Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Efficient Network Anomaly Detection Using Selective Features

JIEWEN MAO, YONGQUAN HU, DONG JIANG, TONGQUAN WEI,(Senior Member, IEEE), FUKE SHEN

School of Computer Science and Technology, East China Normal University, Shanghai 200062, China

Corresponding author: Tongquan Wei (e-mail: tqwei@cs.ecnu.edu.cn).

ABSTRACT Network traffic anomaly detection has become a necessity due to proliferation of information technologies across safety critical applications. However, conventional traffic anomaly detection mechanisms primarily rely on traffic data either with only headers of packets, which makes the accuracy depressed, or with very high dimension, whose cost of high accuracy is lower efficiency. In this paper, we propose an efficient network anomaly detection mechanism by selecting essential features that can quickly detect multiple types of anomalies without compromising the detection accuracy. The proposed scheme selects key features by clustering and ranking all variables, and classifies network traffic to detect anomalies based on selected features. Experimental results show that the proposed scheme can reduce the training time from about 300 seconds to about 200 seconds, and improves various classification metrics by up to 2% on CIC-IDS-2018 data set. It also can improve classification metrics by up to 12% as compared to benchmark method under the condition of the same scale of selected features.

INDEX TERMS feature selection, abnormal detection, feature clustering, correlation coefficient, information gain, decistion tree

I. INTRODUCTION

Network traffic anomaly detection has become a necessity due to proliferation of information technologies across safety critical applications. In recent years, cyber attacks on public information services have become more frequent and have had more serious consequences. On March 2015, GitHub faced a massive DDoS attack. The attack lasted for one week and caused significant damages [1]. On October 2016, Dyn's DNS servers was attacked for two hours, during that time, internet users directed to Dyn servers were unable to reach some of the marquee brands of the internet [2]. On May 2017, the WannaCry ransomware attack bursted and over 200,000 compromised computers across 150 countries were influenced by this virus and economic losses from the cyber attack could reach up to 4 billion dollars [3]. Therefore, how to quickly detect network anomalies has become a key issue for researchers and enterprises.

Many researchers have proposed various methods to solve the problem of quickly detecting network anomalies. Wang et al. [4] proposed a dynamic Multilayer-Perceptron (MLP)-based attack detection method. The feature selection process of their work uses backward selection filter method. This method can remove the features which makes the accuracy of MLP decreasing more than threshold and the remainings are the selected features. Urmila et al. [5] proposed a distributed collaboration detection scheme that combines the advantages of anomaly and signature based method by capturing the packet header in real time. Al-Hawawreh [6] discussed the SYN flood attack in virtual cloud. Network anomalies are detected based on new features that extracted from TCP/IP header. Alsharafi et al. [11] investigated a normal profile updating method (NPUM) for enhancing the PHAD based IDS model, which updates normal profile of anomaly IDS using further processing of both the normal and abnormal data identified by anomaly detector. Above methods only use header information of flow packets, while diverse types of network anomalies cannot be distinguish effectively by only using packet headers. The malicious users may construct and send the packets elaborately to escape the detection from intrusion detection systems (IDS). Once these packets pass the prevension and propagate in the network, the target computer or network device will be compromised.

To solve this issue, significant amount of works used more statistical information extracted from aggregated flows with the header information to detect network anomalies. Fer-

nandes et al. [10] used digital signature of network segment using flow analysis to detect abnormal events. According to the results of this work, Ant Colony Optimization takes higher accuracy while PCA takes shorter detection time. All of methods in [10] use data in sFlow format. The authors in [7] mentioned that using all features in UNSW-NB15 data set [8] can only reach 76.9% of accuracy and its execution time is longer than the proposed strategies. A comprehensive survey [9] reviewed a series of machine-learning-based network anomaly detection methods. This survey mentioned that using machine-learning-based methods to learn a large data set can bring high detection rate for acknowledged attacks, but it may cost high resource consumption and may cause over-fitting. It is clear when more features are used, it will be easier to distinguish between different types of network anomalies, and the accuracy of detection will be higher.

However, improving the ability of these methods to detect network anomalies will reduce their detection efficiency. This will cause the problem of dimensional explosion. Network flows have more than 30 features in the packet header alone. If various derived statistical features are considered, the number of features in each flow will increase significantly. There may also be correlations or derivations between these features. If we use all features for statistical learning, on the one hand, it will reduce the efficiency of anomaly detection, on the other hand, it will make us unable to focus on determining the essential characteristics of network anomalies.

In this paper, we propose an efficient network anomaly detection technique. The key of which is a new feature selection process to reduce the number of features of network flow data and maintain high accuracy of the detection model. The main contributions of this paper are as follows:

- This paper proposes a new feature selection algorithm using clustering and ranking techniques. Feature clustering algorithms are based on Pearson correlation coefficients and combine features with potential correlations into the same cluster. Feature ranking algorithm uses both information gain and gain ratio. These two algorithms can identify key features in network traffic data.
- This paper builds a network traffic anomaly detection model based on a decision tree classifier. Based on the key features of the model, the C4.5 decision tree classifier is used to more quickly generate a decision tree of a specified size, so that the model can detect network traffic anomalies faster while ensuring the accuracy of the detection.
- Experiments show that on the CIC-IDS-2018 dataset, our proposed method can reduce the training time by about 100 seconds and improve various classification metrics by up to 2%. To compare with the benchmark selection methods, under the condition of the same scale of selected features, our method can improve these classification metrics by up to 12%.

The remaining parts of the paper are organized as follows: Section II describes related works. Section III describes the formalization of feature selection problem. Section IV introduces proposed feature clustering and ranking scheme, then explain the building of detection model and the difference between proposed scheme and dimensional reduction methods. Section V shows the experiment results and the correlated discussions. Finally Section VI concludes this paper and indicates future works.

II. RELATED WORKS

In past several years, researchers have proposed a large amount of network anomaly detection schemes. All of them take feature selection techniques into account to reduce the superfluous traffic features.

