doi:10.3772/j.issn.1006-6748.2018.04.010

IAGNES algorithm for protocol recognition①

Deng Lijun(邓丽君)*, Tan Tiantian②**, Han Jingwen***, Tian Tian****
(*College of Information Engineering, Hunan Industry Polytechnic, Changsha 410073, P. R. China)
(**Computer Science College, National University of Defense and Technology, Changsha 410073, P. R. China)
(***Vancouver School of Economics, University of British Columbia, Vancouver V6T 1Z2, BC, Canada)
(****Management College, Xi'an Jiaotong University, Xi'an 710049, P. R. China)

Abstract

In the process of protected protocol recognition, an improved AGglomerative NESting algorithm (IAGNES) with high adaptability is proposed, which is based on the AGglomerative NESting algorithm (AGNES), for the challenging issue of how to obtain single protocol data frames from multi-protocol data frames. It can improve accuracy and efficiency by similarity between bit-stream data frames and clusters, extract clusters in the process of clustering. Every cluster obtained contains similarity evaluation index which is helpful to evaluation. More importantly, IAGNES algorithm can automatically recognize the number of cluster. Experiments on the data set published by Lincoln Laboratory shows that the algorithm can cluster the protocol data frames with high accuracy.

Key words: improved AGglomerative NESting algorithm (IAGNES), protocol recognition, bit-stream

0 Introduction

Communication protocol plays a necessary role in current society, with the rapid development of communication engineering, more and more users concern about communication network security, focusing on communication network security, and researchers have proposed several efficient solutions, such as solutions based on a probabilistic joint detection model[1], the distributed TCAM coprocessor architecture[2], etc. The formats of communication protocols are always necessary during the implement of these solutions. Because of lacking development document and format description, protocol recognition is a better solution for protected protocols, and it is not only applied in information protection and network supervision, but also a vital task for information investigation, etc.

A communication protocol is a set of rules (standards or conventions) established for data exchange in a communication network. Different communication protocols always have different data structures and formats, such as IP header definition, the definition of ARP protocol, etc[3]. The similarity values[4,5] of data frames in one protocol are always in a certain range.

Parsing the bit-stream data frames are always necessary in the process of protected protocol recognition, however, it is difficult to parse the bit-stream data frames without any type information, and how to recognize the bit-stream packets grasped in data link layer and unrecognized by crawlers is a challenge to protected protocol recognition. In order to further analyze protected bit-stream protocol, it is necessary to obtain single protocol data frames from the bit-stream data frames, yet how many formats of protected protocols in bit-stream data frames is not known, so it is an unsupervised classification process.

Based on the two important steps of AGglomerative NESting algorithm (AGNES)[6], an improved AGglomerative NESting algorithm (IAGNES) is proposed to classify the bit-stream data frames of protected protocol by hierarchical clustering algorithm, and automatically recognize the number of clusters for more practical value. From the aspects of accuracy, efficiency, applicability and usability, the algorithm advantages are summarized as follows.

1) Usability: IAGNES can extract results during clustering process, and find the clusters with higher similarity than the clusters preset in time, which can increase the scalability of the algorithm to a certain extent. Because every cluster obtained by IAGNES has a similarity which represents the similar degree of all objects inside it and similarity can provide evaluation references to its cluster, IAGNES makes the method easi-

① Supported by the National Natural Science Foundation of China (No. F020704).
② To whom correspondence should be addressed. E-mail: happinesschild@126.com
Received on Nov. 18, 2017

er to be evaluated and work effectively, and solve the problem that is difficult to evaluate clustering result.

2) Applicability: Current methods of protocol data frames recognition require packets of target protocol for learning format information, and the number of protocols should be preset manually. IAGNES algorithm can determine the number of clusters automatically applied in data set without the packets of the target protocol. Therefore, IAGNES can enhance the applicability of current method applied in multi-protocol by clustering without any prior knowledge and the number of protocols.

3) Accuracy: Comparing the results of the clustering experiment, IAGNES algorithm is proved to have higher accuracy. Its accuracy rate is higher than accuracy rate of well-known clustering algorithms, such as Kmeans, SIB, and EM.

4) Efficiency: The time complexity of IAGNES algorithm is better than $O(N^2)$, spatial complexity is $O(n)$, n is the data frames' number, as cluster extraction described in usability. The speed of IAGNES will become faster and faster. The efficiency of IAGNES is higher than that of well-known clustering algorithms, such as KMeans, SIB and EM.

1 Relevant work

In the aspect of protocol recognition, current technology includes two categories: port-based protocol identification method, the method based on deep packet inspection, and the method based on multi-pattern matching[7]. For a bit-stream protected protocol whose format is completely unknown, none of current methods can work effectively. The traditional solution is to use protocol reverse engineering technology to extract information from the protocol. According to the different analysis object, protocol reverse engineering methods, such as PI, Discoverer, RolePlayer, Ploygot, include two categories: the method based on message sequence analysis and the method based on instruction execution sequence analysis.

