A Hierarchical Feature Selection Method for Network Anomaly Detection

Jiewen Mao, Dong Jiang, Yongquan Hu, Tongquan Wei, Fuke Shen
School of Computer Science and Technology, East China Normal University

***Abstract*—Abnormal detection of network traffic is still an important means of preventing network attacks. In the anomaly detection process, researchers need to deal with a large number of features in network traffic. In order to determine whether the network traffic is the most essential feature of the attack, in this paper we propose a hierarchical feature selection method. The method selects the essential features of network traffic through feature clustering method based on correlation coefficient and feature ranking method based on information gain, then it classifies network traffic by decision tree classifier. Experiments show that our method reduces the number of features, and shortens training time comparing with full feature set. By comparison with chi-square feature selection method, our method improves the metrics including accuracy, precision, recall and F1-score.**

***Index Terms*—feature selection, abnormal detection, feature clustering, correlation coefficient, information gain, decistion tree**

I. Introduction

WHILE the Internet of Things, 5G technology, the improvement of network bandwidth and new network security testing tools bring convenience to people, malicious traffic and network attacks are still serious threats to Internet users and servers. 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].

Traditional anomaly detection methods often only use the header information of the network packets. Wang et.al. [4] proposed a dynamic MLP-based attack detection method. The feature selection process of this paper uses backward selection filter method. It deletes the features which make the accuracy of MLP decreasing more than threshold and the remainings are the selected features. However the method in [4] only uses the header information including source IP, TCP flag, source port, destination port, etc.. However, diverse types of network anomalies cannot be distinguish effectively only by packet header information. Even in some types of attacks 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 problem, this paper takes the statistical information extracted by aggregated flows besides the header information. Different types of anomalies have different statistical patterns. For example, (D)DoS attacks may have larger counts of packets but stable flow duration, while in brute-force attack the packet counts will be smaller but the curve of flow duration fluctuates drasticly.

Another problem of statistical detection methods are dimensional explosion. A flow may have more than 30 features in the header alone. Considering the statistical characteristics of all packets in flows, number of features in a network data set will grow rapidly. There may be linear relationships or other associations between these features. If we take all features into consideration, on one hand the efficient of learning and modeling algorithms will decrease, on the other hand it is hard to find the intrinsic cause that can determine whether a flow is an attack.

This paper proposes a hierarchical feature selection method, including three steps of data preprocessing, feature clustering and feature ranking. First, we preprocess the network traffic data. This procedure includes removing features that are clearly not available for statistical analysis, filling or dropping missing data, and encoding labels to numerical values. Second, we propose a feature clustering algorithm based on Pearson correlation coefficient to cluster the features with strong correlation and then select the cluster center. The third step will continue the second step, a feature rank algorithm based on information gain and information gain ratio is used to further filter the features. Finally, we use the decision tree (DT) as a classifier and conduct experiments among our proposed method, the features selected by chi-square testing algorithm and the full feature set. We compare them by the training time and training metrics including accuracy, precision, recall and F1-score.

The contributions of our paper can be summarized as follow:

1) This paper proposes a feature clustering method based on Pearson correlation coefficient, which uses correlation coefficients to define distances and aggregates the features with similar distances, and finds the cluster center as the representative feature of the cluster.
2) This paper proposes a feature ranking algorithm based on information gain, which sorts the features selected before and choose top k features as the final result of selector.
3) This paper then analyzes the selected feature subset and explains why they can determine whether a flow is normal or attack.

4) This paper uses decision tree as classifier to compare selected feature subset using proposed method, chi-square selection method, and the complete feature set on the aspect of training time, accuracy, precision, recall and F1-score.

The remaining part of the paper is organized as follows: Section II describes related works. Section III describe the formalization of feature selection problem. Section IV introduces our hierarchical feature selection method. Section V shows the experiment results and then the results is discussed in Section VI. Finally Section VII concludes this paper and indicates future works.