A series of researches [12]–[15], [21] considered using information theory and statistical correlations between features of traffic data. Bajaj et al. [12] proposed a method for feature selection using information gain, information gain ratio, and correlation coefficients, and compared accuracy of these three feature selection methods when applied to machine learning algorithms such as Naive Bayes, decision tree, and support vector machine. However, this work did not comprehensively use these three feature selection methods, but applied them independently. In addition, the data set used in this work is the NSL-KDD data set, which is a fairly old data set. The types of network attacks contained in it are outdated and are not suitable for the current network environment. Wahba et al. [14] proposed a hybrid feature selection method, which is also based on the correlation between features and information gain. The author uses greedy search algorithm to select 10 features to form a new feature set, then uses Adaboost algorithm combined with information gain to find the best feature. The highest F-measure was achieved by using this method, on the NSL-KDD dataset. On the one hand, this method does not take into account the impact of the imbalanced distribution of classes on the detection algorithm. On the other hand, the method does not perform well in detecting U2R and R2L attacks. Next, Kumar et al. [13] also used correlation coefficients, information gains, and information gain ratios for feature selection. The author calculates and ranks the correlation coefficients of each feature and class labels to find the feature combination with the greatest conditional probability. Compared with the Naive Bayes method, this method greatly improves the accuracy, and reduces the detection time compared to the decision tree algorithm. However, this method only considers the correlation between each feature and the class labels, while the correlation between features are not considered. Moreover, compared with decision trees, their method has no advantages in terms of accuracy and false alarm rate. The authors of [15] used information gain and Gini covariance as feature selection methods. It finally selected 26 features from the KDD CUP 99 dataset. However, on the one hand, the method uses a deprecated data set. On the other hand, the method performs

poorly when detecting U2R-type attacks. The literature [21] adopted the feature subset selection method based on correlation and wrapper method to reduce 41 features to 25 in the NSL-KDD dataset. In this work, the data set is divided into three subsets according to the protocol type, which are respectively applied to CfsSubsetEval and WrapperSubsetEval procedures. Then the results of these two procedures are combined to obtain a new feature subset. In our opinion, it is not a good practice to group actual network traffic by protocol type, which will lead to overfitting of the model.

Except information theory-based methods, there are also other feature selection techniques [16], [17], [20] used for network traffic classification or anomaly detection. The literature [16] studied the problem of reducing false alarm detection rate through feature selection. This work proposes Simple Hybrid Feature Selection (SHFS). The author uses a combination of multiple feature selection methods to generate the feature subset with the most occurrences, then uses these features for multiple classifiers for comparison. Experimental results show that the proposed method effectively reduces the false alarm rate and has a small improvement on other classification metrics. However, this method is simply a hybrid of several existing feature selection methods, and these feature selection methods have not been properly composed. Genetic-Algorithm-based feature selection [17] finds the optimal features from NSL-KDD data set, which use one-point crossover for the Genetic Algorithm parameters instead of two-point crossover. The results show the proposed approach performs better in classification rate and the training time compared to several other classifiers. Dong et al. [20] studied Internet video traffic data and used a hierarchical method to classify these data. The literature uses a consistency-based method and data mining algorithms for feature selection. In a real network environment, this method can better classify Internet video data and reduce the time overhead. However, the algorithm has high complexity and is slightly weaker than traditional methods in recall rate.

In addition, two other recent literatures have adopted other machine learning methods for feature selection. Shi et al. [18] proposed a novel feature extraction and selection approach to provide the optimal and robust features for traffic classification, which based on multifractal features, the observation of the multifractal features and the analysis of PCA-based feature selection. The results show the approach achieves better classification performance, lower runtime performance and more effective for real-time traffic classification compared to the TLS features. Moreover, the authors then propose a new feature optimization approach based on deep learning and Feature Selection (FS) techniques [19] to provide the optimal and robust features for traffic data sets. The results show the approach achieves the best classification performance and relatively higher runtime performance compared with the approaches used in the previous work.

In summary, we need a method that is able to organically compose various feature selection methods and detect anomalies in new network traffic data.

III. PROBLEM DEFINITION AND SOLUTION

We denote

$$D = \begin{pmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \dots & \boldsymbol{x}_m \end{pmatrix}^T$$

as the data set with m instances, where

$$\boldsymbol{x}_i = \begin{pmatrix} f_{1i} & f_{2i} & \dots & f_{ni} & c_i \end{pmatrix}$$

where f_{ji} is the value of jth feautre of vector $\boldsymbol{x_i}$, and $f_j \in F = \{f_1, f_2, \dots, f_n\}$ is the feature set. c_i is the class label of $\boldsymbol{x}_i$ and $c_i \in C = \{c_1, c_2, \dots, c_k\}$, where C is the set of class labels.

The feature selection problem [22] can be described as a 6-tuple $FS = \langle D, F, C, S, fs, E \rangle$, where D, F, C are data set, feature set and class label set respectively. $S = \{s_1, s_2, \dots, s_l\}$, $l = 2^n - 1$ is the search space, which contains all subsets can be constructed from F with $s_i = \{f_j, f_k, \dots, f_l\}$, $(1 \leqslant j \neq k \neq l \leqslant n)$. E is the evaluation measure and fs represents the function of process of feature selection: $fs : F \to S$.

The target of the function fs, which is the proposed algorithm, is to find the best feature subset $\hat{F} \subset F$. The feature subset should satisfy following conditions:

- Every feature $f \in \hat{F}$ is independent with others. It means that every f should not be calculated from other features.
- The feature subset $\hat{F}$ is the least set of s_l with given count of features l. This feature subset should has enough information to determine whether a flow is an attack or not. If the feature set add or remove any other features, the result of detection will deteriorate.

The workflow of proposed solution is shown in Figure 1. The first and the most important part is the feature selection procedure. It contains three parts: At first we prepocess the data set. Then all correlated features are aggregated by proposed feature clustering algorithm. Next, all cluster center features are ranked and sorted by proposed feature ranking algorithm.

After the three-steps processing, the refined data subset is divided into training set and test set, then they are trained and tested via decision tree classifier and finally our detection model is generated.

IV. PROPOSED ANOMALY DETECTION SCHEME

In this section, we propose our hierarchical feature selection method in turn. We first introduce the source of data set and the preprocessing precedure. Next we explain our feature clustering algorithm and feature ranking algorithm respectively. After that we simply introduce decision tree classifier, which is used as our model-building tools. Finally, we analysis the difference of the theory of our method and dimensionality reduction algorithms like PCA.

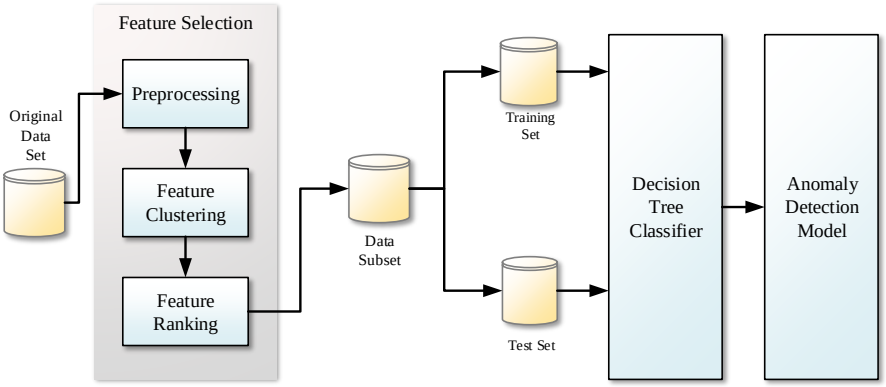

FIGURE 1. The workflow of proposed feature selection scheme.

A. PREPROCESSING DATASET

The data set being studied is CIC-IDS-2018 [23]. It is generated on a simulated network topology on the AWS computing platform. The network topology is shown as Figure 2. It has 5 subnet and 1 attacker network. The data set consists of seven different attack scenarios: Brute-force, Heartbleed, Botnet, DoS, DDoS, Web attacks, and infiltration of the network from inside. It includes the captured network traffic and system logs of each machine, along with 76 features extracted from the captured traffic using CICFlowMeter-V3 [24]. These 76 features are shown in Table 1. Their detailed meaning is listed in [23].