To some extent, typical protocol reverse engineering methods can achieve protected protocol recognition, but need more manual efforts and time consumption. PI can only obtain poor nested structures of fields without fields' semantic, and RolePlayer depends on prior knowledge and has poor effect on complex protocol recognition. Discover also has some problems, for instance, the efficiency of protocol state information extraction is not ideal. Ploygot project is not a good solution to obtain the relationship between fields. At present, the principle of protected protocol recognition is to find out characteristics of protected protocols by data mining and pattern matching, and then use pattern matching features to identify protocols.

If the traffic of protected protocols is a continuous bit-stream, the first phase is to classify the bit-stream into fields, and the next phase is to classify the data fields. Focus on these, scholars have proposed some methods. In the domain of fields delimitation, Wang et al.[8] proposed cutting algorithm for bit-stream. For the threshold preset to present how many categories of protocols, the algorithm can give cutting scheme with high accuracy. Song et al.[9] improved AC algorithm to find the bound of fields based on the fingerprint characteristics. In the domain of fields recognition, Wang et al.[10] proposed a protocol recognition method in a specific environment based on protocol association rules, which can recognize and identify protected protocols through mining association rules. The shortcomings of this method is that it has to use data fields of protected protocol as the learning data.

1.1 Hierarchical clustering algorithm

Hierarchical clustering algorithm is based on the distance between the objects, and usually constructs and maintains a cluster tree composed of clusters and child-clusters until meeting the termination conditions which is preset, the typical algorithm is proposed by Han et al.[11]. In general, according to the turn of hierarchical decomposition, such as bottom-up or top-down, methods based on hierarchical clustering can be classified into two categories[12,13].

1) Hierarchical clustering in AGNES: AGNES algorithm uses the bottom-up strategy, defines each object as a cluster and then according to the standards preset, merges the clusters into larger ones. It stops merging when the number of clusters meets the threshold preset or other termination conditions. This strategy is adopted by most hierarchical clustering. Combining different algorithms with the corresponding application scenarios, it can define different similarities, including the similarity between clusters and the similarity between objects.

2) Split hierarchical clustering algorithm: Split hierarchical clustering algorithm adopts a top-down strategy, treats all objects as one cluster and then classifies the clusters into smaller clusters and stops when reaching a termination condition or each object becomes a cluster.

1.2 Distance measurement

The selection of merging point or splitting point is important to hierarchical clustering algorithms. Accord-

ing to the similarity between clusters, the methods for distance measurement shown in Fig. 1 include the following four categories[13,14].

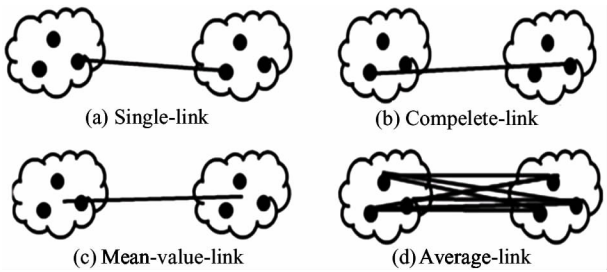

Fig. 1 Categories for distance measurement

1) Single-link method. This method defines an object from cluster A as i, and defines an object from cluster B as j. C_A represents a set of objects inside cluster A, C_B represents a set of objects inside cluster B. The distances between all objects of the two cluster will be calculated, and the nearest object pair (i, j) which has the minimum distance value $D_{\min(CA,CB)}$ will be selected to represent the distance between two clusters. The calculation method is

$$D_{\min(CA,CB)} = \min_{i\in CA,\, j\in CB} | i - j | \tag{1}$$

2) Complete-link. Similar to Single-link, two objects i, j belong to clusters A, B respectively. The pair (i, j) which has the maximum distance value $D_{\max}(C_A, C_B)$ will be selected to represent the distance of two clusters. The calculation method is

$$D_{\max(CA,CB)} = \max_{i\in CA,\, j\in CB} | i - j | \tag{2}$$

3) Mean-value-link. The pair (i, j) which in the center of cluster A and cluster B will be selected to represent the distance of two clusters. The calculation method is as Eq. (3):

$$D_{\text{mean}(CA,CB)} = | m_A - m_B | \tag{3}$$

where, m_A denotes the mean value of C_A, m_B denotes the mean value of C_B.

