JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 36, 765-776 (2020)
DOI: 10.6688/JISE.202007_36(4).0005

Classification and Recognition of Unknown Network Protocol Characteristics

YI-CHUAN WANG[1], BIN-BIN BAI[1], XIN-HONG HEI[1,+], JU REN[1,2] AND WEN-JIANG JI[1]
[1]*College of Computer Science and Engineering*
Xi'an University of Technology
Xi'an, 710048 P.R. China
[2]*School of Information Science and Engineering*
Central South University
Changsha, Hunan, 410083 P.R. China
E-mail: heixinhong@xaut.edu.cn

In recent years, unscrupulous hacker attacks have led to the information leakage of enterprise and individual network users, which makes the network security issue unprecedented concerned. Botnet and dark network, which use C & C channel of unknown protocol format to communicate, are the important parts. With the development of wireless mobile networks technology, this problem becomes more prominent. Classifying and identifying the unknown protocol features can help us to judge and predict the unknown attack behavior in the Internet of things environment, so as to protect the network security. Firstly, this paper compares the protocol features to be detected with the existing protocol features in the feature base through the vectorization operation of protocol features, selects the feature set with high recognition rate, and judges the similarity between protocols. The extracted composite features are digitized to generate 0-1 matrix, then Principal Component Analysis (PCA) dimension reduction is processed, and finally clustering analysis is carried out. A Clique to Protocol Feature Vectorization (CPFV) algorithm is designed to improve the efficiency of protocol clustering and finally generate a new protocol format. The experimental results show that compared with the traditional Clique and BIRCH algorithms, the proposed optimization algorithm improves the accuracy by 20% and the stability by 15%. It can cluster and identify unknown protocols accurately and quickly.

Keywords: wireless mobile network, IoT, protocol recognition, PCA, clique

1. INTRODUCTION

With the rapid development of science and technology, information network has gradually penetrated into all aspects of people's lives. People's demand for wireless communication is getting higher and higher, not only in terms of demand, but also security performance. Moreover, the popularity of mobile devices has brought it into the era of wireless mobile network [1]. It is also due to the continuous enrichment of computing, storage resources of mobile terminals, various mobile operating systems and wireless applications have gradually been developed, resulting in a large number of security hidden dangers in mobile terminals, which makes the security of wireless mobile network systems more and more difficult. Because many botnets are extending to the field of wireless terminals [2], many terminal devices have become the building nodes of botnets. These botnet nodes communicate by using unknown protocol and unknown C&C control channel [25], so we need some new means to resist these illegal means.

Received August 31, 2019; revised December 27, 2019; accepted February 4, 2020.
Communicated by Xiaohong Jiang.

[+] Corresponding author

Based on the above research status, this paper analyzes and studies the related technologies and methods of protocol feature extraction [3], and proposes the clustering and recognition methods suitable for bitstream protocol data. Our method helps to measure the boundary of the botnet in the Internet environment and find the nodes in it to defend and protect the security.

The rest of this article is arranged as follows. The first part introduced the development of unknown protocol in network security. The second part described our work in the unknown protocol analysis. In the third part, we proposed a new feature vectorization method and optimized the traditional Clique clustering algorithm. In sections 4 and 5, we analyzed the performance of the new algorithm from several angles and compared with other algorithms. Finally, we summarized our work.

2. PREPARATORY WORK

The common methods of protocol recognition include data mining and clustering.

Data mining has been applied in many fields of network data processing, among which association rules are a widely used data mining method. But association rule mining does not consider the order between things. Reference [4] proposed the classical association rule algorithm Apriori needs a large number of short sequences when mining sequences are longer, and it may produce a large number of candidate sets when the object database is larger, and the number of scanning transaction database will also bring I/O bottle neck. Reference [5] proposed an application layer protocol feature extraction algorithm based on the longest common subsequence. Although it does not need to exhaustively search the entire search space, the algorithm is less efficient. In reference [6, 7], an application layer protocol feature extraction algorithm is proposed respectively. These algorithms are the improvements of the classical association rule algorithm, Apriori algorithm [22-24], which makes it suitable for the extraction of protocol features.