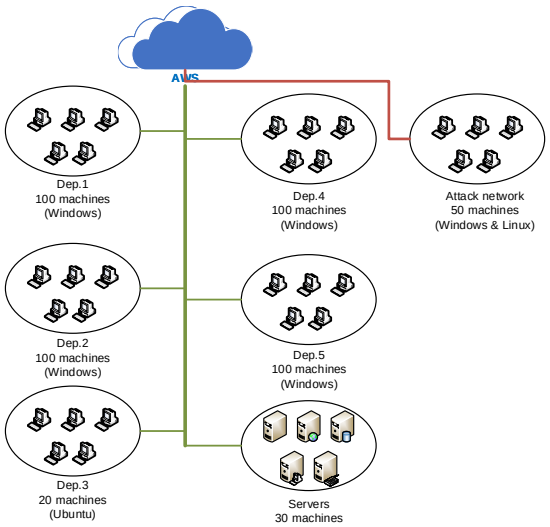

FIGURE 2. Simulated network topology on AWS computing platform

In order to detect all types of attacks as much as possible, we integrated the network traffic data that was originally scattered in each day, and extracted 20% of the data as our main data set while maintaining the label ratio. At the same time, in order to ensure the generalization ability of the model, we randomly select and generate data sets as many times as possible. Finally we get 10 stratified-sampled data set files.

At this moment, the data set is still unavailable because there are some useless nominal features which are not suitable for statistical analyzing, and there may be missing values in the data set. We must remove these features to prevent them interfering proposed algorithms. Our preprocessing strategy is described as follows:

1) Data cleaning

The target of feature selection procedure is to find the decisive statistical features which can precisely distinguish different attack typs. So the traditional 5-tuple, i.e. source IP address, source port, destination IP address, destination port and protocol cannot be used to our algorithm. These nominal features will be removed so that we can focus on other statistical features. We also remove timestamp feature because our algorithm focuses on the type of attacks rather than the time characteristics.

Many missing values also exist in the data set. There are many reasons for missing values. Some flows cannot be calculated in certain features. For example, if the duration time of a flow is too small even equals 0, the values of two features named “Flow Bytes/s” and “Flow Packets/s” can be NaN or infinity. It is a difficult work to examine every missing value, and they cannot be filled using interpolation because the data set is composed randomly. In this situation, we remove these rows containing missing values to ensure our algorithm running normally.

Note that the removed features are only in the context of this paper. These features may be useful in other detection methods.

2) Remove all zero-variance features

Variance is used to describe the discreteness of a variable. In a data set, if the variance of a feature is zero, it means

TABLE 1. 76 Features in CIC-IDS-2018 data set

Features in CIC-IDS-2018 data set			
Flow Duration	Total Forward Packets	Total Backward Packets	Total Length Forward Packets
Total Length Backward Packets	Forward Packet Length Max	Forward Packet Length Min	Forward Packet Length Mean
Forward Packet Length Std	Backward Packet Length Max	Backward Packet Length Min	Backward Packet Length Mean
Backward Packet Length Std	Flow Bytes/s	Flow Packets/s	Flow IAT Mean
Flow IAT Std	Flow IAT Max	Flow IAT Min	Forward IAT Total
Forward IAT Mean	Forward IAT Std	Forward IAT Max	Forward IAT Min
Backward IAT Total	Backward IAT Mean	Backward IAT Std	Backward IAT Max
Backward IAT Min	Forward PSH Flags	Backward PSH Flags	Forward URG Flags
Backward URG Flags	Forward Header Length	Backward Header Length	Forward Packets/s
Backward Packets/s	Packet Length Min	Packet Length Max	Packet Length Mean
Packet Length Std	Packet Length Var	FIN Flag Count	SYN Flag Count
RST Flag Count	PSH Flag Count	ACK Flag Count	URG Flag Count
CWE Flag Count	ECE Flag Count	Down/Up Ratio	Packet Size Avg
Forward Seg Size Avg	Backward Seg Size Avg	Forward Byts/b Avg	Forward Packets/b Avg
Forward Blk Rate Avg	Backward Byts/b Avg	Backward Packets/b Avg	Backward Blk Rate Avg
Subflow Forward Packets	Subflow Forward Byts	Subflow Backward Packets	Subflow Backward Byts
Init Forward Win Byts	Init Backward Win Byts	Forward Act Data Packets	Forward Seg Size Min
Active Mean	Active Std	Active Max	Active Min
Idle Mean	Idle Std	Idle Max	Idle Min

that this feature has only one value. Thus this feature cannot import any new information to help training the model. We remove these features to refine the data set.

3) Encode labels

Most anomaly detection models perform as classifiers which predict output labels by calculating the probabilities of every label in the label set. The models use input data to train a function. Both input data and output data are often numeric. However in our data set, the labels are text which cannot be processed by detection model. Thus we encoder the labels and map them to numeric values.

B. FEATURE CLUSTERING BASED ON PEARSON CORRELATION COEFFICIENT

According to our observation, many features are potentially linearly related. For example, in Figure 3, we pick up two pairs of features to show their potential linear correlation. The left part shows that features "Flow IAT Mean" and "Forward IAT Mean" may have positive correlation. In fact the interval time of a whole flow consists of the forward flow. In the right part, we can see that the feature "Backward Segment Size Average" is almost equals to "Backward Packet Length Mean". We can treat them as the same feature.

The redundancy of features is produced by the featrue extraction tool. If we use the whole data set to train our model, the redundant features on one hand may increase the computation and spend more time, on the other hand these redundant features cannot import any new informations into our model. So it is necessary to merge these redundant features. In this paper, we use clustering method to reach this target.

First the concept of correlation coefficient is reviewed.

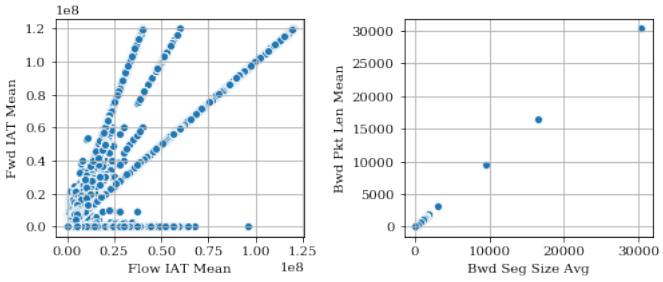

FIGURE 3. Two examples of correlated variables. The feature "Flow IAT Mean" and "Forward IAT Mean", "Backward Segment Size Average" and "Backward Packet Length Mean" are potentially related.