4) Average-link. The distances between all objects of cluster A and B will be calculated, and the average value of distance between cluster A and cluster B will be selected to represent the distance of two clusters A and B. The calculation method is as Eq. (4).

$$D_{\text{mean}(CA,CB)} = \frac{1}{n_A n_B} \sum_{X_i\in C_A} \sum_{X_j\in C_B} | i - j | \tag{4}$$

where, n_A denotes the number of data objects in C_A, n_B denotes the number of data objects in C_B.

2 IAGNES algorithm

2.1 AGNES

AGNES algorithm is a cluster analysis method whose primary purpose is to group objects based on its characteristics. The framework of AGNES algorithm is shown in Fig. 2. It starts with the individual objects until the objects are fused into a single cluster. It is a simple algorithm with high accuracy. It treats each object as a cluster, then it merges clusters into the larger one, and according to the standards preset, it will stop clustering when meeting the threshold of clusters number or other termination conditions which is preset. In general, the indicator of merging is the similarity between objects and clusters. The disadvantages of AGNES algorithm are described as follows[15].

Algorithm 1 Framework of AGNES

Input: A set of data containing C objects;
The clusters number n which is desired;
Output: n clusters;
1: Treat each object as a cluster;
2: According to the definition of distance, find two clusters with greatest similarity;
3: Merge two cluster into a new cluster;
4: Treat (c_i, c_j) as an unit and delete c_j;
5: **Repeat** until the number of clusters reaches the preset value;
6: **return** result;

Fig. 2 The framework of AGNES algorithm

- This algorithm lacks of scalability and has higher complexity.
- It usually needs to preset the number of clusters manually, but in the practical process, it is difficult to know the number of clusters.
- The selection of the merging point plays an important role in algorithm, which directly affects the accuracy of clustering.
- The quality of the clustering results is often difficult to evaluate.

2.2 IAGNES algorithm

To solve the problems that the traditional AGNES algorithm is not scalable enough, the number of clus-

ters n depends on manual definition, and it is difficult to evaluate the clustering results, and the following optimizations are proposed[16].

- Preset the merging condition as the input of algorithm, such as the minimum similarity threshold and other terminate conditions which can be reached easier. These parameters can make the algorithm calculate the number of clusters automatically.
- Extract clusters during the process of clustering. It can find better cluster results in time, and the scalability of the algorithm can be increased to a certain extent in this way.
- The quality of results of each cluster can be measured by the similarity, which can solve the problems that it is difficult to evaluate the clustering results.
- Only the latest clusters need to be saved in order to save time and memory space, select linear structure to improve the efficiency.

3 Implement

This sector introduces the implement of the improved AGNES algorithm.

3.1 Data preprocessing

The IAGNES algorithm needs to preprocess the data fields of original bit-stream. The steps of preprocess are as follows.

1) Process the first n bytes of the input data fields, discard the data over n bytes, and reserve the data less than n bytes, where n is 68B in the experiment.

2) Convert the data fields into 16 decimal formats, and each 4 bits is represented by a 16 digit number, such as 11111111, expressed as FF. Parameter m is the processing unit, the value of m can be half a byte, one byte or two bytes. Value of m is half a byte in our experiment, namely 4 bits. The values of m and n will be discussed in the evaluation section.

3) This preprocessing method is based on a certain similarity between the data fields of the same protocol which indicates that the same characters appear in certain locations. The rationality and limitations of the preprocess methods include:

Rationality

- The program uses 1 bit as the minimum one to read and write in processing data;
- The identifier bits of current protocol (such as addresses, the synchronization code, etc.) are usually set as an integer with multiple bytes, and some identifier bits are duplicated at the same location in the data fields.
- The data section of a data frame has little effect on the identification of the protected protocol. Therefore, set length of a data frame may include all the characteristic bits of the data frame as much as possible and include the data part as little as possible.

Limitations

- The characteristic fields of some protocols, such as control bits, may be smaller than 4 bits. These control bits are important features, but may not be effect when the process unit is half byte.
- Correcting and completing frame heads are necessary for the accuracy of the result, but this can be guaranteed by a reasonable segmentation.
- The appropriate data frame processing length m needs taking several experiments to obtain.

3.2 Algorithm implement

Similarity represents similar degree of two objects. The larger similarity is, the much possibly the two objects belong to one cluster, and the input data of IAGNES algorithm is hybrid data frame of multi-protocol. The definition of similarity of data frame pair (i, j) employs two methods. One is the similarity between the data frames defined by protocol data frame features, which directly reflect the fact that the same protocol may have the same character in the same position; the other method based on edit distance computation can be applied to the definition of string similarity. The method of string similarity computation was first proposed by Russian scientist Levenshtein[17], which is described as follows.

