centers Selection
Building Model using SKM-HFS
Model Training
Data Pre-Processing (Test Phase)
Metadata Extraction
Feature Computation and Normalization
Attack Detection and Evaluation
Detection Model
Evaluation of Detection Results

FIGURE 4. The framework of the proposed DDoS detection prototype.

A. SEMI-SUPERVISED K-MEANS CLUSTERING

K-means algorithm is based on iterative relocation that partition a dataset into k clusters, locally minimizing the average squared distance between the cluster data points and the cluster centers. The objective function based on this distance has been shown as equation (4). However, k-means clustering has the following disadvantages: (1) the selection of k value is very difficult to estimate; (2) the features of the clustering algorithm are equal weighting, which is inappropriate sometimes; (3) the initial cluster centers of the algorithm are randomly selected, and the selection of the center has a great influence on the clustering results.

As we know, k stands for the clustering number. This paper only needs to distinguish two traffics, normal and DDoS attacks. Besides, we use two categories of datasets to train our model, which are Inside Sniffer-Phase 5 of Lincoln Laboratory Scenarios (DDoS) 1.0 as attack dataset, and Week 1 of 1999 training data (attack free) as normal dataset. So, the clustering number k is set to two.

In order to solve the last two disadvantages of k-means algorithm, we use semi-supervised weighted k-means algorithm, which uses a small amount of labeled data to constrain the selection of the initial center points, and improve the classification accuracy of algorithm. With respect to feature weighting, this paper uses the *RSD* value defined by equation (3) to assign the corresponding weight to each feature. The smaller the *RSD* value is, the greater important the corresponding feature is. So the weights of selected feature subset $F'' = \{f_1, f_2, \ldots, f_{l''}\}$ are computed using equation (5).

$$\begin{aligned} W = \left\{w_1, w_2, \cdots, w_f\right\} = & \frac{(RSD\,(f_1))^{-1}}{\sum_{i=1}^{T} (RSD\,(f_t))^{-1}} \\ & \times \frac{(RSD\,(f_2))^{-1}}{\sum_{i=1}^{I} (RSD\,(f_t))^{-1}}, \cdots, \\ & \times \frac{(RSD\,(f_i))^{-1}}{\sum_{i=1}^{i} (RSD\,(f_r))^{-1}}\} \end{aligned} \tag{5}$$

The object function *SSE* of equation (4) is changed as equation (6).

$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} \sum_{m=1}^{l''} \|w_m \cdot (c_{mi} - x_m)\|^2 \tag{6}$$

B. PROPOSED INITIAL CLUSTER CENTERS SELECTION METHOD

As we know, the final clusters of k-means are highly dependent on the initial centers they are fed, and the random selection of the initial cluster center is easy to cause the local optimization of the algorithm. While, our previous study called MF-CKM DDoS detection method [22] uses the mean value of the labeled dataset as the initial center, which solves the local optimal problem to some extent, but it has a high sensitivity to the outlier data.

To solve the outlier problem, Wang and Liu [23] provides a point density-based method to select the initial cluster centers for intrusion detection. However, this method does not consider the situation when there is more than one point with maximum density, such as Fig. 5. This method will randomly select one point with the maximum density as the initial cluster center, which will cause the initial center offset.

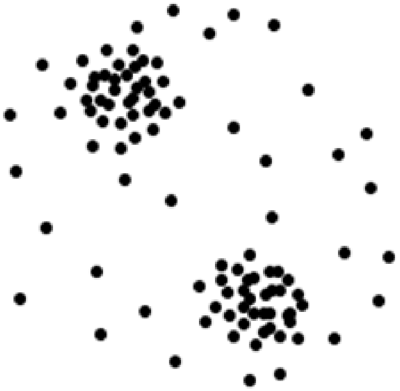


FIGURE 5. A scene with more than one maximum density point in a cluster.

To solve this problem of initial center offset, this paper provides an improved density-based method, given in Algorithm 3, in which the calculation method of the proposed algorithm parameter (Radius λ) is shown in Algorithm 2. The proposed method is based on the concept of point density as Definition 2.

Algorithm 2 Selection Algorithm of Radius (Take Labeled Normal Dataset for Example).

```
Input: Dataset $N$ with Normal Label
Output: Radius $\lambda$
Method:
    1. $x_i$ is the randomly selected data point in $N$,
       and calculate the Euclidean distance between $x_i$
       and other data points in $N$, expressed
       as $d_{i1}, \ldots, d_{i(n-1)}$, adds $n_i$ to dataset $P$
    2. Arrange distance sets from $d_{i1}$ to $d_{i(n-1)}$
       in ascending order and store in dataset $Q$
    3. $k = (n-1)/h$;    // $h > 1$
       $\lambda = d_{ik}$;         // set the $d_{ik}$ as the initial $\lambda$
       $Q.clear()$;        // empty the dataset $Q$
    4. Repeat
       (4.a) $x_j$ is another randomly selected data
       point from $N$-$P$, and calculate the Euclidean
       distance between $x_j$ and other data points in $N$,
       expressed as $d_{j1}, \ldots, d_{j(n-1)}$, adds $x_j$ to dataset $P$
       (4.b) Arrange above distance set in ascending order
       and store in dataset $Q$
       (4.c) $k = (n-1)/h$; $z = d_{jk}$;   $Q.clear()$;
           if $z < \lambda$, $\lambda = z$;
       (4.d) if $P = N$,
               end Repeat;
           else
               return 4.a
    return $\lambda$
```

Algorithm 3 Improved Density-based Initial Cluster Centers Selection Algorithm (Take Labeled Normal Dataset for Example).

```
Input: Radius $\lambda$, Dataset $N$ with Normal Label
Output: The Initial Center $c_1$ of the Normal Dataset
Method:
    1. Calculate the density of each data point
       $x_i$ with the radius $\lambda$, which is expressed
       respectively as $D(x_1, \lambda), \ldots, D(x_n, \lambda)$. ($x_i \in N$, $n$
       is the number of data points in $N$)
    2. Arrange the point density set in the
       descending order $D(x_i, \lambda) \geq D(x_j, \lambda) \geq \cdots \geq$
       $D(x_z, \lambda)$ $(i, j, \ldots, z \in [1, 2, \ldots, n])$
    3. If the data point of the maximum density is not
       unique, the mean value of all corresponding points
       with the maximum density is taken as $c_1$, or else the
       data point $x_i$ with the maximum density $D(x_i, \lambda)$
       is $c_1$
    return $c_1$
```

Definition 2: Given any point x_i within an n-dimensional ball of radius λ, the density of point x_i is the number of data points within radius λ, denoted by $D(x_i, \lambda)$.