II. Related Works

Different approaches have been proposed to apply to feature selection to improve the performance of feature selection. H. C. Law et al. propose an expectation-maximization (EM) algorithm to estimate the importance of different features and the best number of components for Gaussian-mixture clustering[5]. EM can avoid running EM many times with different numbers of components and different feature subsets, and can achieve better performance than using all the available features for clustering. Yang et al.[6] present a modified Network Maximal Correlation (NMC) model as a measure to capture correlation relationships between a characteristic variable and a label variable. The results show the method can obtain an optimal subset of features with faster speed, maximum correlation and minimal redundancy through numerical simulation.

In addition, various of new feature selection approaches have been presented in the past years. Wu et al. [7]propose a new feature selection algorithm based on features unit (FU), which uses entropy of information to obtain features units and sort them to selected the appropriate one. The results in the UCI datasets show that the FU performs better than MIFS-U and mRMR on the whole. Yassine et al.[8] propose a new hybrid filter-wrapper algorithm of feature selection based on pairwise feature selection, which benefits from the speed up and the ease of use of filters and the good performance of wrappers. The results indicate that the selected subset of features by the proposed approach has a good classification performance. Yang et al. [9] propose a novel unsupervised feature selection method where constructing similarity matrix and performing feature selection are together incorporated into a coherent model. The results show the proposed approach has better performance to solve the objective function and extensive experiments on face images and benchmark datasets. Ke et al. [10] propose a redundant window-based optimal feature subset discover algorithm for feature selection, which use the growth algorithm to discover the relevant features and use the shrink algorithm to eliminate the redundant ones. The results show that the method has a good performance in terms of accuracy and scalability, and improves the execution efficiency of feature selection and traffic classification.

Liu et al. [11] propose a differentially private ensemble feature selection algorithm based on output perturbation. The results also demonstrate the high performance under certain privacy preservation degree of the method. Ferriyan et al. [12] propose a new feature selections using Genetic Algorithm to find the optimal features from NSL-KDD Cup 99 dataset, 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. Han et al. [13] propose a novel unsupervised feature selection method via the graph matrix learning and the low-dimensional space learning to obtain their individually optimized result. The results on real datasets verified that the method achieved the best classification performance compared to the state-of-the-art feature selection methods.

Feature selection also have been broadly used in processing traffic data. Shi et al. [14] propose 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 PCABFS. 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[15] 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.

Moreover, there are different feature selection methods aim to process different kinds of data. Dong et al.[16] propose a fine grained classification scheme which based on a hierarchical kNN classifier for network video traffic. The results show that the proposed method outperforms existing methods applying commonly used flow statistical features. Taskin et al. [17] presented a novel feature-selection method based on High Dimensional Model Representation (HDMR) to analyze and test in classification of hyperspectral images. The results show that the proposed approach can be used as a fast and efficient feature-selection method yielding very competitive results compared to the state-of-art feature-selection methods. Valadi et al.[18] propose a new modification of attribute selection with multiple label which can be advantageously used for handling high dimensional multi-level datasets. The results show the proposed approach reduces complexity and computational run time.

III. Problem Definition

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.