Definition 1 Similarity between the data frames. Let i and j be two data frames, and $similar_1(i, j)$ represents the similarity of i and j. The calculation method is

$$Similar_{1(i,j)} = \frac{sam(i, j)}{sum(i, j)} \tag{5}$$

where, $Sam(i, j)$ is used to calculate the number of the same byte between two sequences. It set sequences i and j to left aligned, set 4 bits as one unit to compare two sequences from left to right; $Sam(i, j)$ ++ when it finds the same byte, jump to the next byte of two sequences when the byte from two sequences are different.

$Sum(i, j)$ implies the value of compare times. In general, it is the length of the shorter sequence. The minimum value of similarity between i and j is 0, and the maximum value is 1.

The 16 decimal sequence translated from binary sequences is shown as follows.

00 10 7b 46 33 08

08 00 20 89 a5 9f00 10 7b

The similarity between two sequences is

$$Similar_{1(i,j)} = \frac{sam(i,j)}{sum(i,j)} = 4/16 = 25.00\%$$

Definition 2 String Similarity. Let i, j be two data frames, treat i, j as two strings, define the similarity between i and j to be $similar_{2(i,j)}$, and the calculation method is

$$Similar_{2(i,j)} = 1 - \frac{Distance(i,j)}{\max(length(i), length(j))} \quad (6)$$

where, $length(i)$ and $length(j)$ denote the length of i and j. $Distance(i, j)$ denotes the distance between i and j, which means number of operations times, such as inserting, replacing, deleting, etc. According to Eq. (6), the similarity of two sequences can be obtained:

$$Similar_{2(i,j)} = 1 - \frac{14}{18} = 22.22\%$$

Definition 3 Clusters similarity. Clusters similarity is the average value of the similarities of inner objects, similar to the average value of distance.

Similarity between clusters can be measured by Eq. (4), where $|i - j|$ represents the distance between i and j and can be replaced by $Similar_{1(i,j)}$ in Eq. (5), or $Similar_{2(i,j)}$ in Eq. (6), and the clusters similarity computation is

$$D_c(C_A, C_B) = \frac{1}{n_A n_B} \sum_{i \in C_A} \sum_{j \in C_B} Similar_{1(i,j)} \quad (7)$$

where, $Similar_{1(i,j)}$ can be replaced by $Similar_{2(i,j)}$. The value of the clusters similarity can range from 0 to 1.

Definition 4 $Simi_{\text{Min}}$. $Simi_{\text{Min}}$ refers to the minimum similarity merging which can only occur when the similarity between two clusters is larger than $Simi_{\text{Min}}$.

$Simi_{\text{Min}}$ can be set by manually. The algorithm terminates when no clusters can be merged into a larger one, the merging condition $Simi_{\text{Min}}$ works as follows:

$$D_c(C_A, C_B) Simi_{\text{Min}} \text{ (is merging)? Yes : No} \quad (8)$$

Definition 5 $Size_{\text{Min}}$. In the algorithm, $Size_{\text{Min}}$ refers to the minimum number of objects that an effective cluster needs to contain.

$Size_{\text{Min}}$ can be set manually. Check inner objects number of each cluster during the process of algorithm execution; use $Simi_{\text{Min}}$ to merge the cluster whose inner objects number is less than $Size_{\text{Min}}$; extract clusters with higher inner objects number than $Size_{\text{Min}}$ into results set. The condition for cluster effectiveness $Size_{\text{Min}}$ works as follows:

$$Cx.size() < Size_{\text{Min}} \text{ ? merge : extract} \quad (9)$$

Definition 6 *SimilarStep*. In the algorithm, the similarity decreases from 1 to the $Simi_{\text{Min}}$ gradually. Define the decline step of similarity each time as *SimilarStep*.

The smaller *SimilarStep* is, the more accurate the cluster similarity evaluation will be, but smaller *SimilarStep* can lead to higher complex of algorithm. The *SimilarStep* works as:

$$\text{while}(Simi_{\text{Min}});\ s -= SimilarStep \quad (10)$$

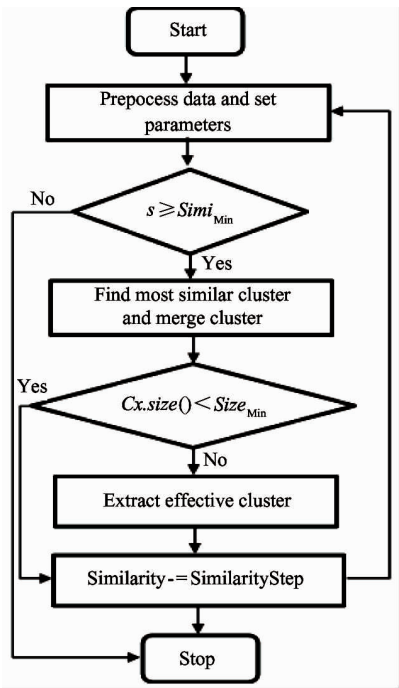

Fig. 3 Flowchart of algorithm execution

Algorithm 2 Frame work of IAGNES

Input: DataSet wihich is a data set include n objects;
Simi$_{\text{Min}}$; Size$_{\text{Min}}$; SimilarStep;
Output: Cluster Result Set
Index Result Set to save data objects id in Cluster Result Set
SimilaritySet to save data objects similarities
ScantDataSet to save the data objects which can't be recognized

```
1: Treat each object as an initial class cluster;
2: Set parameters;
3: for (i=1 to n-1)
4:  Find j meet constraint max (d(c_i, c_j)) and d(c_i, c_j)≥similar
5:  treat (c_i, c_j) as an unit, delete c_j and set flag=true;
6: end for
7: while(s)
8:  for (i=1 to n-1)
9:   if(! Cx.size()<Size_Min)
10:    Add c_i to cluster resultSet;
11:    Add l to indexResultSet;
12:    Add s to SimilarSet;
13:   end if
14: end for;
15: s-=SimilarStep
16: end while;
17: Obtain the scant data objects and push them into the ScantDataSet;
18: return result;
```

Fig. 4 The Framework of IAGNES Algorithm

The flowchart of IAGNES algorithm execution is shown in Fig. 3, the Framework of IAGNES Algorithm is shown as Fig. 4, and the detailed description is as follows: the time complexity of algorithm using Definition 1 is $O(k \times n^2)$. For the Definition 2, the time complexity of the algorithm is $O(k^2 \times n^2)$, where n represents the number of input data frames, k represents the length of the data frame, and it is always a constant, therefore, the time complexity is $O(n^2)$. Extract clusters which meet finish condition into the result set during cluster merging, the number of clusters which need to be merged decreases gradually, and the execution speed can be faster and faster. In addition, the algorithm merges the two clusters directly and uses linear structure (one dimensional array) to save the final clustering result rather than the middle process, so the space complexity is $O(n)$.

This algorithm is suitable for data frames which can be grasped at the link layer, and can't be recognized by the well-known protocol recognition tools. When the data frames are accumulated to a certain number, the algorithm can be used to do cluster analysis. According to the characters of different protocols, the data frames classification can be done, and single protocol data frames can be used for further protected protocol analysis, such as the semantic of each field, the protocol functions, etc.

4 Experiment and evaluation

Experiment is done on 9 protocols, and each one has 300 data frames, and the order of total input data is from 0 to 2 699.

4.1 Data preprocessing

In this experiment, the experimental data set of tcpdump published by Lincoln Laboratory is used, and binary data frames are extracted from 9 protocols, which were used as a protected protocol for experiments, such as DNS, HTTP, rip, SMTP, SSH, ARP, NTP, LLC, and loop protocols. The first n bytes of data frames are extracted, the data over n bytes are discarded and all fields less than n bytes are reserved, and then the bit-stream is converted to hexadecimal format. 68 bytes are set as the value of n, which is the minimum value including all the features of protocol information, the value which is too greater or too less may have a negative influence on accuracy of the results, and increase the amount of computation. Although different protocols data frames always have different length, due to the classified protocol by analyzing the similarity of characters whose length is always a constant, and data part of the protocols can cause interference on analysis results, and the length of each data frame is set to n bytes.

4.2 Experiment 1: using Definition 1

Using the IAGNES algorithm, the minimum similarity is set to be $Simi_{\text{Min}} = 0.1$, the minimum size of effective cluster $Size_{\text{Min}} = 5$, and step of similarity $SimilarStep = 0.1$ is declined. The experimental results are shown in Table 1. The false positive rate is 8.37%. There are 162 data frames whose similarities are less than 0.2 in the ScantDataSet. The data in the ScantDataSet are clustered and the clusters are extracted whose similarity are greater than or equal to 0.1, because of the size less than $Size_{\text{Min}}$, these clusters are called scant cluster, which can provide a reference to protocol recognition to some extent. The clusters which have over 5 data frames are shown in Table 2. The missing rate is 6%.

Table 1 Cluster result (Using Definition 1)

Cluster	Similarity	Object number	Accuracy	False positives
Cluster1	0.9	300	300	0
Cluster2	0.9	300	300	0
Cluster3	0.9	300	300	0
Cluster4	0.4	300	214	0
Cluster5	0.4	300	300	0
Cluster6	0.4	300	300	0
Cluster7	0.2	300	264	226
Cluster8	0.2	300	298	0
Cluster9	0.2	300	262	0

Table 2 ScantDataSet result (Using Definition 1)

Cluster	Similarity	Data frame number
Cluster1	≥ 0.1	86
Cluster2	≥ 0.1	5
Cluster3	≥ 0.1	17
Cluster4	≥ 0.1	13
Cluster5	≥ 0.1	5
Cluster6	≥ 0.1	5
Cluster7	≥ 0.1	5

4.3 Experiment 2: Using Definition 2

Set parameters according to Section 4.2, and the experimental results are shown in Table 3. There are 158 data frames whose similarities are less than 0.2 in the ScantDataSet. The data are clustered in the ScantDataSet and the clusters are extracted whose similarity

is greater than or equal to 0.1. The clusters which have over 5 data frames are shown in Table 4. The false positive rate is 10.26%, and the missing rate is 5.85%.