C. SEMI-SUPERVISED K-MEANS ALGORITHM USING HYBRID FEATURE SELECTION METHOD

This paper provides a semi-supervised clustering detection algorithm, which is named as Semi-supervised K-Means algorithm using Hybrid Feature Selection method (SKM-HFS). This method uses the small amount of labeled data to guide the selection of initial cluster centers, and use other unlabeled data to train and form clusters.

The detection system based on SKM-HFS algorithm is divided into three parts: feature selection, model training and model testing. In the feature selection phase, feature subset is selected using the proposed hybrid method in Section III.

Algorithm 4 Semi-Supervised K-Means Algorithm Using Hybrid Feature Selection Method (SKM-HFS).

```
Input: Set of data points $X = \{x_1, x_2, \ldots, x_n\}$, set of
labeled data $L \subset X$, number of clusters $k$, selected feature
subset $F'' = \{f_1, f_2, \cdots, f_{l''}\}$ using Algorithm 1, feature
weights $W$ using equation (5).
Output: Disjoint $k$ partitioning $\{X_i\}_{i=1}^{k}$ of $X$ such that
SKM-HFS objective function is optimized
Method:
    1. Obtain $k$ initial cluster centers $\{c_i\}$ using
       Algorithm2 $(i = 1, \ldots, k)$
    2. Repeat until algorithm convergence
       (2.a) For $x_j \in L$, if $x_j \in L_i$ assign $x_j$ to the cluster $i$.
           For $x_j \notin L$, assign $x_j$ to the cluster $i^*$,
       satisfying $i^* = \arg\min_i \left( \sum_{m=1}^{l''} \left\| (x_{mj} - c_{mi}) \cdot w_m \right\|^2 \right)$
       (2.b) Update the center of cluster $i$:
             $c_i = \frac{\sum_{x_j \in X_j} x_j}{|X_i|}$
    return $\{X_i\}_{i=1}^{k}$
```

In the model training phase, the SKM-HFS algorithm is provided as Algorithm 4. Using the small number of labeled data, the initial cluster centers are calculated by Algorithm 2. The equation (6) is used to calculate the similarity between other unlabeled data and the initial cluster centers until this algorithm converges (SKM-HFS objective function is optimized).

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

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. DATASET AND DATA PREPROCESSING

We use three public datasets and one real-world dataset shown in the following experiments. In the proposed

SKM-HFS detection method, we make use of labeled data and unlabeled data in the training process. The labeled data is used to initialize the centers for each cluster, and unlabeled data is used to form the initial clusters. The labeled data includes labeled attack data and labeled normal data. We will explain how to select training data and test data for each dataset in the following.

1) PUBLIC DATASETS

The DARPA DDoS dataset [35] is stored as binary file and we use tShark tool to convert binary file to *txt* file, which contains numbers of vectors formed as (data record number, time, source IP, destination IP, source port, destination port, protocol type). We use Inside Sniffer – Phase 5 of Lincoln Laboratory Scenarios (DDoS) 1.0 as labeled attack data, Monday's Tcpdump data of Week 1 of 1999 DAPRA Intrusion Detection Evaluation Data Set as labeled normal data, Tcpdump data of Four-Hour Subset of Training Data (1998 DARPA Intrusion Detection Evaluation Data Set) as unlabeled data, and Monday's Tcpdump data (First Week of Test Data) as test data.

The CAIDA "DDoS attack 2007" dataset [38] is a sequence of anonymized traffic traces from a DDoS attack on 4 August 2007 (20:50:08 UTC to 21:56:16 UTC) containing approximately one hour traffic. The CAIDA "DDoS attack 2007" dataset is stored as *pcap* file and we use Python library scapy to convert *pcap* file to *txt* file, which includes the following protocol fields such as source IP, destination IP, source port, destination port, protocol type and etc. CAIDA dataset only contains DDoS attack data. We firstly exploit 30,000 records in CAIDA as labeled attack data, and use Monday's Tcpdump data of Week 1 of 1999 DARPA Intrusion Detection Evaluation Data Set as labeled normal data. Secondly, we exploit Tcpdump data of Four-Hour Subset of Training Data (1998 DARPA Intrusion Detection Evaluation Data Set) as unlabeled data. Finally, we make the mixture of remaining CAIDA data not in training set and Tuesday's Tcpdump data of Week 2 of 1999 DARPA Intrusion Detection Evaluation Data Set as test data, which include 50,000 records.

The CICIDS "DDoS attack 2017" dataset [39] is acquired by the Information Security Centre of Excellence (ISCX) of New Brunswick University in 2017 after a week of traffic capture, and DDoS attacks are part of the dataset. The CICIDS "DDoS attack 2017" dataset is stored as *csv* file and we convert it to *txt* file, which includes source IP, destination IP, source port, destination port, protocol ID and etc. CICIDS is fully labeled dataset, including attack data and normal data. We firstly exploit 30,000 abnormal data as labeled attack data, and 30,000 normal data as labeled normal data. Secondly, we make the mixture of 30,000 attack data and normal data removing labels as unlabeled data. Finally, we exploit another 50,000 removing labels' data, mixed as test data.