In general, although the existing clustering methods can classify unknown protocols, most methods need to input the number of target clusters, and the accuracy of protocol identification is greatly influenced by the number of target clusters. The flow of unknown protocols cannot be classified automatically, and the practical application is limited. In view of the above situation, this paper takes bitstream data frame as the research object, takes multi-protocol recognition as the goal, analyzes the protocol characteristics, and proposes a new unknown network protocol feature recognition and classification clustering method to measure the boundary of zombie network and thus protect the network security.

3. NUMERICAL ANALYSIS OF BUILDING PROTOCOL FEATURES

In the process of the analysis and identification of the unknown protocol, the type of the protocol message is determined by the identification of the characteristic attribute of each protocol. Fig. 1 is a schematic diagram of an unknown protocol data frame identification process. Comparing the unknown protocol data frame with each of the protocol feature sets in the protocol feature library to be built, if the similarity reaches the decision threshold, the identification is successful. If the decision threshold is not reached, then the N-dimensional vector is first compared with each feature in the protocol feature library and generated. And then the $M*N$ matrix is generated by the vector of all the M unknown protocol message data and the dimensionality reduction operation is performed. Then cluster analysis is carried out. The same type of bit stream protocol data frames is divided into the same cluster by clustering. The data frame characteristics in the same cluster are similar to each other, and the effects of different data frame characteristics in other clusters are different. And finally, the protocol cluster is converted into a protocol format by the comparison protocol library.

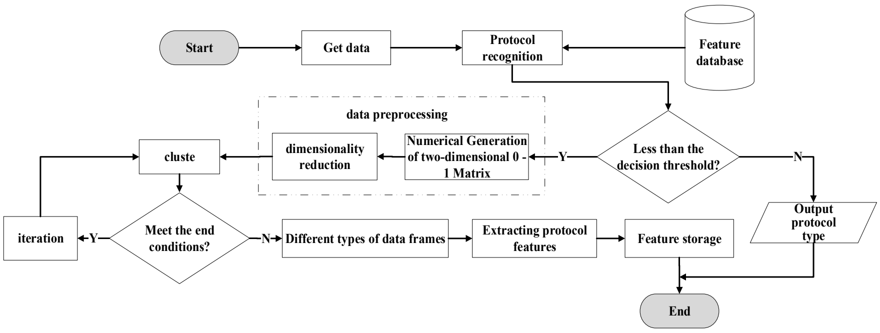


Fig. 1. Flow diagram of message data frame classification for bit stream unknown protocol.

3.1 Protocol Feature Vectorization

The numerical research of protocol data mainly includes two aspects, one is the vector of protocol characteristics, the other is the operation of data dimension reduction. These two steps can be regarded as the preprocessing of clustering data. In this section, the Vectorization of Protocol Feature algorithm (VPF) is proposed to compare the special stream protocol message data frame with the special stream protocol message data frame on the basis of the protocol feature library constructed [9]. The main idea of numerical algorithm is as follows: all the composite features in the protocol feature library are compared with the bitstream data frames, and the matching features are marked as '1' and the unsuccessful features as '0'. According to the position information in the composite feature, all the features in the protocol feature library are reordered, and the 0-1 matrix is generated by comparing it with the protocol feature library.

Assuming that there are N features, $Dn(n = 1, \ldots, n)$ in the protocol feature library D, the data frame x is set to '1' if the data frame overrides a feature, and if it is not covered, it is set to '0' by comparing the data frame with the protocol feature in the library. Each piece of data generates an N-dimensional vector:

$$V(x) = \left\{ \begin{matrix} 1; & Dn\ Covered\ (n = 1, 2, \ldots, n) \\ 0; & Dn\ Not\ Covered\ (n = 1, 2, \ldots, n) \end{matrix} \right\}. \tag{1}$$

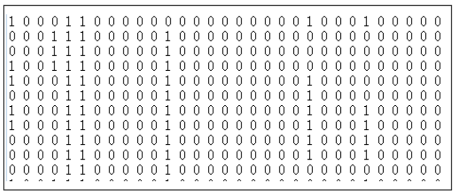
Assuming that there are M data frames in a set of data that are not recognized, the M term N dimensional vector is generated and the $M*N$ matrix $A(D) = A_{mn}$ is defined. At this time, the protocol features are digitized, where A_{mn} is in the form of 0-1 matrix.

The algorithm idea of protocol data frame vector quantification algorithm VPF is described as follows:

Table 1. VPF algorithm ideas.