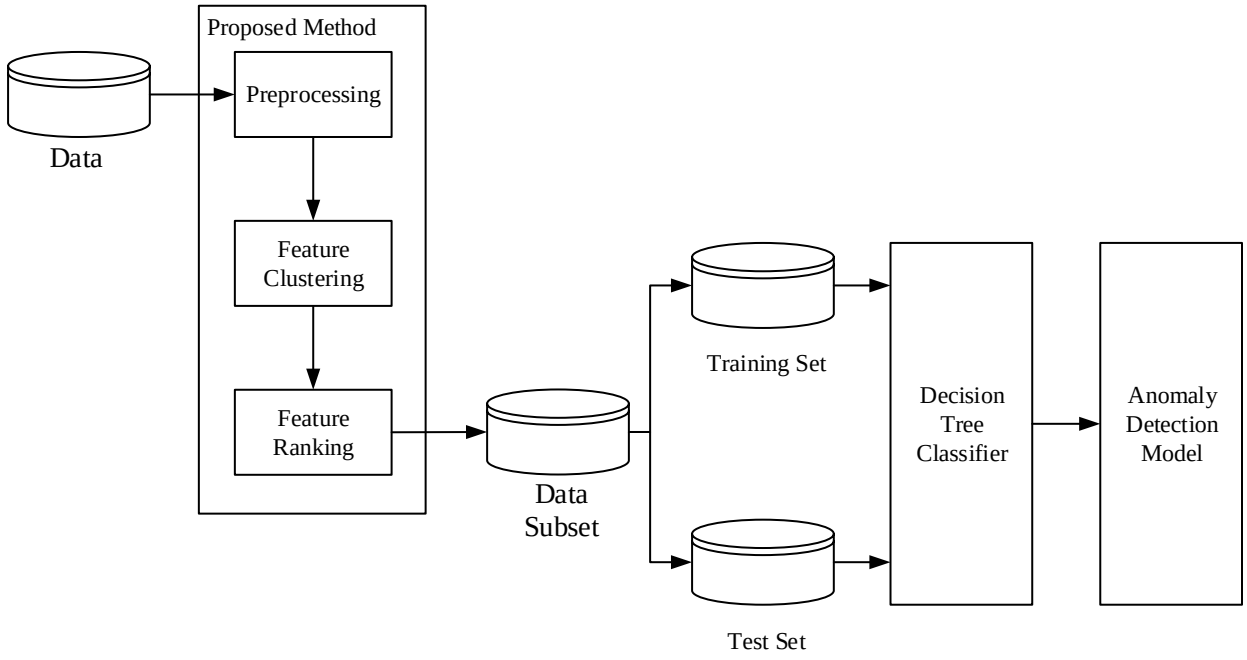

Fig. 1. Detection Framework

The feature selection problem[19] 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. $S = \{s_1, s_2, \ldots, s_l\}$, $l = 2^n - 1$ is the search space, which contains all subsets can be constructed from F with $s_i = \{f_j, f_k, \ldots, f_l\}$, $(1 \leqslant j \neq k \neq l \leqslant n)$. 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.

IV. Hierarchical Feature Selection Method

A. Overview

The system framework of this paper is shown as Fig. 1. The proposed method is the first block in the figure, and it contains three steps or layers: The first step is preprocessing, which cleans data set and makes it to a better form to be processed in next two steps. The second step is feature clustering, which puts all features with potential linear correlation into the same cluster, then all cluster centers are chosen as the representative of every clusters. The third step is feature ranking, which further rank all features selected in step 2 and choose the top k features as the final feature set.

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 our detection model is generated.

In following subsections the details and algorithms of proposed methods are described.

B. Data Set and Preprocessing

The data set being studied is CIC-IDS-2018[20]. 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[21]. These 76 features are shown in Table I.

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:

TABLE I
76 FEATURES OF CIC-IDS-2018 DATA SET

Feature Name	Description	Feature Name	Description
fl dur	Flow duration	pkt len max	Maximum length of a flow
tot fw pk	Total packets in the forward direction	pkt len avg	Mean length of a flow
tot bw pk	Total packets in the backward direction	pkt len std	Standard deviation length of a flow
tot l fw pkt	Total size of packet in forward direction	pkt len va	Minimum inter-arrival time of packet
fw pkt l max	Maximum size of packet in forward direction	fin cnt	Number of packets with FIN
fw pkt l min	Minimum size of packet in forward direction	syn cnt	Number of packets with SYN
fw pkt l avg	Average size of packet in forward direction	rst cnt	Number of packets with RST
fw pkt l std	Standard deviation size of packet in forward direction	pst cnt	Number of packets with PUSH
Bw pkt l max	Maximum size of packet in backward direction	ack cnt	Number of packets with ACK
Bw pkt l min	Minimum size of packet in backward direction	urg cnt	Number of packets with URG
Bw pkt l avg	Mean size of packet in backward direction	cwe cnt	Number of packets with CWE
Bw pkt l std	Standard deviation size of packet in backward direction	ece cnt	Number of packets with ECE
fl byt s	flow byte rate that is number of packets transferred per second	down up ratio	Download and upload ratio
fl pkt s	flow packets rate that is number of packets transferred per second	pkt size avg	Average size of packet
fl iat avg	Average time between two flows	fw seg avg	Average size observed in the forward direction
fl iat std	Standard deviation time two flows	bw seg avg	Average size observed in the backward direction
fl iat max	Maximum time between two flows	fw byt blk avg	Average number of bytes bulk rate in the forward direction
fl iat min	Minimum time between two flows	fw pkt blk avg	Average number of packets bulk rate in the forward direction
fw iat tot	Total time between two packets sent in the forward direction	fw blk rate avg	Average number of bulk rate in the forward direction
fw iat avg	Mean time between two packets sent in the forward direction	bw byt blk avg	Average number of bytes bulk rate in the backward direction
fw iat std	Standard deviation time between two packets sent in the forward direction	bw pkt blk avg	Average number of packets bulk rate in the backward direction
fw iat max	Maximum time between two packets sent in the forward direction	bw blk rate avg	Average number of bulk rate in the backward direction
fw iat min	Minimum time between two packets sent in the forward direction	subfl fw pk	The average number of packets in a sub flow in the forward direction
bw iat tot	Total time between two packets sent in the backward direction	subfl fw byt	The average number of bytes in a sub flow in the forward direction
bw iat avg	Mean time between two packets sent in the backward direction	subfl bw pkt	The average number of packets in a sub flow in the backward direction
bw iat std	Standard deviation time between two packets sent in the backward direction	subfl bw byt	The average number of bytes in a sub flow in the backward direction
bw iat max	Maximum time between two packets sent in the backward direction	fw win byt	Number of bytes sent in initial window in the forward direction
bw iat min	Minimum time between two packets sent in the backward direction	bw win byt	Number of bytes sent in initial window in the backward direction
fw psh flag	Number of times the PSH flag was set in packets travelling in the forward direction (0 for UDP)	Fw act pkt	Number of packets with at least 1 byte of TCP data payload in the forward direction
bw psh flag	Number of times the PSH flag was set in packets travelling in the backward direction (0 for UDP)	fw seg min	Minimum segment size observed in the forward direction
fw urg flag	Number of times the URG flag was set in packets travelling in the forward direction (0 for UDP)	atv avg	Mean time a flow was active before becoming idle
bw urg flag	Number of times the URG flag was set in packets travelling in the backward direction (0 for UDP)	atv std	Standard deviation time a flow was active before becoming idle
fw hdr len	Total bytes used for headers in the forward direction	atv max	Maximum time a flow was active before becoming idle
bw hdr len	Total bytes used for headers in the forward direction	atv min	Minimum time a flow was active before becoming idle
fw pkt s	Number of forward packets per second	idl avg	Mean time a flow was idle before becoming active
bw pkt s	Number of backward packets per second	idl std	Standard deviation time a flow was idle before becoming active
pkt len min	Minimum length of a flow	idl max	Maximum time a flow was idle before becoming active
		idl min	Minimum time a flow was idle before becoming active