2) REAL-WORLD DATASET

In order to test our proposed method in real environment, we use a server as the victim host. Several benign users normally visit the server at first and then three attackers launch DDoS attacks to the server using *Hping3* tool during several minutes shown in Fig. 6. Meanwhile, Wireshark tool is used to capture all the traffics at the victim server including legitimate and attack packets, saved as *pcap* file. This captured dataset is fully labeled. We firstly exploit 30,000 records of data as labeled attack dataset and 30,000 labeled normal dataset. Secondly, we make the mixture of 100,000 attack data and normal data removing labels as unlabeled data. Finally, we exploit another 100,000 removing labels' data, mixed as test data.

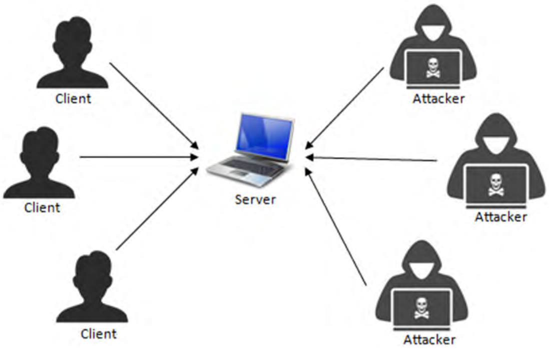


FIGURE 6. Test network architecture of real environment.

When processing the above data files, the feature values are obtained by using the sliding window principle shown as follows:

1) Suppose that M packets are in the same window unit, calculate feature values of this unit.
2) Remove the first packet of the window, add a new packet, and get the new sliding window unit.
3) Calculate feature values of the new window unit.
4) Return to step 2) until the end of packets.

The experiments are carried out using Eclipse neon.2, PyCharm and Hadoop 2.7.6 cluster running on a PC with Intel(R) Core (TM) i5-3210M, 2.50GHz CPU and 4.0GB RAM.

B. RESULTS AND PERFORMANCE ANALYSIS

1) FEATURE SELECTION

The features extracted from the above datasets are normalized, and then the candidate features are sorted by *RSD* value using equation (3). The *RSD* values of all candidate features on the above four datasets are ranked shown in Fig. 7 to Fig. 10.

How to determine the threshold θ in Algorithm 1? In the filter process of feature selection, the purpose of setting threshold θ is to filter out features that contribute little (the larger

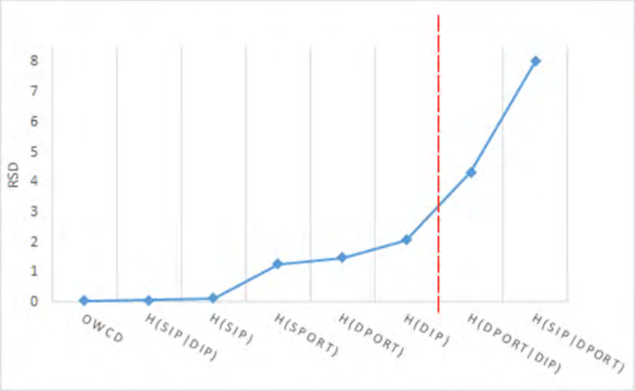


FIGURE 7. RSD values of candidate features on DARPA.

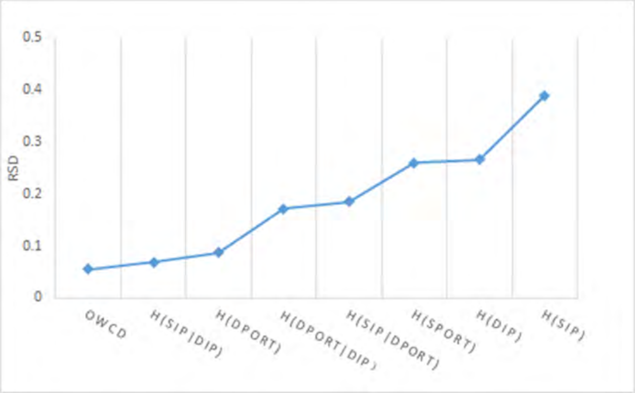


FIGURE 8. RSD values of candidate features on CAIDA.

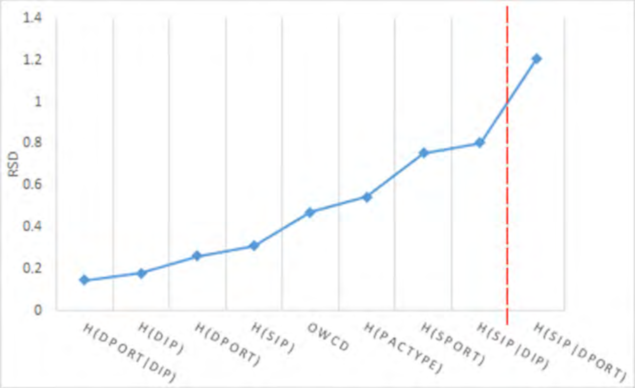


FIGURE 9. RSD values of candidate features on CICIDS.