Input:	Bit stream protocol data to be grouped by $M = \{M_1, M_2, \ldots, M_n\}$, Compound feature set $F_X = \{F_{X1}\#L_1, F_{X2}\#L_2, F_{X3}\#L_3, \ldots, F_{Xn}\#L_n\}$ in Protocol feature Library;
Output:	Numerical two-dimensional matrix A_{mn};
1	Count the number of all feature sequences in the feature set of various types of protocols in the statistical protocol feature library, and denoted as *count*;
2	Select a data frame in the bitstream protocol dataset M_i;
3	Definition vector *Vector*[*count*], All elements are initialized to 0;
4	For each element in the composite feature set $F_X = \{F_{X1}\#L_1, F_{X2}\#L_2, F_{X3}\#L_3, \ldots, F_{Xn}\# L_n\}$, if the feature sequence appears in the corresponding position then *Vector*[L_i] = 1, otherwise *Vector*[L_i] = 0;
5	That records all bit stream data. *Vector*[*count*], Each data frame is represented by a set of 0-1 vectors $M_i = \{x_{i1}, x_{i2}, x_3, \ldots, x_{in}\}$.
6	Read in each bit stream data, generate a two-dimensional numerical matrix A_{mn} and output.

Assuming that there are five features in the protocol feature library, A, B, C, D, E, and a data frame is compared with the protocol feature library, it is found that there are three features of A, C and D, then the data frame generates a vector {1, 0, 1, 1, 0}. N data frames can generate a two-dimensional 0-1 matrix of N*5. Fig. 2 is a partial figure of a two-dimensional matrix generated by 1439 pieces of data. In order to distinguish the data features, no operation is done when the feature comparison result is 0, and the column information of the feature is added when the comparison result is 1. For example, the vector of the above frame can eventually be represented as {11, 0, 31, 41, 0}.



```
1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0
0 0 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
1 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0
1 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0
0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0
0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0
0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Fig. 2. Two-dimensional matrix partial data.

3.2 Dimension Reduction Algorithm

Dimension reduction [10] is a preprocessing method when the data structure is complex and consists of a large number of dimensions. Dimension reduction preserves some important features of high latitude data, and completes the recognition and elimination of unrelated and unimportant variables [11]. In practical research and application, dimension reduction is helpful to improve the performance of data mining methods and reduce information variables, while saving us a lot of time and cost [12]. The main advantage of dimension reduction operation in data mining is that it improves the accuracy of data set classification and clustering, improves the computational efficiency and better data visualization.

PCA [13] is a linear dimension reduction method, which can generate a linear combination of original features and can project raw data on a reduced space. The main idea of PCA algorithm is to map n-dimensional features to k-dimensions [14]. K is reconstructed on the basis of the original features of the data, is a new orthogonal feature, and is also known as the principal component.

Description of the idea of PCA algorithm:

Table 2. PCA algorithm description.

Input	Data set $X = \{x_1, x_2, x_3, \ldots, x_n\}$. Need to go down to dimensions k;
Output	Sample set after dimension reduction Y;
1	Centralize the sample matrix (take the mean value);
2	Calculating the covariance matrix of the input dataset $\frac{1}{n}XX^T$;
3	The eigenvalues and corresponding eigenvectors of the covariance matrix $\frac{1}{n}XX^T$ of the data set are calculated.
4	According to the descending order of eigenvalues, the eigenvectors (that is, principal components) are arranged in descending order, and the largest k of them is selected.
5	The corresponding k eigenvectors are used as row vectors to form a new eigenvector matrix P.
6	When the projection data is built into a new space constructed by k eigenvectors, $Y = PX$ is the data after dimension reduction to k dimension, thus achieving the purpose of dimension reduction.

The following introduces the formulas that need to be used in several algorithm ideas:

Formula 1 sample mean:

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{N} X_i \tag{2}$$

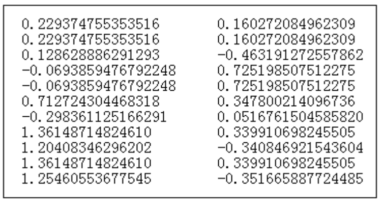
Formula 2 variance:

$$var = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})}{n-1} \tag{3}$$

Formula 3 covariance:

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{n-1} \tag{4}$$

The generated data with protocol feature column information is used as input, and some of the data is shown in Fig. 3 after dimension reduction by PCA algorithm.