Definition 1 (Correlation Coefficient)**.** If the variances σ_X^2 and σ_Y^2 of two random variables X and Y exists and they satisfy that $\sigma_X^2 > 0$ and $\sigma_Y^2 > 0$, the correlation coefficient of X and Y can be denoted as

$$\mathrm{Corr}(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y} \tag{1}$$

where $\mathrm{Cov}(X,Y)$ is the covariation of X and Y. And

$$\mathrm{Corr}(X,Y) \begin{cases} > 0, & X \text{ and } Y \text{ are positive correlated.} \\ = 0, & X \text{ and } Y \text{ are not linear correlated.} \\ < 0, & X \text{ and } Y \text{ are negative correlated.} \end{cases}$$

Then we use the correlation coefficient to define the distance of any two features.

Definition 2 (The distance of two features)**.** The distance of two features f_i and f_j is the reciprocal of the absolute value of correlation coefficient of them, that is

$$d(f_i, f_j) = \frac{1}{|\mathrm{Corr}(f_i, f_j)|} \tag{2}$$

According to Equation 2, if two features have potential linear correlation, the distance between them is small. On the contrary, if the distance between two features are large, these two features are independent relatively. Here we consider both positive correlation and negative correlation are the same, so we use absolute value of the correlation coefficient. However, the more indepent two features are, the less absolute correlation coefficient of them is. This is the exact opposite of our general definition of distance. Hence we use reciprocal to solve this problem.

The clustering algorithm is described as Algorithm 1 and Algorithm 2. We calculate the distance between every feature and others. If the distance is less than a threshold δ, the feature is treated as linear related with the other and they belong to the same cluster. Otherwise, the feature will be put in a new cluster. The procedure "compare_and_join" at the 4th line of Algorithm 1 is complicated, so we list it as Algorithm 2.

Algorithm 1 Feature clustering based on Pearson correlation coefficient

Input:
Data set $D = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M)^{\mathrm{T}}$,
feature set $F = (f_1, f_2, \ldots, f_N)$,
distance threshold δ.
Output:
Cluster set $C = \{c_1, c_2, \ldots, c_K\}$

```
1: C ← ∅
2: for i = 1, 2, ..., n do
3:   if ∄c ∈ C s.t. f_i ∈ c then
4:     cluster c_join ← compare_and_join(D, f_i, C, δ)
5:     if ∃c_join which f_i can join then
6:       c_join.add(f_i)
7:     else
8:       Create a new cluster c'
9:       c'.add(f_i)
10:      C.add(c')
11:    end if
12:  end if
13: end for
14: return C
```

Note that in Algorithm 2. We directly use the absolute correlation coefficient instead of complying with the definition of Equation 2. This is done because we can limit the range of δ to $[0, 1]$.

After the clusters are generated, next step is to find the center of these clusters. This procedure is listed as Algorithm 3. We calculate the average distance of every feature between others in each cluster, then we pick the feature with minimum average distance with other features as the center of this cluster. If a cluster only have two features, the algorithm will select the first feature in the cluster as the center of it.

Algorithm 2 Compare new feature to all other features

Input:
Data set $D = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M)^{\mathrm{T}}$,
feature f_i,
distance threshold δ, currently existing clusters $C = \{c_1, c_2, \ldots, c_K\}$
Output:
A integer c_{join} which indicate the cluster which f_i can join in.

```
1: A vector including all maximum values of all existing cluster d_max(C) ← ∅
2: for k = 1, 2, ..., K do
3:   Distance vector for cluster c_k, i.e.d(c_k) ← ∅
4:   for j = 1, 2, ..., sizeof(c_k) do
5:     d = |Corr(D^(f_i), D^(f_j))|
6:     d(c_k).add(d)
7:   end for
8:   d_max(C).add(max d(c_k))
9: end for
10: Maximum distance in cluster c_k denoted as d_max = max d_max(C)
11: if d_max > δ then
12:   c_join = arg max_c d_max(C)
13:   return c_join
14: else
15:   return NULL
16: end if
```

Algorithm 3 Find the cluster center

Input:
Data set $D = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M)^{\mathrm{T}}$,
clusters $C = \{c_1, c_2, \ldots, c_K\}$ calculated in Algorithm 1
Output:
A feature list $F' = \{f'_1, f'_2, \ldots, f'_K\}$ whose features are the center of every cluster.

```
1: Feature list f' = ∅
2: for i = 1, 2, ..., K do
3:   if sizeof(c_i) = 1 then
4:     F'.add(f ∈ c_i)
5:   else if sizeof(c_i) = 2 then
6:     select a feature randomly
7:     F'.add(f ∈ c_i)
8:   else
9:     for each f ∈ c_i do
10:      d̄_f = 1/(|c_i| − 1) Σ d_{f,f'}
11:    end for
12:    f_c = arg min_f d_c
13:    F'.add(f_c)
14:  end if
15: end for
16: return F'
```

C. FEATURE RANKING BASED ON INFORMATION GAIN AND GAIN RATIO SIMULTANEOUSLY

After clustering all features, we generate a group of features which are pairwise independent. In order to further refine the feature space of our data, we choose those have better classification ability. The measure of feature classification ability is the information gain.

We select the top k best feature using information gain and information gain ratio simultaneously. The related definitions are listed as follows:

Definition 3 (Entropy of data set)**.** Entropy [25] is the measure of uncertainty of a random variable. In the context of our paper, the entropy of data set is defined as the entropy of labels. If the probability of label L picking value l_i equals to $P(L = l_i) = p_i$, the entropy of data set is denoted as

$$H(D) = -\sum_{i=0}^{N} p_i \log_2 p_i \tag{3}$$

Specially, if $p_i = 0$ then we define $0 \log 0 = 0$.

Definition 4 (Conditional entropy of data set with given feature)**.** The conditional entropy $H(D|f)$ is the uncertainty of data set D under the condition of known feature f, which is denoted as

$$H(D|f) = -\sum_{j} p(f_j) \sum_{i} p(l_i|f_j) \log_2 p(l_i|f_j) \tag{4}$$

where $p(f_j)$ is the probability when feature f takes f_j, $p(l_i|f_j)$ is the conditional probability when label L takes l_i under the condition of $f = f_j$.

The information gain indicates the degree to which the uncertainty of the information of the category Y is reduced by the information of the feature X.

Definition 5 (Information gain)**.** The information gain from feature f to data set D is defined as the difference between the entropy $H(D)$ of the set D and the conditional entropy $H(D|F)$ of D for a given feature f, i.e.

$$IG(D, f) = H(D) - H(D|f) \tag{5}$$

Obviously, for data set D the information gain is determined by its features. Different features have different information gain. If a feature has greater information gain, it has stronger ability to classify the data.

Definition 6 (Information gain ratio)**.** The problem of using information gain is that it tend to choose the feature which has larger value range. In order to eliminate this effect, the information gain ratio is introduced. It is defined as the information gain divided by entropy of the feature.

$$IGR(D, f) = \frac{IG(D, f)}{H(f)} \tag{6}$$

Now our feature ranking algorithm is introduced in Algorithm 4. We consider the information gain and information gain ratio simultaneously. First we calculate the information gain of all features which has been clustered by Algorithm 1 and 3. Then we calculate the average information gain and sort them by their information gain ratio. Finally we select top K features to compose new feature set. Here we set $K = 10$. In different anomaly detection scenarios, the value of K should be chosen on demand.