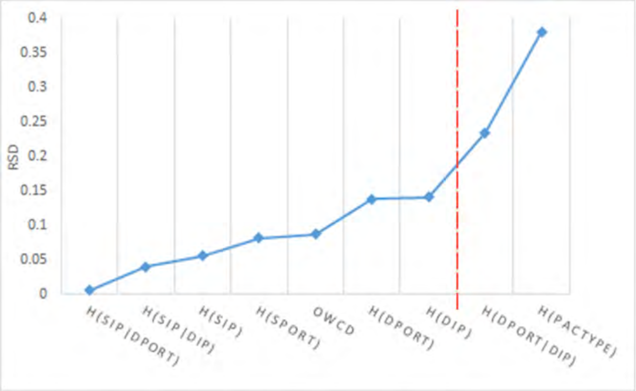


FIGURE 10. RSD values of candidate features on real-world dataset.

the RSD value is, the little the contribution of corresponding feature is) to attack detection and reduce the time of following feature subset searching. The setting of fixed threshold often requires expert experience and domain knowledge. However, in different scenarios or datasets (e.g., the four datasets mentioned in this paper), threshold is not always a fixed value, so it is necessary to provide some methods to assist threshold setting. In this paper, we propose to determine the threshold according to the way that the RSD value of one feature increases significantly compared with the RSD value of other features. That is to say, the RSD values of feature subset are arranged incrementally, and remove the features, whose RSD values increased sharply. That is, the slope of the RSD curve is significantly larger than that of the previous one. As shown in Fig. 7, Fig. 8 and Fig. 10, the slope at the dotted line increases sharply, and the RSD value corresponding to the dotted line is the threshold θ. So, we retain the feature whose RSD value is less than the threshold value. Of course, when there are few candidate features and no features contributed obviously little to attack detection, these features can be sorted without discarding, and all of them enter the feature subset searching process.

As we can see from Fig. 7, RSD value of the features on DARPA dataset rises sharply after the feature $H(Dip)$, so the candidate features $H(Dport|Dip)$ and $H(Sip|Dport)$ are discarded, and the initial feature subset is $F' = \{OWCD$, $H(Sip|Dip)$, $H(Sip)$, $H(Sport)$, $H(Dport)$, $H(Dip)\}$. In the same way, we can get the initial feature subsets of other three datasets shown in Fig.8 to Fig. 10.

After we get the initial feature subset F', F' will be input to the feature subset searching stage. The result of this stage is always evaluated by detection rate (RR) and false positive rate (FPR) simultaneously. In our algorithm, the detection effects of different feature subsets are evaluated by the fitness function $T(RR,FPR)$ provided in Section III. The smaller the $T(RR,FPR)$ value is, the selected feature subset is much better for detecting attacks.

The $T(RR,FPR)$ of different feature subsets are shown in Fig. 11 to Fig. 14. Fig. 11 shows that the minimum $T(RR,FPR)$ value is obtained by the third feature subset, and the final selected feature subset is $F'' = \{H(Sip)$, $OWCD$, $H(Dport)\}$. In the same way, we can get the final feature subsets of other three datasets shown in Fig. 12 to Fig. 14. The selected features of the proposed method used for all the four datasets are shown in Tab. 2.

Do different feature selection methods affect the detection effect? In this section, the proposed hybrid feature selection algorithm is compared with other feature selection methods [17]–[19], [21] for DDoS detection on the above datasets. We respectively implement the five different feature

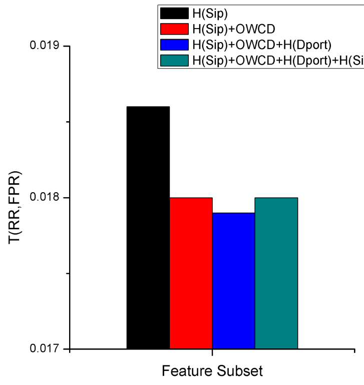


FIGURE 11. T(RR,FPR) of different feature subsets on DARPA.

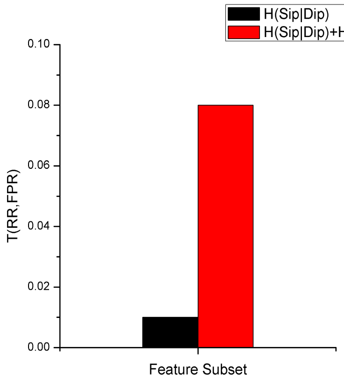


FIGURE 12. T(RR,FPR) of different feature subsets on CAIDA.

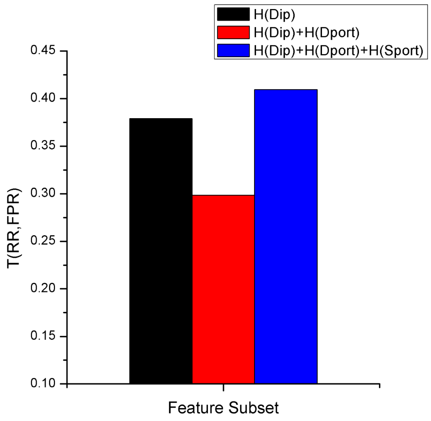


FIGURE 13. T(RR,FPR) of different feature subsets on CICIDS.

TABLE 2. Selected features for all the four datasets.

Dataset	Final Selected Feature Subset
DARPA	$\{H(Sip), OWCD, H(Dport)\}$
CAIDA	$\{H(Sip\|Dip)\}$
CICIDS	$\{H(Dip), H(Dport)\}$
Real-world Dataset	$\{H(Sip\|Dip), H(Sip)\}$

selection methods combined with the same detection method (k-means) in the same experimental environment, and calculate *RR* and *FPR* values for each method in different datasets.

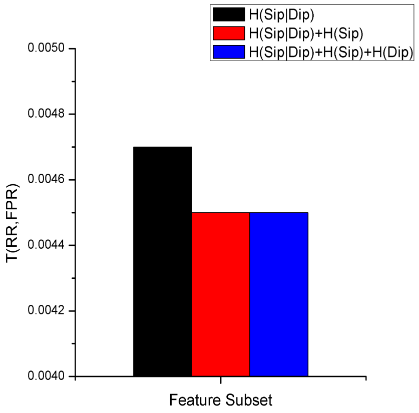


FIGURE 14. T(RR,FPR) of different feature subsets on real-world dataset.

The definitions for these metrics are

$$RR = \frac{TP}{TP+FN} \text{ and } FPR = \frac{FP}{FP+TN},$$

in which TP, FN, FP, and TN indicate successfully classified or misclassified samples. TP indicates successfully detected malicious samples while TN indicates correctly detected benign samples. FN indicates omitted malicious samples and FP indicates wrongly alarmed benign samples. Then we calculate the *T(RR, FPR)* using TOPSIS method for each method.