```
0.229374755353516        0.160272084962309
0.229374755353516        0.160272084962309
0.128628886291293        -0.463191272557862
-0.0693859476792248      0.725198507512275
-0.0693859476792248      0.725198507512275
0.712724304468318        0.347800214096736
-0.298361125166291       0.0516761504585820
1.36148714824610         0.339910698245505
1.20408346296202         -0.340846921543604
1.36148714824610         0.339910698245505
1.25460553677545         -0.351665887724485
```

Fig. 3. Partial figure of PCA dimensionality reduction result.

3.3 Clique to Protocol Feature Vectorization (CPFV) Algorithm

At present, there are many mature clustering algorithms [16]. This paper mainly studies the Clique algorithm based on density. The algorithm is widely used in experiments because it does not need to know the category of sample data in advance, so it belongs to unsupervised learning algorithm. Even if some sample data lack some information, it can still complete the clustering work. The data frames of the same type of bit stream protocol are divided into the same class cluster by clustering, and the effect that the data frame features in the same class cluster are similar to each other and different from the data frame features in other clusters is achieved.

Clique algorithm [17] also has an inherent advantage in using the concept of subspace to cluster, the clustering formed by it doesn't necessarily exist in the full-dimensional space, but can exist in a subspace in the original full-dimensional space [18].

A property exploited by Clique: If a k-dimensional element is dense, then its projection in $(k-1)$ dimensional space is also dense. That is to say, given a k-dimensional candidate dense cell, if we check its $(k-1)$ dimensional projection cell and find that any one is not dense, then we know that the k dimensional cell is not dense. Therefore, we can infer the potential or candidate intensive units from the discovered intensive units in $(k-1)$ dimensional space. (Similar to Apriori nature).

In the present section, the bit stream protocol message data frame acquisition can be first matched on the basis of the constructed protocol feature library to represent one or more composite features of the bit stream protocol type, and the bit stream composite feature set is used as a clustering attribute. Then the bit stream data frames are clustered, the different data sets are separated, and a group of data frames with similar protocol types can be obtained. But it is necessary to specify the step size of the mesh when the Clique algorithm is adopted, that is, how many dimensions are finally generated. The selection of the step size directly affects the effect of the cluster. If the step size is too large, a large amount of memory space will be wasted. If the step size selection is too small, a large amount of data will be lost. However, it is sometimes impossible to obtain a step value that needs to be set in advance, and the result of the cluster may be directly affected once the initial step value selection is not appropriate so this paper has made some improvements to the Clique algorithm for the research of bit stream protocol data.

The basic principle of the Clique to Protocol Feature Vectorization (CPFV) algorithm: Enter the result data $Data = \{x_1, x_2, x_3, \ldots, x_n\}$ after PCA dimension reduction. The algorithm first identifies the dense subspace with clustering. That is, it first traverses all the data to determine the dense unit in one dimensional case, and then the k dimensional candidate dense cell can be obtained from the $k1$ dimensional dense cell by using the candidate set generation algorithm. The algorithm terminates when no new candidate set generated. Next, the recognition clustering is carried out, and the depth-first algorithm is used to find the clustering algorithm in the space, and then a minimized description is generated for each cluster. Finally, the generated clustering information is processed and converted into protocol information. In the process of conversion to the protocol, we need to compare the protocol library to vector the protocol, and the converted protocol by this method will not have a large deviation in the original protocol, thus avoiding noise interference. The algorithm flow is shown in the Fig. 4.