Algorithm 4 Feature ranking based on information gain

Input:

Data set D' with clustered features calculated in Algorithm 1 and Algorithm 3

Output:

A feature list $F'' = \{f''_1, f''_2, \ldots, f''_k\}$ whose features are top k after ranked.

1: Calculate the entropy $H(D')$ by its labels.
2: **for** $i = 1, 2, \ldots, K$ **do**
3: Calculate the conditional entropy $H(D'|f_i)$.
4: Calculate the information gain

$$IG_{f_i} = H(D') - H(D'|f_i)$$

5: Calculate the information gain ratio

$$IGR_{f_i} = IG_{f_i}/H(f_i)$$

6: **end for**
7: Calculate the average information gain $\bar{IG} = (\sum IG)/K$
8: Choose the features $F_{IG} = \{f|IG_f > \bar{IG}\}$
9: Sort the features according to IGR
10: **return** F''

D. SUMMARY OF PROPOSED ALGORITHMS

In this subsection, we summarize the proposed algorithm in a higher perspective. The complete algorithm is shown in Algorithm 5. Since we have selected all features we need to determine whether a flow is an attack. Furthermore we want to detect detailed attack types. We choose C4.5 decision tree [26] as our classification algorithm. C4.5 decision tree uses information gain ratio to classify data instances. The implementation of decision tree algorithms is beyond the scope of this article. However, our method in fact previously complete the feature selection process of decision tree, so the tree classifier can use our result and speed the training process.

E. ANALYSIS OF THE PROPOSED ALGORITHM

Our method is one kind of feature selection algorithm. It is different from dimensionality reduction algorithms like PCA [27]. PCA transforms the original data into a set of linearly independent representations of each dimension by linear transformation to extract the main linear components of the data, while feature selection methods do not transform the data. It just chooses the key features which can provide the most information about the data. The data transformed by PCA will lose their explanation of real meaning, while feature selection algorithms can keep the real meaning of

Algorithm 5 Complete algorithm of proposed scheme

Input:

Data set D of 20% randomly sampled, labels L, distance threshold δ, number of features to be seleced K.

Output:

A feature list $F'' = \{f''_1, f''_2, \ldots, f''_K\}$ and decision tree model M_T.

1: Drop all rows with missing values in D.
2: Drop all columns with zero-variance in D.
3: Encode all labels in L to numeric values.
4: Clustering all features using Algorithm 1. Get all clusters C.
5: Find centers in every clusters using Algorithm 3. Get F'.
6: Sort all features in F' using Algorithm 4. Get F''.
7: Split the data subset D'' with features in F'' into training set D''_{train} and test set D''_{test}.
8: Train the classification model M_T using decision tree algorithm with D''_{train}.
9: Test the classification effect using D''_{test}, and generate metrics Acc, Pre, Rec, F_1.
10: **return** F'', M_T.

every reserved features. Chandrashekar et.al. [28] pointed out that feature selection methods must not be compared with dimension reduction methods. Therefore we cannot compare the proposed scheme with PCA by experiments.

Meanwhile, our proposed method includes some characteristics of PCA. First, we calculate the covariance and correlation coefficient of all feature vectors, which is also used in calculation of PCA. Second, the features selected by our method are orthogonal and PCA do the same thing. Last but not least, every component generated by PCA have maximum variance, and our method contains the similar step. But in our method, the features with small variance are filtered.

V. EXPERIMENT RESULTS

In this section, the experiments and evaluation results are shown. First we describe the experiment setup. Then we list selected feature by our algorithm and explain why these feature are selected. Next, since the distance threshold δ is a changeable parameter, we describe how the numbers of clusters are impacted by different distance threshold. Further, we compare the training time and classification metrics including accuracy, precision, recall and F1-score between our method and the full original data set, as well as our method and benchmark method respectively. The benchmark method is Select-K-Best method using chi-square function, denoted as SKB-chi2 method. Finally we illustrate and explain the confusion matrix of our prediction results.

A. EXPERIMENT SETUP

All experiments are running on a quad-core Intel PC with 2.90GHz CPU and 16 GB of memory. Our algorithms are implemented in Python 3.7 and the benchmark method is implemented in a Python library scikit-learn [29]. We investigate the impact of distance thresholds and feature spaces on the number of clusters and classification results.

- We investigate the effect of different threshold δ on the number of clusters, then explain how to choose best threshold to improve the classification metrics best.
- We compare the training time and classification metrics of this method and the naive method, respectively. In the naive method, the data set is not processed, and all features are used for model traning and anomaly detection.
- We compare the training time and classification metrics of the features selected by the proposed algorithm and by SKB-chi2 respectively on the decision tree classifier under the condition of the same scale of features.

The related classification metrics including accuracy(Acc), precision(Pre), recall(Rec) and F1-score(F_1). Their definitions are listed as follows:

- **Accuracy**. In classification problems, we denote TP as true positve prediction, FP as false positive prediction, TN as true negative prediction and FN as false negative prediction. The accuracy is defined as the ratio of $TP + TN$ in all test instances, which is
$$\text{Acc} = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$
which means the all true instances classified in all test instances.
- **Precision**. The precision is defined as the ratio of TP in all positive instances classified by the model, which is
$$\text{Pre} = \frac{TP}{TP + FP} \tag{8}$$
which means the all true positive instances classified in all positve instances.
- **Recall**. The recall is defined as the ratio of TP in all real positive instances, which is
$$\text{Rec} = \frac{TP}{TP + FN} \tag{9}$$
- **F1-score**. The F1-score is defined as the harmonic average of precision and recall.
$$F_1 = \frac{2}{\frac{1}{\text{Pre}} + \frac{1}{\text{Rec}}} \tag{10}$$
which is a trade-off between the precision and recall.

B. SELECTED FEATURE SUBSET

Table 2 shows the features selected by our method and their descriptions. Notice that although we set $K = 10$ for our feature ranking algorithm, the selected features are not same in all split data set. Because randomicity exists

when we build the split data subset. Here we use the union of all selected feature subsets. From Table 2 we make the following observations:

- The sizes of packets are important. In the top 10 important features, there are 4 features about packet sizes. Considering the MTU of an IP network, a flow shouldn't contain too large packets. If the flow meter observes a flow contain large packets, the probability that this stream is an attack stream will be high.
- Another important feature type is the count of packets. In benign flows, the number of packets often small because current network services intend to use short connection to complete the interaction with each other, in order to not occupy the bandwidth resources of the network. However, the attackers intend to send large amounts of packets to exhaust the connection resources and prevent connection from normal users.
- Time-related features are also important. As we mentioned before, benign services intend to use short connection, while attackers may use long connection. A typical attack is to control the interval of any two flows which is a little shorter than the TCP waiting time. It prevents the TCP connection closing and finally exhaust the connection resources. In practise, time-related features should be considered with the count of packets together.