The performance of different feature selection methods is shown in Tab. 3. It can be seen from Tab. 3 that the proposed hybrid feature selection algorithm has the minimum $T(RR, FPR)$ value on these four datasets, which demonstrates that the proposed hybrid feature selection is mostly effective for DDoS detection. All in all, the performance of the feature selection method does affect the detection results, and the proposed method is superior to other existing methods.

2) INITIAL CLUSTER CENTERS SELECTION USING DIFFERENT METHODS

In this experiment, we want to verify the effect of different initial cluster center selection methods on detection performance. We compares four selection methods mentioned in section IV, namely random selection method (original k-means), average value-based selection method (MF-CKM) [22], density-based selection method [23] and proposed improved density-based selection method. The comparison experiments use the above four datasets in the condition of having outliers or not respectively.

The results are shown in Fig. 15 to Fig. 18, and the four numbers from 1 to 4 in the figures represent the four selection methods mentioned above. As can be seen from these figures, there is little difference of detection performance between four methods in the condition of no outliers, no matter what the datasets are. However, It can be clearly seen that TOPSIS values of the first two methods increases significantly when outliers exist, which indicates that outliers seriously affect these two algorithms and reduce their detection effects.

TABLE 3. Performance comparison of Different Feature Selection Methods using different datasets.

Feature Selection Method	DARPA			CAIDA			CICIDS			Real-world Dataset		
	RR (%)	FPR (%)	T(RR,FPR)	RR (%)	FPR (%)	T(RR,FPR)	RR (%)	FPR (%)	T(RR,FPR)	RR (%)	FPR (%)	T(RR,FPR)
CSE+DCF[17]	90.00	0.90	0.1900	95.40	1.80	0.064	90.58	40.70	0.5012	91.00	1.28	0.1028
Symmetric Uncertainty[18]	99.20	1.00	0.0180	99.60	4.70	0.051	96.00	30.00	0.34	99.50	0.00	0.005
Self-Correlation Coefficient[19]	99.45	1.30	0.0185	99.10	2.30	0.032	95.30	29.30	0.34	99.25	0.00	0.0075
Four Indicators Combined[21]	98.99	0.88	0.0189	99.10	2.30	0.032	90.60	27.50	0.369	98.12	0.00	0.018
Proposed Feature Selection	**99.59**	**1.38**	**0.0179**	**99.50**	**2.50**	**0.030**	**96.50**	**30.50**	**0.34**	**99.75**	**0.20**	**0.0045**

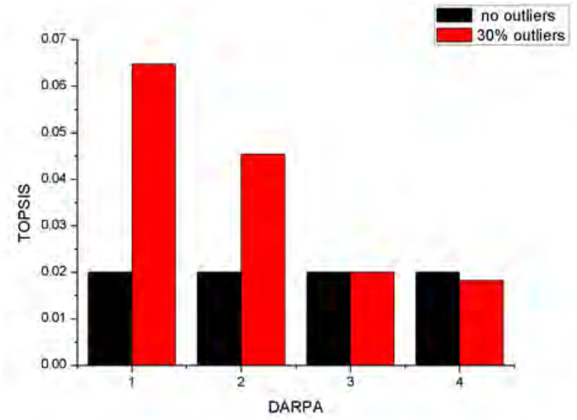


FIGURE 15. TOPSIS values of four initial center selection methods on DARPA dataset.

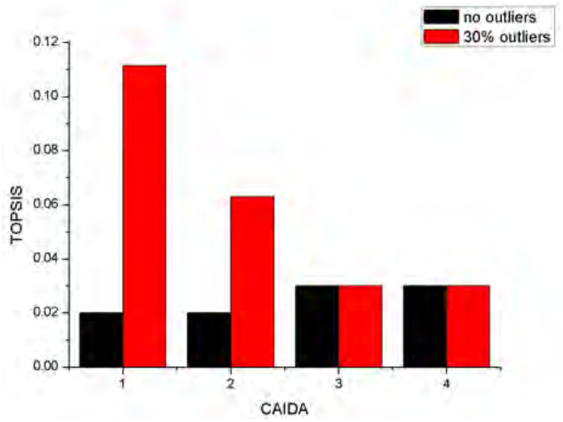


FIGURE 16. TOPSIS values of four initial center selection methods on CAIDA dataset.

As for the third method, outliers have little influence on it except for the fourth dataset. In contrast, our proposed selection method always performs well when outliers exist. Besides, if the condition of "more than one point with maximum density in the cluster" is satisfied, our proposed method performs better than the third one, shown in Fig. 15 and Fig. 18. If the condition is not satisfied, our method is not worse than the third one, shown in Fig. 16 and Fig. 17.

In conclusion, the proposed initial center selection method outperforms the existing methods when there are outliers, especially when there is more than one point with maximum density in the cluster.

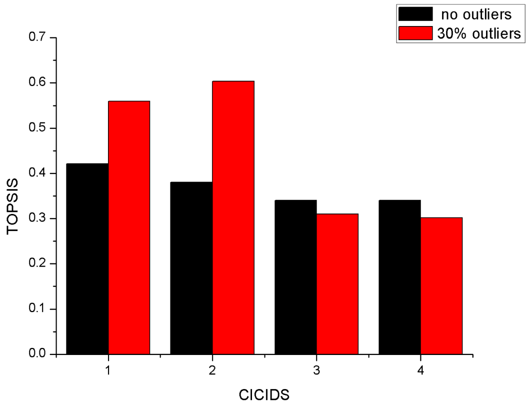


FIGURE 17. TOPSIS values of four initial center selection methods on CICIDS dataset.