Table 3 Cluster result (Using Definition 2)

Cluster	Similarity	Object number	Accuracy	False positives
Cluster1	0.9	300	300	0
Cluster2	0.9	300	300	0
Cluster3	0.5	300	223	77
Cluster4	0.4	300	300	0
Cluster5	0.3	300	300	67
Cluster6	0.3	300	293	7
Cluster7	0.3	300	254	2
Cluster8	0.2	300	295	5
Cluster9	0.2	300	201	0

Table 4 ScantDataSet result(Using Definition 2)

Cluster	Similarity	Data frame number
Cluster1	≥ 0.1	14
Cluster2	≥ 0.1	15
Cluster3	≥ 0.1	23

4.4 Experiment 3: using the Weka tool

The packets grasped from 9 protocols are clustered using well-known clustering algorithms in Weka tools. Firstly, StringToWordVector is used to preprocess data, and 9 is specified as the number of clusters for each clustering algorithm. Classes to clusters evaluation was used to evaluate each clustering algorithm which is done for 3 times using different random seeds, and the mean value is taken as one evaluation index of results. For KMeans algorithm, the results are 60.3%, 69.82% and 66.26%; when random seeds are 10, 15, and 20, the average accuracy rate is 65.45%. For SIB algorithm, the results are 75.07%, 75.07%, 73.04%; when random seeds are 15 and 10, the average accuracy rate is 74.40%. For EM algorithm, the results are 80.04%, 73.63% and 64.74%; when random seeds are 100, 150 and 200, the average accuracy rate is 72.80%.

4.5 Evaluation

Experiment 1 shows that data fields extracted from 9 protocols are successfully merged into 8 clusters by the improved AGNES algorithm. As shown in Table 1, there are 3 clusters whose similarity is in the range of 0.9 – 1.0, and 3 clusters whose similarity is in the range of 0.4 – 0.9, and 2 clusters whose similarity is in the range of 0.2 – 0.4. It is because that similarities of clusters are less than 0.1 or the number of objects in clusters are less than 200. There are 162 data sequences in ScantDataSet.

The accuracy of clustering is 2312/2700 = 85.63%, showing better clustering results.

Experiment 2 shows that the accuracy of clustering is 2265/2700 = 83.89%, slightly lower than the results of Experiment 1, but the advantages of the calculation similarity by Definition 2 is that it can cluster more data sequences with less outliers and its disadvantages is that it tends to produce more false clusters in processing the data sequence with low similarity.

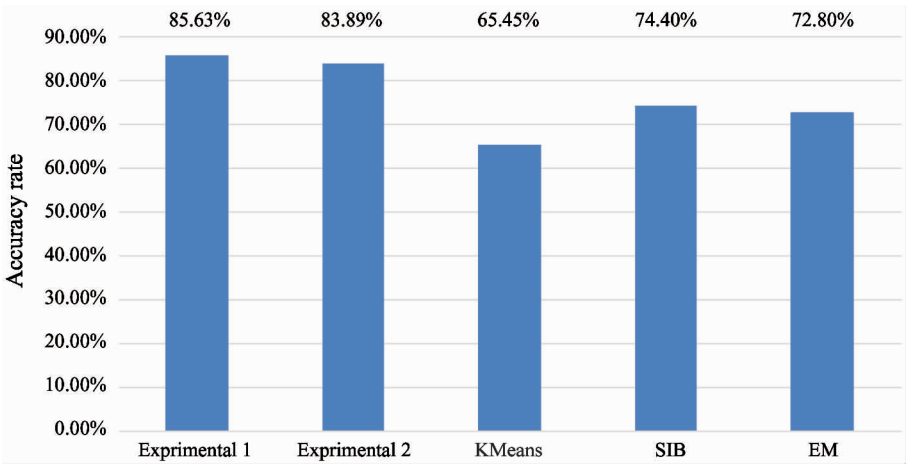

Fig. 5 The accuracy rate of five algorithms

Compared with the results of clustering algorithms in Weka tools, it is shown in Fig. 5 that the accuracy rate of the improved AGNES algorithm in Experiment 1 is 20.18% higher than that of KMeans algorithm,

11.32% higher than that of SIB algorithm and 12.83% higher than that of EM algorithm.