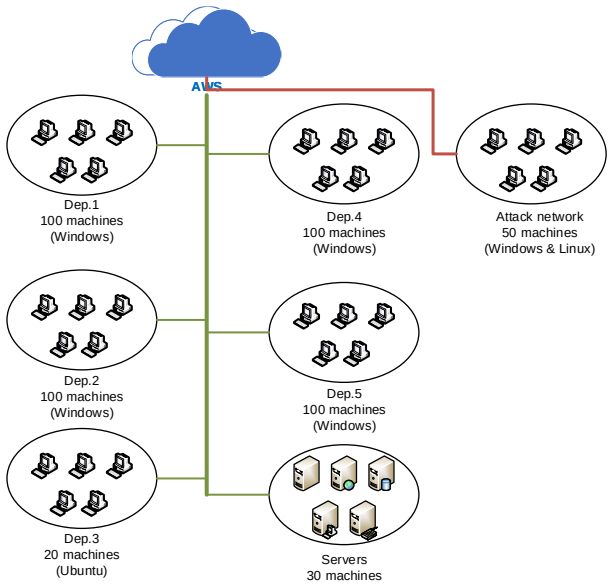

Fig. 2. Simulated network topology on AWS computing platform

1) Data cleaning: The target of this paper is to find the decisive statistical features which can determine whether a network flow is an attack. 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 to 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 Pkts/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 a physical quantity used to describe the degree of discreteness of a variable. In a data set, if the variance of a feature is zero, it means 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: The type of elements in column "Label" is text. It is not a good type because it may decrease the efficiency when process it in our algorithm. We encoder them to numeric code.

C. Layer 1: Feature Clustering Algorithm based on Pearson Correlation Coefficient

Many features of the original data set are derived from others. According to our observation, there are linear correlations between many features. The most important step of our method is to merge these linear related features via clustering method. First the concept of correlation coefficient is reviewed.

Definition 1 (Correlation Coefficient): The correlation coefficient $Corr(X, Y)$ between two variables X and Y is defined by their respective standard deviation and their co-variation. That is

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

where $Cov(X, Y)$ is the co-variation of X and Y, and σ_X and σ_Y are the standard deviation of X and Y respectively.

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 is the reciprocal of the absolute value of correlation coefficient, that is

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

If two features have potential linear relationship, the distance between them is small. On the contrary, if the distance between two features are large, these two features are independent relatively.

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 consist 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

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: $C \leftarrow \emptyset$
2: **for** $i = 1, 2, \ldots, n$ **do**
3: **if** $\nexists c \in C$ s.t. $f_i \in c$ **then**
4: cluster $c_{join} \leftarrow$ compare_and_join(D, f_i, C, δ)
5: **if** $\exists c_{join}$ 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

Algorithm 2 Compare new feature to all features of all existing cluster

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.

A vector including all maximum values of all existing cluster $\boldsymbol{d}_{\max}(C) \leftarrow \emptyset$
for $k = 1, 2, \ldots, K$ **do**
  Distance vector for cluster c_k, i.e.$\boldsymbol{d}(c_k) \leftarrow \emptyset$
  for $j = 1, 2, \ldots,$ sizeof(c_k) **do**
    $d = \mathrm{Corr}(D^{(f_i)}, D^{(f_j)})$
    $\boldsymbol{d}(c_k)$.add(d)
  end for
  $\boldsymbol{d}_{\max}(C)$.add$(\max \boldsymbol{d}(c_k))$
end for
Maximum distance in cluster c_k denoted as $d_{\max} = \max \boldsymbol{d}_{\max}(C)$
if $d_{\max} > \delta$ **then**
  $c_{join} = \arg\max_c \boldsymbol{d}_{\max}(C)$
  return c_{join}
else
  return NULL
end if

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 as the center of this cluster. If a cluster only have two features, the algorithm will select a feature as the center randomly.

D. Layer 2: Feature Ranking Algorithm based on Information Gain

Based on the result of layer 1, we select the top k best feature using information gain and information gain ratio simultaneously. The related definitions are list as follow:

Definition 3 (Entropy of data set): The entropy of data set is defined as the entropy of labels.

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

where L is the labels of data set.

Definition 4 (Conditional entropy of data set with given feature): When the information of a feature is introduced, the entropy of the labels will change. That means the unpredictability of the data set decreases when new information is introduced.