3) PERFORMANCE COMPARISION OF DIFFERENT DETECTION METHODS ON DIFFERENT DATASETS

By using the above four datasets, this experiment compares the proposed detection method (SKM-HFS) with existing works. Because DARPA and CAIDA datasets are commonly used in DDoS detection experiments, we compare the ready-made result in each paper using the corresponding dataset with our result, including *RR* and *FPR*. For CICIDS datasets and our datasets, no papers have experimented with them and no ready-made results are available. Therefore, we implement several detection methods and compare them

TABLE 4. Performance of Different Detection Methods using different datasets.

Dataset	Algorithm	RR (%)	FPR (%)	T(RR,FPR)
DARPA	Al-Mamory[40]	52.17	0.68	0.4851
	Nguyen[7]	91.89	8.11	0.1622
	Chonka[14]	94	0.45	0.0645
	Kumar[42]	99.4	3.7	0.043
	Wu[15]	98.04	1.96	0.0392
	Bhaya[41]	96.29	0	0.0371
	Praman[3]	98.885	1.135	0.0225
	SKM-HFS	**99.68**	**1.40**	**0.0172**
CAIDA	Andrysiak[44]	93.20	12.1	0.189
	Luo[43]	94.87	3.85	0.0898
	Liu[46]	95	0	0.05
	Srihari[45]	97.95	1.75	0.038
	SKM-HFS	**99.00**	**0**	**0.01**
CICIDS	Nguyen[7]	1.14	59.03	1.5789
	Vijayasarathy[9]	0	0	1
	Liu[28]	0	0	1
	SKM-HFS	**98.86**	**28.72**	**0.2986**
Real-world Dataset	Vijayasarathy[9]	100	1.34	0.0134
	Liu[28]	99.84	0.31	0.0047
	Nguyen[7]	99.54	0	0.0046
	SKM-HFS	**99.75**	**0.20**	**0.0045**

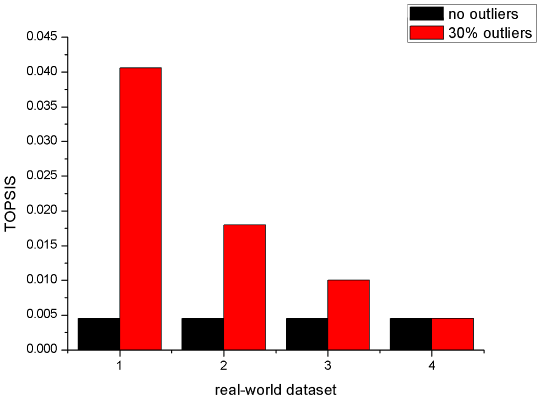


FIGURE 18. TOPSIS values of four initial center selection methods on real-world dataset.

TABLE 5. TRAINNING time between parallel and non-parallel algorithm.

Algorithm	Time Consumed (m)
Non-parallel	32.5
Parallel	2.9

TABLE 6. Detection time of different algorithms using real-world data.

Algorithm	Time Consumed (s)
Nguyen[7]	190006.314
Vijayasarathy[9]	1.024
Liu[28]	1.015
SKM-HFS	1.000

with our methods by calculating *RR* and *FPR*. Furthermore, we use TOPSIS method to calculate *T(RR, FPR)* for each method using different datasets. The comparison result is shown in Tab. 4. Each algorithm performance in the table is arranged in the descending order according to TOPSIS values in each dataset. We can see from Tab. 4 that the proposed method has the smallest $T(RR, FPR)$ value and outperforms all other methods in all of four datasets.

4) TIME COMPLEXITY ANALYSIS

The time of the proposed algorithm mainly includes training time and detection time. In the training phase, the time complexity of the proposed method is $O(k \cdot n^2)$, in which n represents the number of training samples and k stands for feature dimension. By using hadoop-based parallel algorithm, the training time is greatly reduced. The comparison of training time between parallel and non-parallel algorithm using real-world data is shown in Tab. 5.

In the detection phase, this paper compares the time consumed by different methods using real-world data, which is shown in Tab. 6. As can be seen from the table, the proposed method has the smallest detection delay.

VI. CONCLUSION AND FUTURE WORK

In order to tackle the issues of supervised and unsupervised based DDoS detection methods, this paper presents a semi-supervised weighted k-means detection method. Specially, we firstly provide a hadoop-based hybrid feature selection method to find the most effective feature set. Secondly, we present an improved density-based initial cluster centers selection method to solve the problem of outliers and local optimal of k-means clustering. Then, we propose a semi-supervised weighted k-means method using hybrid feature selection algorithm (SKM-HFS) to achieve better detection performance. Finally, we exploit DARPA DDoS dataset, CAIDA "DDoS attack 2007" dataset, CICIDS "DDoS attack 2017" dataset and real-world dataset to carry out the verification experiments. Three conclusions are drawn from the experiment results. Firstly, the hybrid feature selection method is much better than other feature selection methods using TOPSIS as evaluation factor. Secondly, the improved density-based initial cluster centers selection algorithm is the most effective in the presence of outliers and more than one maximum density point. Thirdly, the proposed detection method outperforms the benchmark in the respect of detection performance and TOPSIS.

In the future, more and larger datasets will be used to verify the advantages of the provided algorithm in terms of the generalization and robustness. In addition, the parallel ability of the proposed method will be further improved.

REFERENCES

[1] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system," *Expert Syst. Appl.*, vol. 67, pp. 296–303, Jan. 2017.

[2] J. Yu, Z. Li, H. Chen, and X. Chen, "A detection and offense mechanism to defend against application layer DDoS attacks," in *Proc. Int. Conf. Netw. Services (ICNS)*, Athens, Greece, Jun. 2007, p. 54.

[3] M. I. W. Praman, Y. Purwanto, and F. Y. Suratman, "DDoS detection using modified K-means clustering with chain initialization over landmark window," in *Proc. Int. Conf. Control, Electron., Renew. Energy Commun. (ICCEREC)*, Bandung, Indonesia, Aug. 2015, pp. 7–11.