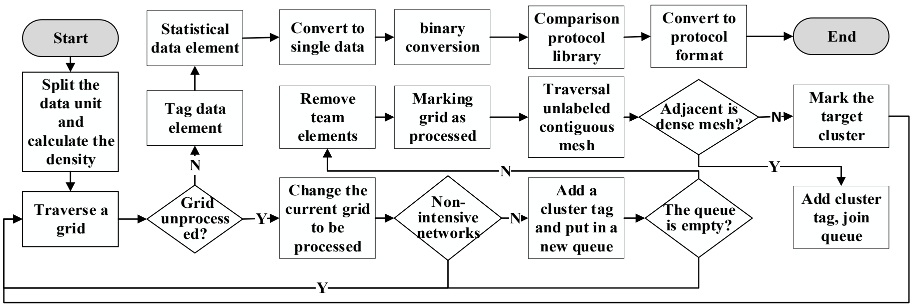

Start
Statistical data element
Convert to single data
binary conversion
Comparison protocol library
Convert to protocol format
End
Split the data unit and calculate the density
Tag data element
Remove team elements
Marking grid as processed
Traversal unlabeled contiguous mesh
Adjacent is dense mesh?
N
Mark the target cluster
N
N
Y
Traverse a grid
Grid unprocessed?
Y
Change the current grid to be processed
Non-intensive networks
N
Add a cluster tag and put in a new queue
The queue is empty?
Add cluster tag, join queue
Y

Fig. 4. Improved Clique algorithm flow chart.

The input of the CPFV is a two-dimensional dataset after dimension reduction, and the output is a protocol format with protocol characteristics. A total of 1439 pieces of data were used for the test, but only 243 coordinate points were shown in the clustering diagram, because when different data frames were compared with the protocol feature library, the 0-1 vector would appear the same situation, and the coordinate points would coincide in the dimensionality reduction, but it would not affect the accuracy and recall rate of the test clustering. The CPFV algorithm idea description:

Table 3. Improve the idea of Clique algorithm.

Input	Bit stream protocol data set X_{mn};
Output	Sample set after dimension reduction Y;
1.	Divided the data space $D = \{D_1, D_2, D_3, \ldots, D_n\}$, and calculated the density of each grid. According to the given threshold ω, $dense(D_i) > \omega ?1{:}0$, $i = 1, 2, 3, \ldots, n$, and $State(D_i)$ = "*unprocessed*", $i = 1, 2, \ldots, n$;
2.	Traverse all the grids and determine whether the current grid D_i has "unprocessed". If not, the next grid is processed, otherwise the following Steps 3-10 processing is performed until all grid processing is complete. Transfer Step 11;
3.	Change the grid mark to processed. If the grid D_i is not dense, switch to Step 2;
4.	If the dense grid D_i is given a new cluster tag, a queue is created and the dense grid D_i is placed in the queue.
5.	Determine whether the queue is empty, if empty, transfer to the next grid D_{i+1}, or Step 2; otherwise, do the following:
6.	Take out the grid elements at the head of the team and check for all adjacent "unprocessed" of the grid:
7.	$State(D_i)$ = "*processed*";
8.	If $dense(D_i) = 1$, the current cluster mark is given to the adjacent grid, then the current cluster mark is added into the queue and the Step 5 is carried out; otherwise, Step 6 is carried out;
9.	At the end of the inspection of the density connected region, the dense grid with the same mark forms the density connected region, that is, the target cluster.
10.	Modify the cluster tag, look for the next cluster, and turn to Step 2;
11.	The whole data set is traversed, and the data element is marked as the grid cluster mark value in which the data element is located;
12.	The data set after Step 11 is upgraded is counted and converted into a single protocol;
13.	Binary conversion of data $B_i = \{x_{i1}, x_{i2}, \ldots, x_{in}\}$;
14.	the converted protocol B_i is compared with the protocol library to generate a new protocol Y_{mn};

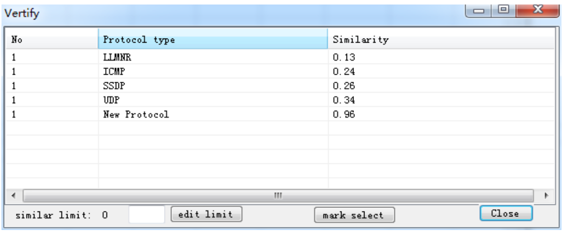
4. EXPERIMENTAL RESULTS AND COMPARISON