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

Definition 5 (Information gain): The information gain is defined as the difference between entropy of data set and conditional entropy with given feature.

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

Definition 6 (Information gain ratio): Using information gain may 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.

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

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.

Feature list $f' = \emptyset$
for $i = 1, 2, \ldots, K$ **do**
  if sizeof(c_i) = 1 **then**
    F'.add$(f \in c_i)$
  else if sizeof(c_i) = 2 **then**
    select a feature randomly
    F'.add$(f \in c_i)$
  else
    for each $f \in c_i$ **do**
      $\bar{d}_f = \frac{1}{|c_i|-1} \sum d_{f,f'}$
    end for
    $f_c = \arg\min_f d_c$
    F'.add(f_c)
  end if
end for
return F'

E. Decision Tree Classifier

V. Evaluations

The evaluations consists of two parts: the first part is comparison of training time, and the second part is comparison of metrics including accuracy, precision, recall and f1-score. All comparisons are conducted with the features selected by our method and by chi-square method, and the full feature set.

Algorithm 4 Feature ranking

Input:

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

Output:

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

Calculate the entropy $H(D')$ by its labels.
for $i = 1, 2, \ldots, K$ **do**
  Calculate the conditional entropy $H(D'|f_i)$.
  Calculate the information gain $IG_{f_i} = H(D') - H(D'|f_i)$
  Calculate the information gain ratio $IGR_{f_i} = \frac{IG_{f_i}}{H(f_i)}$
end for
Calculate the average information gain $\bar{IG} = \frac{1}{K}\sum IG$
Choose the features $F_{IG} = \{f|IG_f > \bar{IG}\}$
Sort the features according to IGR
return F''

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 chi-square method is implemented in a Python library scikit-learn[22].

B. Experiment Results

1) Feature subset: Table II shows the features selected by our method and their descriptions. From Table II we make the following observations:

1) 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 shouldnt 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.
2) Another type of important features 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.
3) 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.
4) Some miscellaneous. In our result, the ACK Flag count is selected in our feature subset. In fact the number of ACKs accounts for a large proportion of the TCP flow.

TABLE II
SELECTED FEATURES AND THEIR DESCRIPTION

Feature Name	Description
Bwd Pkts/s	Number of backward packets per second
Fwd Seg Size Min	Minimum segment size observed in the forward direction
Init Fwd Win Byts	Number of bytes sent in initial window in the forward direction
Flow Pkts/s	flow packets rate that is number of packets transferred per second
ACK Flag Cnt	Number of packets with ACK
Flow IAT Mean	Average time between two flows
Bwd IAT Max	Maximum time between two packets sent in the backward direction
Idle Mean	Mean time a flow was idle before becoming active
Fwd Pkt Len Min	Minimum size of packet in forward direction
Flow Duration	Flow duration
Bwd Pkt Len Mean	Mean size of packet in backward direction
Pkt Len Min	Minimum length of a flow
Fwd Pkts/s	Number of forward packets per second
Fwd IAT Tot	Total time between two packets sent in the forward direction

2) Training Time: The result of training time in three situations is shown in Table III and Figure 3. Our method get the shortest training time in three situations. The decision tree will calculate the information gain of all features in the feature set. In fact, our method selects key features previously so it can shorten the training time of the model.

TABLE III
TRAINING TIME COMPARISON

	Full	Chi-square	Hierarchical
Training Time(s)	308.588	83.771	**61.198**

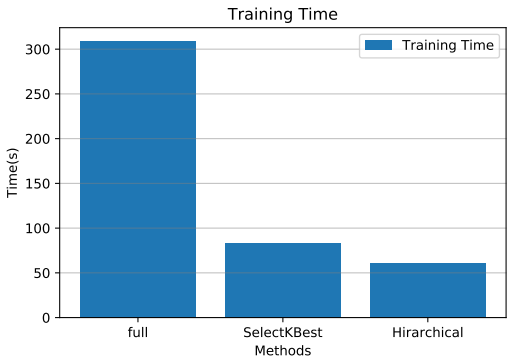

Fig. 3. Training Time

3) Metrics: The metrics of model include accuracy, precision, recall and f1-score. The test result is shown in Table IV and Figure 4. The result indicates that Except that it is only slightly lower in accuracy, our model is generally better than the model trained with the full feature set. Besides, the model using the features selected by our method performs better than the model trained with the features selected by chi-square method.