TABLE 2. Selected Features and Their Description

Feature Name	Description
Backward Packets/s	Number of backward packets per second
Forward Segment Size Min	Minimum segment size observed in the forward direction
Init Forward Win Byts	Number of bytes sent in initial window in the forward direction
Flow Packets/s	flow packets rate that is number of packets transferred per second
ACK Flag Count	Number of packets with ACK
Flow IAT Mean	Average time between two flows
Backward IAT Max	Maximum time between two packets sent in the backward direction
Idle Mean	Mean time a flow was idle before becoming active
Forward Packet Length Min	Minimum size of packet in forward direction
Flow Duration	Flow duration
Backward Packet Length Mean	Mean size of packet in backward direction
Packet Length Min	Minimum length of a flow
Forward Packets/s	Number of forward packets per second
Forward IAT Total	Totalal time between two packets sent in the forward direction

TABLE 3. Average number of clusters under different distance threshold

δ	0.7	0.75	0.8	0.85	0.9	0.95
$\bar{N}$	26	29	30	32	36	42

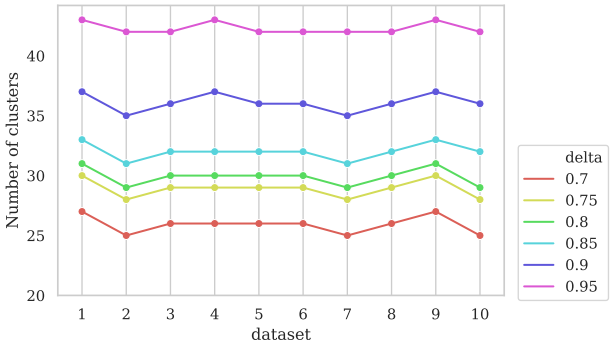

FIGURE 4. Number of clusters in 10 split data set under the effect of different distance threshould δ.

C. INVESTIGATE THE EFFECT OF THE DISTANCE THRESHOLD ON THE CLUSTER NUMBERS

In our algorithm, an important parameter is the distance threshold δ. It takes effects on the number of clusters generated in Algorithm 1. Figure 4 shows the result of number of generated clusters from 10 split data set files under different threshold δ. We set δ from 0.7 to 0.95 with interval 0.05 between any two values. The average number of clusters are listed in Table 3. Here a trade-off exists about the number of clusters and the number of features in every cluster. If δ is set too small, the number of clusters will be small too, however many features with weak correlation will be merged in the same cluster and it will finally impact subsequent feature ranking and decision tree learning procedures. Meanwhile, a larger δ will produce too many clusters which contains only one feature, because the condition is too stricted to merge features. In our practice, setting δ between 0.8 and 0.9 is a better choicce.

D. INVESTIGATE THE EFFECT OF SCALE OF FEATURES ON THE TRAINING TIME AND METRICS

In this subsection we compare the training time of decision tree classifier and the metrics of classification effects between our method under different distance threshold δ and the full original data set.

1) Training Time

The training time in our method is the sum of running time of all algorithms mentioned before. The training time under different distance threshold δ is shown in Figure 5 and Table 4. When we use the full original data set to train a decision tree classifier and then prune the tree to 10 decision nodes too, the training time is near 300 seconds averagely. Using our method, the training time decreases near 100 seconds

averagely. Meanwhile this figure shows that the average training time increases slowly with δ increasing.

TABLE 4. Training Time comparing to full data set and chi-square method

dataset	Training Time(s)
$\delta = 0.7$	264.70
$\delta = 0.75$	266.21
$\delta = 0.8$	265.12
$\delta = 0.85$	267.21
$\delta = 0.9$	269.39
$\delta = 0.95$	269.89
Full data set	308.14
SKB-Chi2	203.36

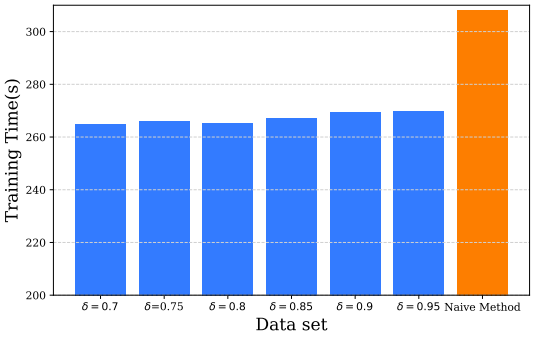

FIGURE 5. Training Time comparison with full original data set

2) Metrics

The four metrics of our methods under different δ is shown in Figure , along with the metrics of the model trained on full original data set. The result is shown in Figure 6 and Table 5. The average accuracy of our method is 97.94%, which is a little higher than the accuracy on full data set. The average precision, recall and F1-score are 79.46%, 80.73% and 79.33% respectively. They are all higher than the results on full data set.

The result proved that the feature selected by our method can help improving performance of decision tree classifier comparing with using tree classifier directly on the full original data set.

E. INVESTIGATE THE EFFECT OF DIFFERENT FEATURE SPACE AT THE SAME SCALE ON TRAINING TIME AND METRICS

1) Training Time

In this experiment, we compare the training time and metrics of our method and SKB-chi2 method in the same size of feature set, in other words the number of features are both set to 10 in these two methods.

The training time comparison is shown in Figure 7 and also listed in Table 4. The table and figure shows that our average training time is longer than the chi-square method

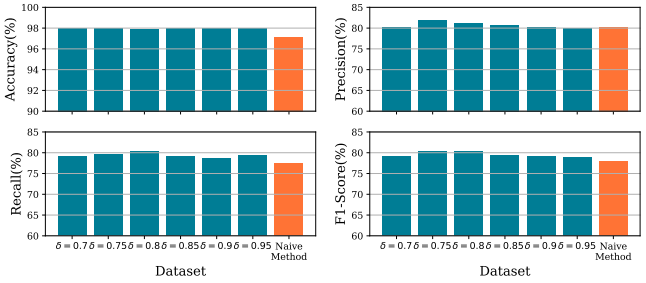

FIGURE 6. The metrics including accuracy, precision, recall and F1-score of the data subset with different δ and the full data set.

performs. The chi-square method selects n features with the highest values for the test chi-squared statistic from input data X. However, it only considers whether every single feature is correlated with the class label instead of considering the correlation between these features, which makes the training time lower than our methods. As a result, the selected feature subset may contain features with inner relationship, which may bring the redundant information in the model. While our method can effectively eliminate the redundant information.

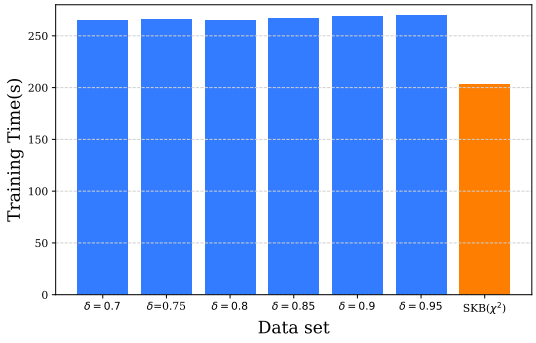

FIGURE 7. Training Time comparison with Select-K-Best method based on χ^2 function.