The experimental environment of this paper is Windows 7 system. The programming language is C# and the experimental platform is Visual Studio 2015. The 80000 known data used in this paper are from the data obtained by Wireshark packet capturing tool, among which 1029 unknown protocols are from the unknown protocol data of public network. In this paper, the experimental data is divided into 8 groups, each group has 10000 pieces of data. We classify and identify each group of data, and make records. In this paper, the dimension $k = 2$, threshold $\omega = 0.75$. Each experiment took about 8 hours. Table 4 is our experimental data set in TXT format, 80% for training and 20% for testing.

Table 4. Protocol data set.

Protocol type	Total number of data frames (Pieces)	Total data frame size (KB)	Protocol type	Total number of data frames (Pieces)	Total data frame size (KB)
HTTP-like	10000	31838	LLMNR-like	10000	4146
ICMP-like	10000	2908	SSDP-like	10000	5853
ARP-like	10000	2204	TCP-like	10000	10455
DNS-like	10000	3825	UDP-like	10000	13931

The following figure shows the protocol recognition diagram, which is the final result diagram of the protocol recognition of the captured unknown bit stream protocol with the LLMNR, ICMP, SSDP, UDP protocol after the new protocol is added to the protocol library. Through the graph, we find that for the unknown protocol, the effect of the original protocol is poor, and the protocol can't be distinguished. Through the protocol format generated by this method, the recognition rate of the protocol has been significantly improved.



No	Protocol type	Similarity
1	LLMNR	0.13
1	ICMP	0.24
1	SSDP	0.26
1	UDP	0.34
1	New Protocol	0.96

similar limit: 0

Fig. 5. Protocol identification diagram.

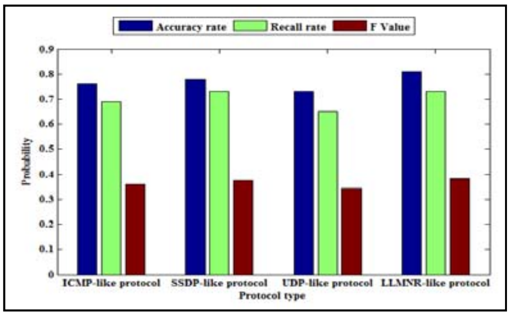


Fig. 6. Comparisons of experimental accuracy.

Fig. 6 is the experimental test results of recognition rate and recall rate of LLMNR-like protocol, ICMP-like protocol, SSDP-like protocol and UDP-like protocol. We find that the average accuracy of ICMP-like protocol is 81%, the average recall rate is 73%, and the average accuracy of other protocols is above 70%. The F1 value of all data is greater than 0.34.

In order to make the experimental comparison more accurate, we introduce the comparison experiment of BIRCH protocol recognition, as shown in Fig. 7, the accuracy and recall rate of the CPFV algorithm are compared with those of the traditional Clique algorithm and BIRCH algorithm. Because the steps of the traditional Clique algorithm are tedious, and many steps are used approximate algorithms, the accuracy of the clustering results is high and low so its stability is not good. From the graphs, we can see that for each bit flow protocol, the accuracy and recall rate of the CPFV algorithm are higher than those of the traditional Clique algorithm and BIRCH algorithm, and the CPFV algorithm is superior to the traditional Clique algorithm and BIRCH algorithm in terms of accuracy and stability. Therefore, the CPFV method proposed in this paper can accurately cluster the data frames according to the composite features.

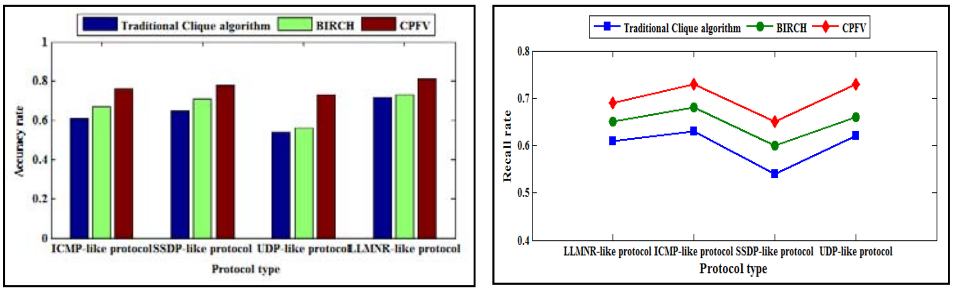


Fig. 7. Comparison of clustering accuracy and recall rate between different algorithms.

5. CONCLUSIONS

In this paper, the limitation of the current bit stream protocol data analysis is analyzed, and a bit stream protocol analysis and identification method is proposed. For the discriminant unrecognized protocol, the VPF vector operation of the protocol feature is carried out on the basis of the constructed protocol feature library. Next, the vector matrix is reduced by PCA. The bit stream data frames are then clustered and a set of data frames with similar protocol types can be obtained. By CPFV algorithm clustering, the same type of bit stream protocol data frames is divided into the same cluster. The data frame characteristics in the same cluster are similar to each other, and the effect of different data frame characteristics in other clusters is different. And finally, the clustering effect of the CPFV algorithm is evaluated by using the accuracy rate, the recall rate and the F1 value. In contrast to the accuracy of the traditional Clique algorithm and the BIRCH algorithm, we find that the CPFV algorithm is superior to the traditional Clique algorithm and the BIRCH algorithm in terms of accuracy or stability. The CPFV algorithm is higher than the traditional Clique algorithm by about 20% in accuracy, and the stability is higher than that of the traditional Clique algorithm by about 15%.

