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

With the development of the Internet of Things, 5G technology, the improvement of network bandwidth and the emergence of more and more network security testing tools, 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].

Network attacks include not only DDoS attacks but also infiltration, brute force attacks, zero-day attacks and so on. They have their own characteristics on the pattern of flow traffic, which will be reflected in statistical characteristics that are different from normal traffic.(Here should be more references) However, traditional anomaly detection methods often only use the header information of the network packets. This may be effective in the face of typical network attacks, but if an attacker carefully constructs network packets whose header information is close to the normal mode, the traditional attack detection method may fail.

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

III. Problem Definition

The feature selection problem[4] can be described as a 6-tuple $FS = \{D, F, C, S, fs, E\}$, where $D = \{i_1, i_2, \dots, i_m\}$ is the dataset with m instances. $F = \{f_1, f_2, \dots, f_n\}$ is the feature set of D with n features. $C = \{c_1, c_2, \dots, c_k\}$ is the class label set with k labels. $S = \{s_1, s_2, \dots, s_l\}$, $l = 2^n - 1$ 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: $fs : F \to S$.

Thus the target of our algorithm is to find the best feature subset $\hat{S} \subset S$ subject to a best evaluation measure E.

IV. Hierarchical Feature Selection Method

A. Overview

Firstly the system is shown as Fig. 1. The proposed method is the first block in the figure, and it includes three steps or layers: The first step is preprocessing, which makes the data set 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 it finds all cluster center as the representative. 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[5]. The data set includes seven different attack scenarios: Brute-force, Heartbleed, Botnet, DoS, DDoS, Web attacks, and infiltration of the network from inside. The data set includes the captures network traffic and system logs of each machine, along with 76 features extracted from the captured traffic using CICFlowMeter-V3[6]. These 80 features are shown in Table 1.

In order to detect all types of attacks as much as possible, we integrated the network traffic that was originally scattered in each day's data, 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.

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 interfere proposed algorithms.

Our preprocessing strategy is described as follows:

1) Remove some nominal features and drop all rows contain missing value: 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 is as useless as the timestamp, because in this paper we assumed that attack may happen at any time and from any place. We remove these nominal features to focus on other statistical features.

Note that the removed features are only in the context of this paper. These feature 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 should 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.

After the clusters are generated, next step is to find the center of these clusters. This procedure is listed as Algorithm