[4] X. Qin, T. Xu, and C. Wang, "DDoS attack detection using flow entropy and clustering technique," in *Proc. 11th Int. Conf. Comput. Intell. Secur. (CIS)*, Shenzhen, China, Dec. 2015, pp. 412–415.

[5] L. Guo, P. Li, X. Di, and L. Cong, "The research of application layer DDoS attack detection based the model of human access," *Comput. Secur.*, vol. 6, pp. 11–14, Jun. 2014.

[6] E. Balkanli, J. Alves, and A. N. Zincir-Heywood, "Supervised learning to detect DDoS attacks," in *Proc. IEEE Symp. Comput. Intell. Cyber Secur. (CICS)*, Orlando, FL, USA, Dec. 2014, pp. 1–8.

[7] H. V. Nguyen and Y. Choi, "Proactive detection of DDoS attacks utilizing k-NN classifier in an anti-DDoS framework," *Int. J. Elect., Comput., Syst. Eng.*, vol. 4, no. 4, pp. 247–252, Feb. 2010.

[8] P. Xiao, W. Qu, H. Qi, and Z. Li, "Detecting DDoS attacks against data center with correlation analysis," *Comput. Commun.*, vol. 67, pp. 66–74, Aug. 2015.

[9] R. Vijayasarathy, S. V. Raghavan, and B. Ravindran, "A system approach to network modeling for DDoS detection using a Naive Bayesian classifier," in *Proc. 3rd Int. Conf. Commun. Syst. Netw.*, Bangalore, India, Jan. 2011, pp. 1–10.

[10] Y. Bouzida and F. Cuppens, "Detecting known and novel network intrusions," in *Proc. IFIP Int. Inf. Secur. Conf.*, Karlstad, Sweden, 2006, pp. 258–270.

[11] J. Li, Y. Liu, and L. Gu, "DDoS attack detection based on neural network," in *Proc. 2nd Int. Symp. Aware Comput.*, Tainan, China, Nov. 2010, pp. 196–199.

[12] J. Cheng, M. Li, X. Tang, V. S. Sheng, Y. Liu, and W. Guo, "Flow correlation degree optimization driven random forest for detecting DDoS attacks in cloud computing," *Secur. Commun. Netw.*, vol. 2018, Nov. 2018, Art. no. 6459326.

[13] K. J. Singh, K. Thongam, and T. De, "Entropy-based application layer DDoS attack detection using artificial neural networks," *Entropy*, vol. 18, no. 10, pp. 350–366, 2016.

[14] A. Chonka, J. Singh, and W. Zhou, "Chaos theory based detection against network mimicking DDoS attacks," *IEEE Commun. Lett.*, vol. 13, no. 9, pp. 717–719, Sep. 2009.

[15] X. Wu and Y. Chen, "Validation of chaos hypothesis in NADA and improved DDoS detection algorithm," *IEEE Commun. Lett.*, vol. 17, no. 12, pp. 2396–2399, Dec. 2013.

[16] S. M. T. Nezhad, M. Nazari, and E. A. Gharavol, "A novel DoS and DDoS attacks detection algorithm using ARIMA time series model and chaotic system in computer networks," *IEEE Commun. Lett.*, vol. 20, no. 4, pp. 700–703, Apr. 2016.

[17] A. R. Yusof, N. I. Udzir, A. Selamat, H. Hamdan, and M. T. Abdullah, "Adaptive feature selection for denial of services (DoS) attack," in *Proc. IEEE Conf. Appl., Inf. Netw. Secur. (AINS)*, Miri, Malaysia, Nov. 2017, pp. 81–84.

[18] E. Balkanli, A. N. Zincir-Heywood, and M. I. Heywood, "Feature selection for robust backscatter DDoS detection," in *Proc. IEEE 40th Local Comput. Netw. Conf. Workshops (LCN Workshops)*, Clearwater Beach, FL, USA, Oct. 2015, pp. 611–618.

[19] L. Zi, J. Yearwood, and X.-W. Wu, "Adaptive clustering with feature ranking for DDoS attacks detection," in *Proc. 4th Int. Conf. Netw. Syst. Secur.*, Melbourne, VIC, Australia, Sep. 2010, pp. 281–286.

[20] H. Jiang, S. Chen, H. Hu, and K. Qian, "Lightweight detection approach of DDoS attacks based on GAIG algorithm for feature selection," *Appl. Res. Comput.*, vol. 33, no. 2, pp. 502–506, Feb. 2016.

[21] O. Osanaiye, H. Cai, K. K. Choo, A. Dehghantanha, Z. Xu, and M. Dlodlo, "Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing," *EURASIP J. Wireless Commun. Netw.*, vol. 1, pp. 130–139, May 2016.

[22] Y. Gu, Y. Wang, Z. Yang, F. Xiong, and Y. Gao, "Multiple-features-based semisupervised clustering DDoS detection method," *Math. Problems Eng.*, vol. 2017, Dec. 2017, Art. no. 5202836.

[23] Q. Wang and S. H. Liu, "Application research of improved K-means algorithm in intrusion detection," *Comput. Eng. Appl.*, vol. 51, no. 17, pp. 124–127, 2015.

[24] N. Hoque, H. Kashyap, and D. K. Bhattacharyya, "Real-time DDoS attack detection using FPGA," *Comput. Commun.*, vol. 110, pp. 48–58, Sep. 2017.

[25] X. Ma and Y. Chen, "DDoS detection method based on chaos analysis of network traffic entropy," *IEEE Commun. Lett.*, vol. 18, no. 1, pp. 114–117, Jan. 2014.

[26] S. Behal and K. Kumar, "Detection of DDoS attacks and flash events using information theory metrics—An empirical investigation," *Comput. Commun.*, vol. 103, pp. 18–28, May 2017.

[27] M. Sachdeva, K. Kumar, and G. Singh, "A comprehensive approach to discriminate DDoS attacks from flash events," *J. Inf. Secur. Appl.*, vol. 26, pp. 8–22, Feb. 2016.