TABLE IV
METRICS COMPARISON

	Hierarchical	Full	Chi-square
Accuracy(%)	**97.96**	97.91	94.30
Precision(%)	**80.40**	81.38	72.43
Recall(%)	**79.08**	77.35	67.76
F1-Score(%)	**79.14**	78.52	68.78

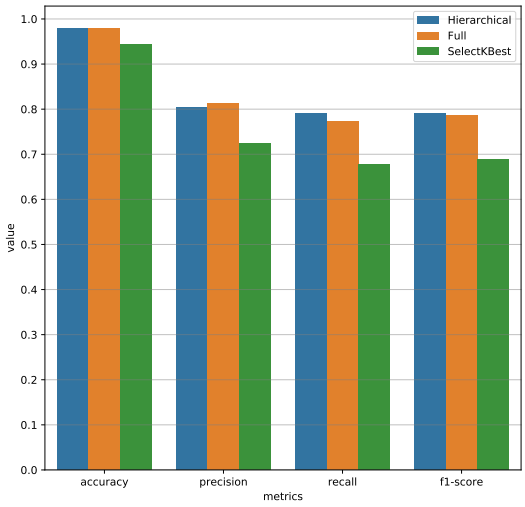

Fig. 4. Accuracy, Precision, Recall and F1-score

C. Confusion Matrix

VI. DISCUSSION

VII. CONCLUSION

This paper proposed a novel hierarchical feature selection method for network anomaly detection. We applied feature clustering algorithm using correlation-coefficient-based distance between features on network flow traffic data set, then ranked these feature cluster centers using information gain and information gain ratio simultaneously. After that we chose decision tree as our training algorithm to train the model based on the selected feature set. The experiment results shows our method can select more critical features which can determine whether a network flow is an attack.

In the future, we will continue researching the methods of real-time network data analysis and real-time model of training and detection for network traffic.