ACKNOWLEDGMENT

This research work is supposed by the National Key R&D Program of China (2018 YFB1201500), National Natural Science Founds of China (61602376, 61773313, 61602 374, 61702411), National Natural Science Founds of Shaanxi (2017JQ6020, 2016JQ60 41), Key Research and Development Program of Shaanxi Province (2017 ZDXM-GY-098, 2019TD-014), Science Technology Project of Shaanxi Education Department (16JK1573, 16JK1552).

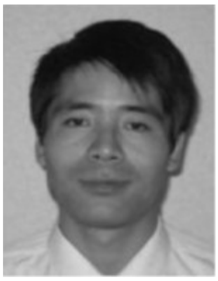
REFERENCES

1. B. D. Sija, Y. H. Goo, K. S. Shim, *et al.*, "Protocol reverse engineering methods for undocumented ethernet and wireless protocols," in *Proceedings of Symposium of the Korean Institute of Communications and Information Sciences*, 2017.
2. R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog *et al.*: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, Vol. 78, 2018, pp. 680-698.
3. J. Duchêne, C. L. Guernic, E. Alata, *et al.*, "State of the art of network protocol reverse engineering tools," *Journal of Computer Virology & Hacking Techniques*, 2017, pp. 1-16.
4. P. N. Vo and T. V. T. Ngoc, "Data mining for social network analysis using a CLIQUE Algorithm," *Cognitive Social Mining Applications in Data Analytics and Forensics*, IGI Global, 2019, pp. 160-187.
5. Z. Jie and L. Jianping, "Feature identification of unknown protocol," in *Proceedings of the 13th International Computer Conference on Wavelet Active Media Technology and Information Processing*, 2016, pp. 147-149.
6. L. Pradittasnee, S. Camtepe, and Y. C. Tian, "Efficient route update and maintenance for reliable routing in large-scale sensor networks," *IEEE Transactions on Industrial Informatics*, 2016, p. 1.
7. A. H. Mohsin, K. A. Bakar, and A. Zainal, "Optimal control overhead based multi-metric routing for MANET," *Wireless Networks*, Vol. 2, 2017, pp. 1-17.
8. X. Long, S. Wu, B. Cui, *et al.*, "Analysis of satellite observation task clustering based on the improved clique partition algorithm," in *Proceedings of IEEE Congress on Evolutionary Computation*, 2019, pp. 1314-1321.
9. F. Ricciato, P. Svoboda, J. Motz, *et al.*, "Traffic monitoring and analysis in 3G networks: lessons learned from the METAWIN project," *Elektrotechnik Und Informationstechnik*, Vol. 123, 2006, pp. 288-296.
10. J. Granjal, E. Monteiro, and J. S. Silva, "Security for the internet of things: a survey of existing protocols and open research issues," *IEEE Communications Surveys & Tutorials*, Vol. 17, 2015, pp. 1294-1312.
11. R. Mahmoud, T. Yousuf, F. Aloul, *et al.*, "Internet of things (IoT) security: Current status, challenges and prospective measures," in *Proceedings of IEEE 10th International Conference for Internet Technology and Secured Transactions*, 2015, pp. 336-341.
12. C. V. Wright, F. Monrose, and G. M. Masson, "On inferring application protocol behaviors in encrypted network traffic," *Journal of Machine Learning Research*, Vol. 6, 2006, pp. 2745-2769.

13. V. Hassija, V. Chamola, V. Saxena, *et al.*, "A survey on IoT security: application areas, security threats, and solution architectures," *IEEE Access*, Vol. 7, 2019, pp. 82721-82743.
14. A. Vlăduţu, D. Comaneci, and C. Dobre, "Internet traffic classification based on flows' statistical properties with machine learning," *International Journal of Network Management*, Vol. 27, 2016.
15. C. N. Lu, C. Y. Huang, Y. D. Lin, *et al.*, "High performance traffic classification based on message size sequence and distribution," *Journal of Network & Computer Applications*, Vol. 76, 2016, pp. 60-74.