2) Metrics

The classification metrics comparison is shown in Figure8 and Table 5. The figure shows that the metrics of our method are all better than the chi-square funtion. Although the training time is longer, an advantage of our method is that it consider more correlations between any two features comparing with only considering the relationship between any single feature and the class label. Therefore, the experimental results show that our method can better determine the statistical law of traffic when a network attack occurs.

F. CONFUSION MATRIX

Finally we analyze the confusion matrix of our method. The confusion matrix is shown in Tabel 6. We add the

TABLE 5. The metrics of different δ and full data set.

dataset	Accuracy(%)	F1-Score(%)	Precision(%)	Recall(%)
$\delta = 0.7$	97.9604	79.0699	80.1735	79.0867
$\delta = 0.75$	97.9297	80.2657	81.9095	79.6229
$\delta = 0.8$	97.9104	80.2791	81.1272	80.3653
$\delta = 0.85$	97.9558	79.3655	80.7983	79.0457
$\delta = 0.9$	97.9557	78.9994	80.3196	78.6552
$\delta = 0.95$	97.9489	78.8134	79.9506	79.2532
Full data set	97.1315	78.0325	80.2580	77.3223
SKB-Chi2	94.2846	72.1371	68.7105	69.1960

TABLE 6. Confusion Matrix of classification results

Labels	Benign	Bot	Web[1]	XSS[2]	HOIC[3]	LOIC-UDP[4]	LOIC-HTTP[5]	Slow-HTTPTest[6]	Hulk[7]	Infilter-ation	Slow-loris[8]	SSH[9]	Golden-Eye[10]	SQL[11]	FTP[12]
Benign	5320195	1135	65	22	10	0	185	33	17	8	76	3	34205	13	11
Bot	626	113842	0	7	0	0	4	0	0	0	0	0	1	0	0
Web	66	0	161	0	0	0	2	0	0	0	0	0	1	10	0
XSS	20	9	0	57	0	0	0	0	0	0	0	0	0	4	0
HOIC	3	0	0	0	274397	0	0	0	0	0	0	0	0	0	0
LOIC-UDP	2	0	0	0	0	537	151	0	0	0	0	0	0	0	0
LOIC-HTTP	164	7	0	0	0	160	230148	0	0	0	0	0	1	0	0
SlowHTTPTest	6	0	0	0	0	0	0	16567	2	0	25	0	0	0	0
Hulk	11	0	0	0	0	0	0	3	184755	0	0	0	1	0	0
Infilteration	0	0	0	0	0	0	0	0	0	29100	0	26860	0	0	0
Slowloris	102	0	0	0	0	0	0	18	1	0	4279	0	0	0	0
SSH	0	0	0	0	0	0	0	0	0	8747	0	68593	0	0	0
GoldenEye	58771	8	0	0	0	0	1	1	0	3	0	1	5408	1	0
SQL	17	1	2	0	0	0	0	0	0	0	0	0	0	10	0
FTP	13	0	1	0	0	0	0	0	0	1	0	9	0	0	75016

[1] BruteForce-Web [2] BruteForce-XSS [3] DOS attacks-HOIC [4] DOS attacks-LOIC-UDP [5] DOS attacks-LOIC-HTTP
[6] DOS attacks-SlowHTTPTest [7] DOS attacks-Hulk [8] DOS attacks-Slowloris [9] SSH-BruteForce [10] DOS attacks-GoldenEye
[11] SQL Injection [12] FTP-BruteForce

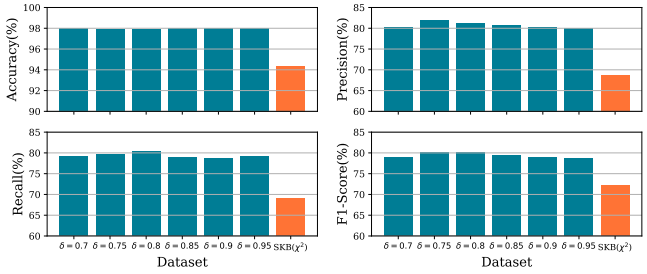

FIGURE 8. The metrics of the data subset with different δ and the SelectKBest method based on χ^2 function.

confusion matirx of all split data set. The matrix shows that our method can mainly classify the data into correct class label. However, there are still many false positive and false negative instances. The most severe miss happens on the label "DOS attacks-GoldenEye" and "Infilteration". The flow patterns of these two situation performs similar with benign flow pattern so that our method could not discriminate them properly. The flow patterns of these two situation should be further study and new method should be proposed to detect them.

VI. CONCLUSION

Traditional network anomaly detection methods need to consider a large number of traffic features. But this will cause a dimensional disaster problem, which will seriously reduce the efficiency and accuracy of the anomaly detection method. In this paper, we propose a novel efficient network anomaly detection scheme, which uses feature selection methods to reduce the dimensions of traffic data and improves the efficiency and accuracy of detection. The most important part of this method is the feature selection algorithm proposed in this paper. The algorithm contains two parts: feature clustering and feature ranking. The feature clustering algorithm uses correlation coefficients to merge features with correlated features into the same cluster. The feature ranking algorithm uses both information gain and gain ratio to select the most important features. Based on this, we use the C4.5 decision tree as a classifier to construct a network traffic anomaly detection model.

The experimental results show that on the CIC-IDS-2018 dataset, our proposed method can reduce the training time by about 100 seconds and improve the classification index by at least 1%. Compared with the SKB-Chi2 method, with

the same number of features, our method can improve the classification index by at least 3%.

In the future, we will continue to study the analysis of real-time network traffic data and the real-time detection model of network anomalies to improve the practicality of detection schemes.