REFERENCES

[1] Github triumphant over its largest ever cyber pummeling. [Online]. Available: https://fortune.com/2015/04/03/github-ddos-china/
[2] Read dyns 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*, vol. 88, p. 101645, 2020.
[5] M. H. C. Law, A. T. Figueiredo, and A. K. Jain, "Simultaneous Feature Selection and Clustering Using Mixture Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1154–1166, 2004.
[6] X. Yang, Q. Wang, and Y. Wang, "Feature selection based on network maximal correlation," in *International Symposium on Wireless Personal Multimedia Communications, WPMC*, vol. 2017-Decem, no. Mc, 2018, pp. 448–452.
[7] J. Wu and C. Li, "Feature selection based on features unit," in *Proceedings - 2017 4th International Conference on Information Science and Control Engineering, ICISCE 2017*, 2017, pp. 330–333.
[8] A. Yassine, C. Mohamed, and A. Zinedine, "Feature selection based on pairwise evalution," in *2017 Intelligent Systems and Computer Vision, ISCV 2017*, 2017, pp. 1–6.
[9] S. Yang, F. Nie, and X. Li, "Unsupervised Feature Selection with Local Structure Learning," in *Proceedings - International Conference on Image Processing, ICIP*. IEEE, 2018, pp. 3398–3402.
[10] W. Ke, Y. Wang, X. Lei, and B. Wei, "Spark-Based Feature Selection Algorithm of Network Traffic Classification," in *Proceedings - 13th International Conference on Computational Intelligence and Security, CIS 2017*, vol. 2018-Janua, 2018, pp. 140–144.
[11] Z. Liu, Y. Li, and W. Ji, "Differential Private Ensemble Feature Selection," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 2018-July, 2018.
[12] 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 - International Electronics Symposium on Knowledge Creation and Intelligent Computing, IES-KCIC 2017*, vol. 1, 2017, pp. 46–49.
[13] X. Han, P. Liu, L. Wang, and D. Li, "Unsupervised feature selection via graph matrix learning and the low-dimensional space learning for classification," *Engineering Applications of Artificial Intelligence*, vol. 87, no. October 2019, p. 103283, 2020. [Online]. Available: https://doi.org/10.1016/j.engappai.2019.103283
[14] 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.
[15] 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. [Online]. Available: https://doi.org/10.1016/j.comnet.2018.01.007
[16] Y. ning Dong, J. jie 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. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2017.03.019
[17] G. Taskin, H. Kaya, and L. Bruzzone, "Feature selection based on high dimensional model representation for hyperspectral images," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2918–2928, 2017.
[18] J. K. Valadi, P. T. Ovhal, and K. J. Rathore, "A Simple Method of Solution For Multi-label Feature Selection," in *Proceedings of 2019 3rd IEEE International Conference on Electrical, Computer and Communication Technologies, ICECCT 2019*. IEEE, 2019, pp. 1–4.
[19] 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.
[20] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *4th International Conference on Information Systems Security and Privacy (ICISSP)*, 2018.
[21] Cicflowmeter. [Online]. Available: https://www.unb.ca/cic/research/applications.html#CICFlowMeter
[22] scikit-learn: machine learning in python. [Online]. Available: https://scikit-learn.org/

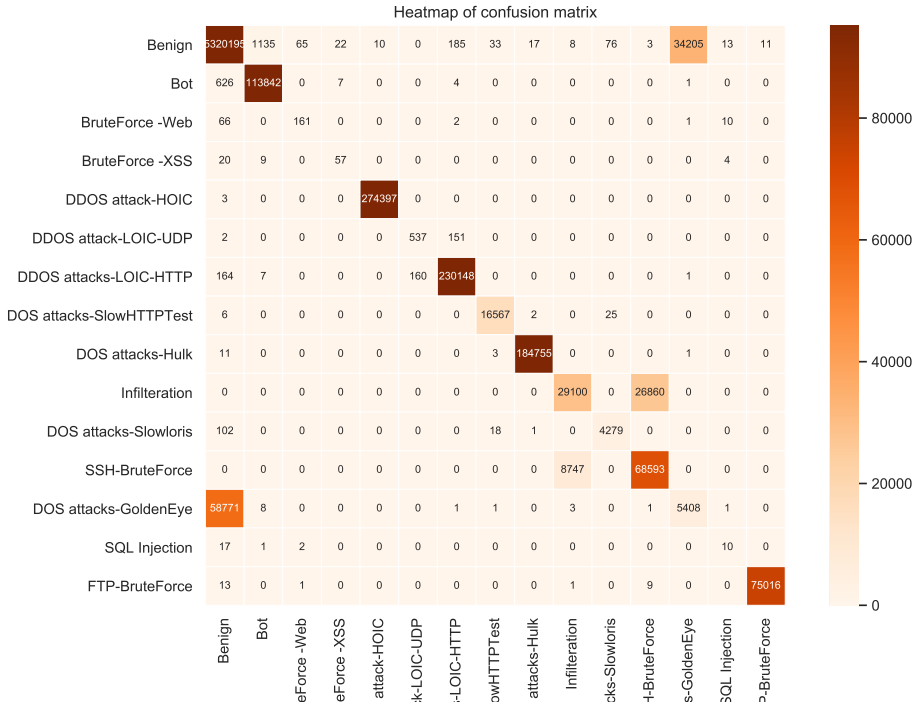

Fig. 5. Heatmap of confusion matrix