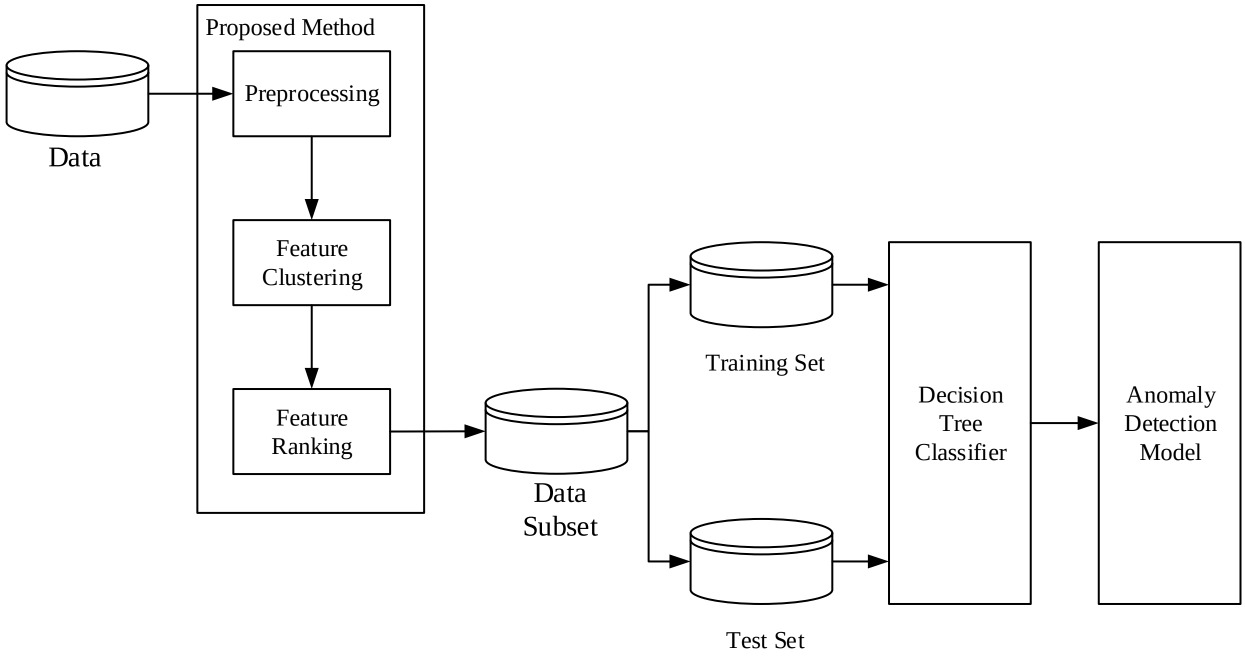

Fig. 1. Detection Framework

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

Features			
Flow Duration	Tot Fwd Pkts	Tot Bwd Pkts	TotLen Fwd Pkts
TotLen Bwd Pkts	Fwd Pkt Len Max	Fwd Pkt Len Min	Fwd Pkt Len Mean
Fwd Pkt Len Std	Bwd Pkt Len Max	Bwd Pkt Len Min	Bwd Pkt Len Mean
Bwd Pkt Len Std	Flow Byts/s	Flow Pkts/s	Flow IAT Mean
Flow IAT Std	Flow IAT Max	Flow IAT Min	Fwd IAT Tot
Fwd IAT Mean	Fwd IAT Std	Fwd IAT Max	Fwd IAT Min
Bwd IAT Tot	Bwd IAT Mean	Bwd IAT Std	Bwd IAT Max
Bwd IAT Min	Fwd PSH Flags	Bwd PSH Flags	Fwd URG Flags
Bwd URG Flags	Fwd Header Len	Bwd Header Len	Fwd Pkts/s
Bwd Pkts/s	Pkt Len Min	Pkt Len Max	Pkt Len Mean
Pkt Len Std	Pkt Len Var	FIN Flag Cnt	SYN Flag Cnt
RST Flag Cnt	PSH Flag Cnt	ACK Flag Cnt	URG Flag Cnt
CWE Flag Count	ECE Flag Cnt	Down/Up Ratio	Pkt Size Avg
Fwd Seg Size Avg	Bwd Seg Size Avg	Fwd Byts/b Avg	Fwd Pkts/b Avg
Fwd Blk Rate Avg	Bwd Byts/b Avg	Bwd Pkts/b Avg	Bwd Blk Rate Avg
Subflow Fwd Pkts	Subflow Fwd Byts	Subflow Bwd Pkts	Subflow Bwd Byts
Init Fwd Win Byts	Init Bwd Win Byts	Fwd Act Data Pkts	Fwd Seg Size Min
Active Mean	Active Std	Active Max	Active Min
Idle Mean	Idle Std	Idle Max	Idle Min

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

Algorithm 1 Feature clustering

```
Input:
    Data set $D = (\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_M)^{\mathrm{T}}$,
    feature set $F = (f_1, f_2, \dots, f_N)$,
    distance threshold $\delta$.
Output:
    Cluster set $C = \{c_1, c_2, \dots, c_K\}$
$C \leftarrow \emptyset$
for $i = 1, 2, \dots, n$ do
    if $\nexists c \in C$ s.t.$f_i \in c$ then
        cluster $c_{join} \leftarrow$ compare_and_join($D, f_i, C, \delta$)
        if $\exists c_{join}$ which $f_i$ can join then
            $c_{join}$.add($f_i$)
        else
            Create a new cluster $c'$
            $c'$.add($f_i$)
            $C$.add($c'$)
        end if
    end if
end for
return $C$
```

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

```
Input:
    Data set $D = (\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_M)^{\mathrm{T}}$,
    feature $f_i$,
    distance threshold $\delta$, currently existing clusters $C = \{c_1, c_2, \dots, c_K\}$
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 $k = 1, 2, \dots, K$ do
    Distance vector for cluster $c_k$, i.e.$\boldsymbol{d}(c_k) \leftarrow \emptyset$
    for $j = 1, 2, \dots,$ 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
```

Algorithm 3 Find the cluster center

```
Input:
    Data set $D = (\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_M)^{\mathrm{T}}$,
    clusters $C = \{c_1, c_2, \dots, c_K\}$ calculated in Algorithm 1
Output:
    A feature list $F' = \{f'_1, f'_2, \dots, f'_K\}$ whose features are the center of every cluster.
Feature list $f' = \emptyset$
for $i = 1, 2, \dots, K$ 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 $F'$
```

introduced.

$$H(D|f) = \\ -\sum_j p(f = f_j) \times \\ \sum_i p(L = l_i | f = f_j) \log_2 p(L = l_i | f = f_j) \quad (4)$$

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

$$IG(D, f) = H(D) - H(D|f) \quad (5)$$

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.

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

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.