REFERENCES

[1] GitHub triumphant over its 'largest ever' cyber pummeling. [Online]. Available: https://fortune.com/2015/04/03/github-ddos-china/
[2] Read Dyn's Statement on the 10/21/2016 DNS DDoS Attack. [Online]. Available: https://dyn.com/blog/dyn-statement-on-10212016-ddos-attack/
[3] WannaCry ransomware attack. [Online]. Available: https://en.wikipedia.org/wiki/WannaCry_ransomware_attack
[4] M. Wang, Y. Lu, and J. Qin, "A Dynamic MLP-Based DDoS Attack Detection Method Using Feature Selection and Feedback," Computers & Security, to be published. DOI: 10.1016/j.cose.2019.101645.
[5] T.S. Urmila and R. Balasubramanian, "A novel framework for intrusion detection using distributed collaboration detection scheme in packet header data," International Journal of Computer Networks and Communications, vol. 9, no. 4, pp. 97–112, 2017.
[6] M.S. Al-Hawawreh, "SYN Flood Attack Detection in Cloud Environment Based on TCP/IP Header Statistical Features," in Proceedings of International Conference on Information Technology, pp. 236–243, 2017.
[7] H.M. Anwer, M. Farouk and A. Ayman, "A Framework for Efficient Network Anomaly Intrusion Detection with Features Selection," in Proceedings of 9th International Conference on Information and Communication Systems, pp. 157–162, 2018.
[8] "UNSW-NB15 dataset". [Online]. Available: https://cloudstor.aarnet.edu.au/plus/index.php/s/2DhnLGDdEECo4ys?path=%2F
[9] G. Fernandes, J. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi and M.L. Proença, "A comprehensive survey on network anomaly detection," Telecommunication Systems, vol. 70, no. 3, pp. 447–489, 2019
[10] G. Fernandes, E.H.M. Pena, L.F. Carvalho, J. Rodrigues, M.L. Proença, "Statistical, Forecasting and Metaheuristic Techniques for Network Anomaly Detection," in Proceedings of the ACM Symposium on Applied Computing, pp. 701–707, 2015.
[11] W.M. Alsharafi, M.N. Omar, N.A. Al-Majmar and Y. Fazea, "Normal Profile Updating Method for Enhanced Packet Header Anomaly Detection," Emerging Trends in Intelligent Computing and Informatics, pp. 734–737, 2019.
[12] K. Bajaj and A. Arora, "Dimension Reduction in Intrusion Detection Features Using Discriminative Machine Learning Approach," International Journal of Computer Science Issues, vol. 10, no. 4, pp. 324–329, 2013.
[13] K. Kumar and J. S. Batth, "Network Intrusion Detection with Feature Selection Techniques using Machine-Learning Algorithms," International Journal of Computer Applications, vol. 150, no. 12, pp. 1–13, 2016.
[14] Y. Wahba, E. ElSalamouny and G. ElTaweel, "Improving the Performance of Multi-class Intrusion Detection Systems using Feature Reduction," International Journal of Computer Science Issues, vol. 12, no. 3, pp. 255–262, 2015.
[15] P. Kaur, P. Chaudhary, A. Bijalwan and A. Awasthi, "Network Classification Using Multiclass Classifier," in Proceedings of International Conference on Advances in Computing and Data Sciences, pp. 208-217, 2018.
[16] J. R. Beulah and D. ShaliniPunithavathani, "Simple Hybrid Feature Selection (SHFS) for Enhancing Network IntrusionDetection with NSL-KDD Dataset," International Journal of Applied Engineering Research, vol. 10, no. 19, pp. 40498–40505, 2015.
[17] A. Ferriyan, A.H. Thamrin, K. Takeda and J. Murai, "Feature selection using genetic algorithm to improve classification in network intrusion detection system," in Proceedings of International Electronics Symposium on Knowledge Creation and Intelligent Computing, pp. 46–49, 2017.
[18] H. Shi, H. Li, D. Zhang, C. Cheng and W. Wu, "Efficient and robust feature extraction and selection for traffic classification," Computer Networks, vol. 119, pp. 1–16, 2017.
[19] H. Shi, H. Li, D. Zhang, C. Cheng and X. Cao, "An efficient feature generation approach based on deep learning and feature selection techniques for traffic classification," Computer Networks, vol. 132, pp. 81–98, 2018.
[20] Y. Dong, J. Zhao and J. Jin, "Novel feature selection and classification of Internet video traffic based on a hierarchical scheme," Computer Networks, vol. 119, pp. 102–111, 2017.
[21] U. Cavusoglu, "A New Hybrid Approach for Intrusion Detection Using Machine Learning Methods," Applied Intelligence, vol. 49, no. 7, pp. 2735–2761, 2019.
[22] M. Sofiane and T. Mohamed, "Feature selection algorithms in intrusion detection system: A survey," KSII Transactions on Internet and Information Systems, vol. 12, no. 10, pp. 5079-5099, 2018.
[23] I. Sharafaldin, A. H. Lashkari and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in Proceedings of International Conference on Information Systems Security and Privacy (ICISSP), pp. 108–116, 2018.
[24] CICFlowMeter. [Online]. Available: https://www.unb.ca/cic/research/applications.html#CICFlowMeter
[25] C.E. Shannon, "A Mathematical Theory of Communication," Bell System Technical Journal, vol. 27, no. 3, pp. 379–423, 1948.
[26] J.R. Quinlan, "C4.5: programs for machine learning," Elsevier, 2014.
[27] S. Wold, K. Esbensen and P. Geladi, "Principal component analysis," Chemometrics and Intelligent Laboratory Systems, vol. 2, no. 1–3, pp. 37–52, 1987.
[28] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," Computers and Electrical Engineering, vol. 40, pp. 16–28, 2014.
[29] scikit-learn: machine learning in Python. [Online]. Available: https://scikit-learn.org/

JIEWEN MAO received the B.S. degree from the Department of Computer Science and Technology at East China Normal University, Shanghai, China, in 2014. He is currently pursuing his Ph.D. degree with the School of Computer Science and Technology at East China Normal University, Shanghai, China. His current research interests are in the area of anomaly detection of network traffic flow, including the Internet and IoT. He is also interested in machine learning techniques.

YONGQUAN HU received the B.S. degree from the Department of Engineering of Internet of Things at Zhejiang University of Technology, Zhejiang Province, China, in 2019. He is currently pursuing his master degree with the Department of Computer Science and Technology, East China Normal University, Shanghai, China. His current research interests are in the area of cloud computing, edge computing and machine learning techniques.

DONG JIANG received the B.S. degree from the Department of Electronic Information Science and Technology, Shandong University of Science and Technology, Qindao, Shandong Province, China, in 2017. He received the master degree from the Department of Computer Science and Technology, East China Normal University, Shanghai, China, in 2019. His current research interests are in computer networks.

TONGQUAN WEI(M'11-SM'19) received his Ph.D. degree in Electrical Engineering from Michigan Technological University in 2009. He is currently an Associate Professor in the Department of Computer Science and Technology at East China Normal University. His research interests are in the areas of internet of things (IoT), edge computing, cloud computing, and design automation of intelligent systems and cyber physical systems (CPS). He has published numerous papers in these areas, most of which are published in premium conferences and journals. He serves as a Regional Editor for Journal of Circuits, Systems, and Computers since 2012. He also served as the Guest Editor of the IEEE TII SS on Building Automation, Smart Homes, and Communities, the ACM TESC SS on Embedded Systems for Energy-Efficient, Reliable, and Secure Smart Homes, and the ACM TCPS SS on Human-Interaction-Aware Data Analytics for Cyber-Physical Systems. He is a senior member of the IEEE.

FUKE SHEN received his Ph.D. degree in Department of Computer Science and Technology of East China Normal University in 2011. He is currently an Professor in the Department of Computer Science and Technology at East China Normal University, and director of Information Center at East China Normal University. His research interests are in the areas of computer networks communication, next generation network architecture, network protocols and their implementation, network traffic monitoring and management, and digital campus network.