16. W. Yu, F. Liang, X. He, *et al.*, "A survey on the edge computing for the Internet of Things," *IEEE Access*, Vol. 6, 2018, pp. 6900-6919.
17. A.Vlăduţu, D. Comăneci, and C. Dobre, "Internet traffic classification based on flows' statistical properties with machine learning," *International Journal of Network Management*, Vol. 27, 2017, p. e1929.
18. C. Perera, A. B. Zaslavsky, P. Christen, *et al.*, "Context aware computing for the Internet of Things: A survey," *IEEE Communications Surveys and Tutorials*, Vol. 16, 2014, pp. 414-454.
19. D. He, S. Chan, and M. Guizani, "Security in the internet of things supported by mobile edge computing," *IEEE Communications Magazine*, Vol. 56, 2018, pp. 56-61.
20. C. Xu, J. Ren, L. She, *et al.*, "EdgeSanitizer: Locally differentially private deep inference at the edge for mobile data analytics," *IEEE Internet of Things Journal*, Vol. 6, 2019, pp. 5140-5151.
21. H. Guo, J. Ren, D. Zhang, Y. Zhang, and J. Hu, "A scalable and manageable IoT architecture based on transparent computing," *Journal of Parallel and Distributed Computing*, Vol. 118, 2018, pp. 5-13.
22. K. Zhang, J. Ni, K. Yang, *et al.*, "Security and privacy in smart city applications: Challenges and solutions," *IEEE Communications Magazine*, Vol. 55, 2017, pp. 122-129.
23. J. Ren, H. Guo, C. Xu, *et al.*, "Serving at the edge: A scalable IoT architecture based on transparent computing," *IEEE Network*, Vol. 31, 2017, pp. 96-105.
24. Y. Wang, J. Ma, L. Zhang, *et al.*, "Dynamic game model of botnet DDoS attack and defense," *Security and Communication Networks*, Vol. 9, 2016, pp. 3127-3140.
25. Y. Wang, Y. Zhang, W. Ji, *et al.*, "Gleer: A novel Gini-based energy balancing scheme for mobile botnet retopology," *Wireless Communications and Mobile Computing*, Vol. 2018, 2018, Article No. 7805408.

Yi-Chuan Wang (王一川) received his Ph.D. degree in Computer System Architecture from Xidian University of China in 2014. He is an ACM member and a CCF member. Now he is a Lecturer in Xi'an University of Technology and with Shaanxi Key Laboratory of Network Computing and Security Technology. His research areas include cloud computing and networks security.

Bin-Bin Bai (白彬彬) received his bachelor's degree in Computer Science and Technology from Zhengzhou Business College in 2018. He is a CCF member. He is currently pursuing the master's degree in Computer Application at Xi'an University of Technology. His research areas include network security and big data.



Xin-Hong Hei (黑新宏) received his B.S. degree and M.S. degree in Computer Science and Technology from Xi'an University of Technology, Xi'an, China, in 1998 and 2003, respectively, and his Ph.D. degree from Nihon University, Tokyo, Japan, in 2008. He is currently a Professor with the Faculty of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China. His current research interests include intelligent systems, safety-critical system, and train control system.

Ju Ren (任炬) received the B.Sc. (2009), M.Sc. (2012), Ph.D. (2016) degrees all in Computer Science, from Central South University, China. During 2013-2015, he was a visiting Ph.D. student in the Department of Electrical and Computer Engineering, University of Waterloo, Canada. Currently, he is a Professor with the School of Computer Science and Engineering, Central South University, China. His research interests include Internet-of-Things, wireless networking systems, network computing and edge computing. He is a member of IEEE and ACM.

Wen-Jiang Ji (姬文江) received his B.S. degree in Information and Computational Science from Xidian University of China in 2006 and Ph.D. degree in Cryptology from Xidian University in 2013. He is currently a Lecturer in Department of Computer Science and Engineering from Xi'an University of Technology. His research interests include information and network security, privacy preserving in VANET and network simulation.