In addition, for the comparison of time complexity, the time complexity of the improved ANGES algorithm is at least $O(N2)$, due to extraction during clustering, the data size will decline rapidly and significantly, the spatial complexity of IAGNES is $O(n)$, where n is the data frame number. The time complexity of the Kmeans algorithm is $O(tKmn)$, and its spatial complexity is $O((m + K)n)$, where t is the iteration times, K is the number of inputs, m is the number of input data, and n is the dimension. The time complexity of the SIB algorithm is $O(l|X||T||Y|)$, and its spatial complexity is $O(|X|^2)$, where l is iteration times, $|X|$, $|T|$, $|Y|$ represent the number of elements in the random variable X, T, Y respectively. The complexity of the EM algorithm is determined by two factors: the number of iterations for convergence, and the complexity of each E and M steps. In Experiment 2 and Experiment 1, the objects in scant clusters are shown in Table 2 and Table 4. It can be seen that the similarity of scant cluster is less than 0.2 and can be no longer merged, but these scant clusters can also provide reference for data analysis.

4.6 Parameter discussion

Value of m and n. In the preprocessing part of the algorithm, there are two key parameters: one is to fetch first n bytes from the data frame, and the other is to treat m bytes as one data processing unit. If the value of n is too small, it may not include all the features of a protocol; if n is too large, it may contain too many characteristics of protocol. In the use of the AdaBoost classification algorithm to solve the protocol classification problem, researchers have proved that the first 64B of a data frame can contain almost all the characteristics of a protocol[18]. In this work, 68B is considered to be more appropriate after repeated experiments. It can be achieved by many experiments to obtain a more ideal n in the actual application process. In this paper, $m \in [1/2B, 2B]$, if the value of m is too small, such as 1bit or 2 bits, the features of a protocol will not be represented, or even not be found. The result is that the similarity between the data frames increases and the number of clusters is few; If m is too large, such as 4 bit or 8 bit, characteristics of a protocol will be redundant. The result is more clusters will be obtained when the similarity between the data frames is reduced. Therefore, if more cluster number is got in the application, the value of m should be adjusted.

Value of $Simi_{\text{Min}}$, $Size_{\text{Min}}$ and *SimilarStep*. In this paper, the IAGNES algorithm contains three important parameters: the minimum similarity for merging $Simi_{\text{Min}}$, the minimum size of one cluster $Size_{\text{Min}}$, and the decline step size of similarity *SimilarStep*. $Simi_{\text{Min}}$ is a threshold value that cluster can be further merged whose similarity value is greater than threshold. The algorithm will finish when the similarity between every two clusters is less than $Simi_{\text{Min}}$. Since the algorithm is merged from high similarity to low similarity, the change of $Simi_{\text{Min}}$ will not affect the cluster whose similarity is greater than the value, such as $Simi_{\text{Min}} = 0.1$ or $Simi_{\text{Min}} = 0.2$. The clusters with similarity greater than 0.2 will be clustered into one protocol, therefore, in the application, this value is appropriately reduced to obtain more clusters. After repeated experiments and analysis, for the clustering of protocol data frames, it is considered to be more appropriate to set $Simi_{\text{Min}}$ to 0.1, but there is no optimal value. $Size_{\text{Min}}$ is the minimum size of one cluster, when the number of data frames in a cluster is greater than or equal to $Size_{\text{Min}}$, then the clusters can be extracted. This parameter is set by experience, for example, it is considered that more than 200 data frames can be further analyzed, so $Size_{\text{Min}}$ is set to 200. Every cluster obtained by the algorithm contains at least 200 data objects. This parameter has no optimal value. Similarity reduction step size, *SimilarStep*, equivalent to computational accuracy, is shown in Table 3. It can be seen that there are 667 data frames with similarity in the range of 0.3 – 0.4, but the data object with similarity greater than $Simi_{\text{Min}}$ can not be reflected. It can be reflected if the *Similarstep* is set to 0.05, reducing the value of $Simi_{\text{Min}}$ will make a finer-grained classification, and the cyclic count will increase to $(1 - Simi_{\text{Min}})/SimilarStep$. With the increase of $Simi_{\text{Min}}$, the span of similarity between clusters will increase, and cycle count will decrease and algorithm will execute faster.

5 Scope

The algorithm uses similarity (distance) as the basis of distinguishing objects, so it can be used to data sets whose difference (distance) between elements can be measured by similarity, and the method can be used to compute different similarity (distance) in different scenarios.

Because of this, those differences between objects are distinguished by other characteristics of the dataset instead of similarity, it will be difficult to obtain satisfactory results. For example, in Table 3, among clusters with similarity in the range of 0.3 – 0.4, one clus-

ter with 667 data frames contains data more than one protocol. The data frames in different protocols may have similar characters, which should be considered in actual analysis.