[28] Y. Liu, J. Yin, J. Cheng, and B. Zhang, "Detecting DDoS attacks using conditional entropy," in *Proc. Int. Conf. Comput. Appl. Syst. Modeling (ICCASM)*, Taiyuan, China, Oct. 2010, pp. 278–282.

[29] M. Baskar, T. Gnanasekaran, and S. Saravanan, "Adaptive IP traceback mechanism for detecting low rate DDoS attacks," in *Proc. IEEE Int. Conf. Emerg. Trends Comput., Commun. Nanotechnol. (ICECCN)*, Tirunelveli, India, Mar. 2013, pp. 373–377.

[30] S. Behal and K. Kumar, "Detection of DDoS attacks and flash events using novel information theory metrics," *Comput. Netw.*, vol. 116, pp. 96–110, Apr. 2017.

[31] N. Furutani, T. Ban, J. Nakazato, J. Shimamura, J. Kitazono, and S. Ozawa, "Detection of DDoS backscatter based on traffic features of darknet TCP packets," in *Proc. 9th Asia Joint Conf. Inf. Secur.*, Wuhan, China, Sep. 2014, pp. 39–43.

[32] N. A. Singh, K. J. Singh, and T. De, "Distributed denial of service attack detection using naive Bayes classifier through info gain feature selection," in *Proc. Int. Conf. Inform. Anal.*, Aug. 2016, p. 54.

[33] N. Hoque, D. K. Bhattacharyya, and J. K. Kalita, "Denial of service attack detection using multivariate correlation analysis," in *Proc. 2nd Int. Conf. Inf. Commun. Technol. Competitive Strategies*, Hangzhou, China, Mar. 2016, p. 100.

[34] I. L. Meitei, K. J. Singh, and T. De, "Detection of DDoS DNS amplification attack using classification algorithm," in *Proc. Int. Conf. Inform. Anal.*, Pondicherry, India, Aug. 2016, p. 81.

[35] *Lincoln Laboratory Scenario (DDoS) 1.0 of DARPA Intrusion Detection Evaluation Data Sets*. Accessed: 2000. [Online]. Available: http://www.ll.mit.edu/ideval/data/2000/LLS_DDOS_1.0.html

[36] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *Proc. ACM Conf. Appl., Technol., Archit., Protocols Comput. Commun. (SIGCOMM)*, Philadelphia, PA, USA, Aug. 2005, pp. 217–228.

[37] A. Keikha and H. M. Nehi, "A complex method based on TOPSIS and Choquet integral to solve multi attribute group decision making problems with interval type-2 fuzzy numbers," in *Proc. 4th Iranian Joint Congr. Fuzzy Intell. Syst. (CFIS)*, Zahedan, Iran, Sep. 2015, pp. 1–5.

[38] *The CAIDA UCSD DDoS Attack 2007 Dataset*. Accessed: Aug. 2007. [Online]. Available: http://www.caida.org/data/passive/ddos-20070804_dataset.xml

[39] *The CICIDS DDoS Attack 2017 Dataset*. Accessed: 2017. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html

[40] S. O. Al-Mamory and Z. M. Algelal, "A modified DBSCAN clustering algorithm for proactive detection of DDoS attacks," in *Proc. Annu. Conf. New Trends Inf. Commun. Technol. Appl. (NTICT)*, Baghdad, Iraq, Mar. 2017, pp. 304–309.

[41] W. Bhaya and M. Ebadymanaa, "DDoS attack detection approach using an efficient cluster analysis in large data scale," in *Proc. Annu. Conf. New Trends Inf. Commun. Technol. Appl. (NTICT)*, Baghdad, Iraq, Mar. 2017, pp. 168–173.

[42] P. A. R. Kumar and S. Selvakumar, "Distributed denial of service attack detection using an ensemble of neural classifier," *Comput. Commun.*, vol. 34, no. 11, pp. 1328–1341, 2011.

[43] H. Luo, Y. Lin, H. Zhang, and M. Zukerman, "Preventing DDoS attacks by identifier/locator separation," *IEEE Netw.*, vol. 27, no. 6, pp. 60–65, Nov./Dec. 2013.

[44] T. Andrysiak, Ł. Saganowski, and M. Choraś, "DDoS attacks detection by means of greedy algorithms," in *Image Processing and Communications Challenges 4*. Berlin, Germany: Springer, 2013, pp. 303–310.

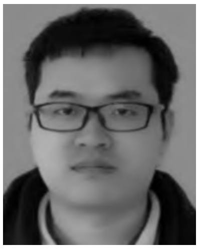
[45] V. Srihari and R. Anitha, "DDoS detection system using wavelet features and semi-supervised learning," in *Security in Computing and Communications*. Berlin, Germany: Springer, 2014, pp. 291–303.

[46] H. Liu, Y. Sun, V. C. Valgenti, and M. S. Kim, "TrustGuard: A flow-level reputation-based DDoS defense system," in *Proc. IEEE Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2011, pp. 287–291.

YONGHAO GU received the Ph.D. degree from the Beijing University of Posts and Telecommunications, China, in 2007, where he is currently a Lecturer with the Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, School of Computer. His current research interests include network security and privacy preservation.

KAIYUE LI received the B.S. degree from the Hebei University of Technology, China, in 2017. She is currently pursuing the M.Eng. degree with the Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, School of Computer, Beijing University of Posts and Telecommunications, China. Her current research interest includes network security.

ZHENYANG GUO received the B.S. degree in software engineering from Heilongjiang University, China, in 2017. He is currently pursuing the M.Eng. degree with the Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, School of Computer, Beijing University of Posts and Telecommunications, China. His current research interests include machine learning, network security, and detection on botnet.



YONGFEI WANG received the B.S. degree from Hebei North University, China, in 2012. He is currently pursuing M.Eng. the degree with the Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, School of Computer, Beijing University of Posts and Telecommunications, China. His current research interests include network security and privacy preservation.