A. Experiment Setup

All experiments are running on a PC with a 4-core Intel i5-4460S 2.90GHz CPU and 16GB memory. The chi-square method is implemented in a Python library scikit-learn[7].

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 $F''$
```

B. Experiment Results

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

1) Feature subset:

2) Training Time: The result of training time in three situations is shown in Table III and Figure 2. 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**

3) Metrics: The metrics of model include accuracy, precision, recall and f1-score. The test result is shown in Table IV and Figure 3. 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.

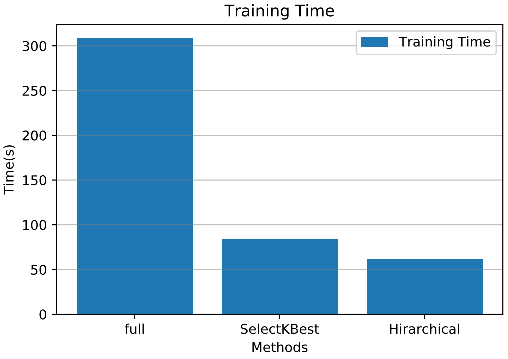

Fig. 2. Training Time

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

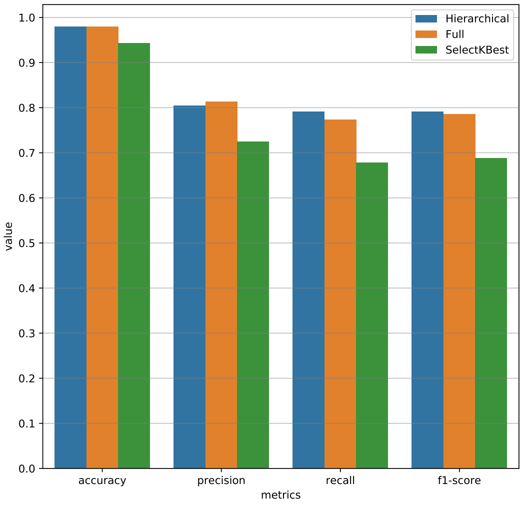

Fig. 3. Accuracy, Precision, Recall and F1-score

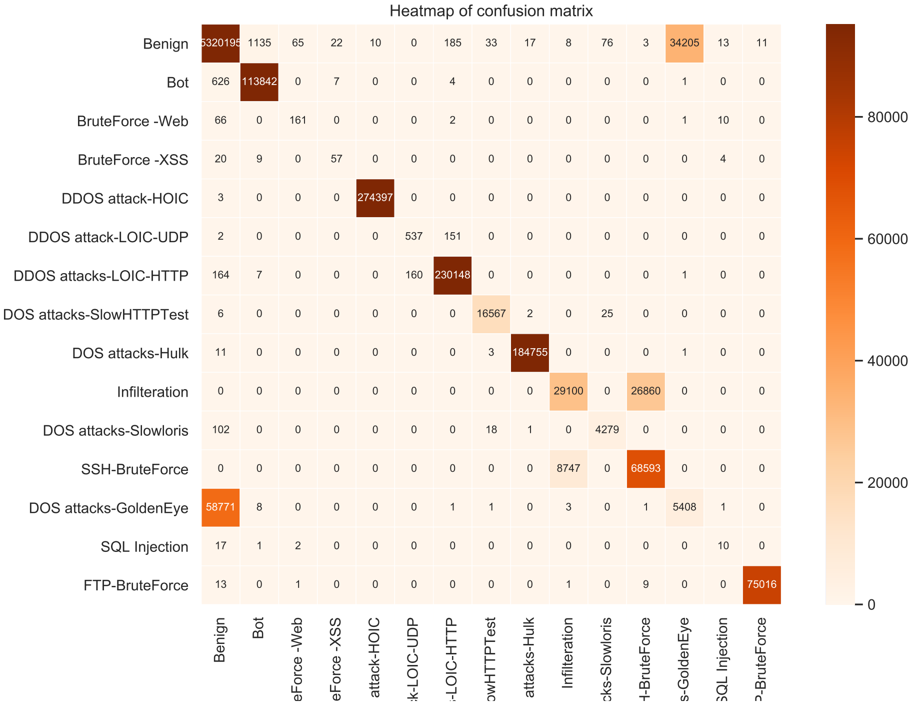

Fig. 4. Heatmap of confusion matrix

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 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. 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.

[5] 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.

[6] Cicflowmeter. [Online]. Available: https://www.unb.ca/cic/research/applications.html#CICFlowMeter

[7] scikit-learn: machine learning in python. [Online]. Available: https://scikit-learn.org/