6 Conclusion

Aiming at the problem of protected protocol classification, this paper designs and implements an improved hierarchical clustering algorithm, which can effectively classify protected multi-protocol data frames into single protocol data frames. The proposed algorithm is based on traditional AGNES algorithm. IAGNES algorithm can automatically determine the number of clusters. Each cluster has a similarity evaluation index. The proposed algorithm in the clustering process will detect the current clustering results, and extract the clusters satisfying conditions in time. In addition, there is no similarity evaluation index for obtaining scant clusters, and the scant clusters which have less objects than $Size_{Min}$ can't recognize the protocol. Compared with the traditional AGNES algorithm, the time complexity of the algorithm can be improved to deal with a large number of datasets in future work.

References

[1] Cai Z P, Chen M, Chen S G, et al. Searching for widespread events in large networked systems by cooperative monitoring[C]. In: Proceedings of the IEEE International Conference on Network Protocols, Francisco, USA, 2015. 123-133

[2] Cai Z P, Wang Z J, Zheng K, et al. Distributed TCAM coprocessor architecture for integrated longest prefix matching, policy filtering, and content filtering[J]. *IEEE Transactions on Computers*, 2013, 62(3):417-427

[3] Yao X, Li X. Protocol recognition technology for the CCSDS space data link layer[J]. *Aerospace Electronic Warfare*, 2012,02:26-29

[4] Aksoy A, Gunes M H. Operating system classification performance of TCP/IP protocol headers[C]. In: Proceedings of the IEEE Local Computer Networks Workshops, Dubai UAE, 2017. 112-120

[5] Tuck, Teo W, Singh R. Computer Networks, US 20040249975 A1[R]. Australian Intellectual Property Reporting, 2004

[6] Chattamvelli R. Data Mining Algorithms[M]. Alpha Science International, 2008. 20-38

[7] Peng W F, Du Z J. Research on the application layer protocol identification technology[J]. *Modern Computer*, 2015 (3):68-70

[8] Zheng J. An association analysis and identification for unknown protocol of bitstream oriented[J]. *Concurrency & Computation Practice*, 2016, 28(15):4067-4081

[9] Abdullah, Radhwan M, Song J. Efficient wireless network discovery method for vertical handover between WiMAX and WLAN[J]. *International Journal of Advancements in Computing Technology*, 2013, 5(11):1-10

[10] Wang Z F, Shan G L. Characteristic parameters extraction and correlation analysis of unknown protocol bit streams[C]. In: Proceedings of the IEEE International Conference on Instrumentation & Measurement, Qinghuangdao, China, 2015. 1502-1505

[11] Han K J, Kim S, Narayanan S S. Strategies to improve the robustness of agglomerative hierarchical clustering under data source variation for speaker diarization[J]. *IEEE Press*, 2008,16(8): 1590-1601

[12] Shi W Y. Research on multi-layer hierarchical clustering topology construction algorithm[J]. *Journal of Changchun Normal University*, 2016, (8):13-18 (In Chinese)

[13] Zheng L, Li T. Semi-supervised hierarchical clustering[C]. In: Proceedings of the IEEE International Conference on Data Mining, Vancouver, Canada, 2011. 982-991

[14] Srinivas M, Mohan C K. Efficient clustering approach using incremental and hierarchical clustering methods[C]. In: Proceedings of the IEEE International Joint Conference on Neural Networks, Barcelona, Spain, 2010. 1-7

[15] Zha C J, Dou Y N, Guo M J, et al. A new hybrid clustering algorithm based on stimulated annealing[C]. In: Proceedings of the IEEE International Conference on Intelligent Human Machine Systems and Cybernetics, Hangzhou, China, 2013. 94-99

[16] Kang Z H, Landry S J. An eye movement analysis algorithm for a multielement target tracking task: maximum transition – based agglomerative hierarchical clustering[J]. *IEEE Transactions on Human Machine Systems*, 2015, 45(1):13-24

[17] Jiang H, Han A Q, Wang M J, et al. Solution algorithm of string similarity based on improved levenshtein distance[J]. *Computer Engineering*, 2014, 40(1):222-227

[18] Haffner P, Sen S, Spatscheck O, et al. ACAS: automated construction of application signatures[C]. In :Proceedings of the ACM SIGCOMM Workshop on Mining Network Data, Philadelphia, USA, 2005. 197-202

Deng Lijun, born in 1982. She received her B. S. degree in engineering from Hunan Normal University in 2003 and M. S. degree from Hunan University in 2014. Her research interests include the design of algorithms for parallel processing, high performance computing and grid computing systems.