Automatic determination of types number of mixed binary protocols

Chunrui Zhang1,2, Shen Wang1, Dechen Zhan1, Yonghui Yang2
1Department of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, People's Republic of China
2Institute of Computer Application China Academy of Engineering Physics, Mianyang, 621900, People's Republic of China
E-mail: zhangcr@caep.cn

Abstract: In the absence of prior knowledge, it is a challenge to determine the number of protocol types in frames, which are completely unknown. These frames might be mixed with multiple protocols from the data link layer to the application layer. In this study, the authors combine the spectral clustering algorithm with a method of determining the number of protocol frame types, and further more design the refinement clustering of different hierarchical protocols based on the eigenvectors of the Laplace matrix. They use three clustering validity indices, which are Calinski–Harabasz index, Davies–Bouldinn index and Silhouette index, to quantify the clustering effect in order to calculate the number of protocol types. Extensive experiments on several open datasets obtain relatively satisfying results without prior knowledge and demonstrate the significant advantages of their methods clearly.

1 Introduction

Protocol reverse engineering (PRE) plays an important role in cyber security and information countermeasure [1]. Regardless of the security measures, protection methods, and security products used in the confrontation process, the confronters need to analyse communication data from various types of protocols, such as satellite communications, C&C botnet control protocols, supervisory control and data acquisition protocols [2], wireless communication protocols [3] and so on. Standards and specifications of these protocols are not available. Hence, the protocol specifications or standards must be reversed to determine the malicious behaviour. PRE is divided into two types [4, 5]: (i) the first type depends on the dynamic analysis which requires protocol implementation software, which is difficult to be obtained in a lot of confrontation scenarios; (ii) the second type depends on the network trace which is the main method used in the protocol reverse field.

A premise of PRE based on the network trace is the assumption that the analysed data is from only one type of protocol in many previous researches such as PI [6], Discover [7], and AutoReEngine [8]. However, in many scenarios, the protocol used for information countermeasure might be mixed with multiple protocols. Consequently, with the rapid expansion of applications and systems, a large number of unknown custom or private protocols might be mixed together to accomplish complex tasks such as Stuxnet [9] and Flame [10]. These protocols usually have the following characteristics:

- Protocol frames consist of binary data and rarely have recognisable text characters.
- Protocols are completely unknown from the data link layer to the application layer, such as those in satellite communications.
- The protocol is mixed in multiple layers. Fig. 1 illustrates an example of internet protocol where there are two kinds of protocols in layer_2 and five in layer_3.

As on the internet, the mixture of unknown binary protocols in different layers brings many difficulties to PRE. To achieve a reverse analysis of unknown protocols, we need to separate the mixed protocols into independent types firstly. Without prior knowledge, only relative clustering methods can be used to classify the mixed binary protocol frames. There are several difficulties in this part of work as follow:

- Binary protocol frames are mixed in different protocol layers. The effective distinguishing of them has not been researched.
- It is hard to find an effective way to determine the clustering quality to obtain good results at each layer.
- Binary protocol frames are difficult to be expressed in Euclidean space. We cannot obtain the centroids of clusters. Furthermore, without these centroids we can hardly calculate clustering validity index(CVI). Generally, CVIs are used to evaluate the clustering effectiveness. They can directly estimate the quality of a partition by measuring the compactness and separation of the clusters.

In this study, we propose a novel method to cluster the mixed binary protocol frames based on spectral clustering algorithm. The main contributions are as follows:

- Affinity matrix construction is a crucial step in clustering. We use the Smith–Waterman algorithm to get the pairwise similarity values and construct the affinity matrix of binary frames.
- We use three CVIs to guide the clustering process. Some researchers had compared nearly 30 indices by different properties and characteristics [11–13]. We select three best performing indices, which are, respectively, Calinski–Harabasz index, Davies–Bouldinn index and Silhouette index. The calculation method of these CVIs is given by the affinity matrix and the intuitionistic explanation is given.

![Fig. 1 Multi-layer of mixed protocol frames](image-url)
• The optimal values of CVI are used to distinguish protocol frames in the different layers.

2 Related work

Recently, researches on the classification of unknown protocols have two main aspects. One is based on semi-supervised learning and the other is based on unsupervised learning.

The semi-supervised method can be used to analyse some unknown protocol data. Based on Erm et al.’s study of flow correlation [14, 15], multiple studies used three-tuple {destination IP, destination Port, protocol} in training data to label unknown traffic. The main difference among these studies is that the labelled data was calculated by different clustering algorithms. Zhang et al. [16] used supervised learning and binary search to optimize parameters, thus improved the robustness of the classifier. Lin et al. [17] used the K-means algorithm to classify labelled data. They checked whether the labelled data was contained in the classification result to determine whether there were unknown protocols. Wang et al. [18] applied three constrained variants of the K-means algorithm to merge background information by performing hard or soft constraint satisfaction and distance metric learning during clustering. Ran et al. [19] used an alternative approach to clustering unknown streams by dynamically adding cluster centres and iteratively semi-supervised K-means to select optimal system parameters. However, these above semi-supervised methods must have labelled training data, and their implied premise is that the unknown traffic and the training data are at least from the same protocol at the data link layer, network layer, or even transport layer.

In the absence of prior knowledge and training data, some studies used unsupervised learning methods, such as clustering, for the classification of unknown protocol data. Clustering works suffer from two main problems: model selection and appropriate grouping [20]. The determination of cluster number is the key of the classification result to determine whether there were unknown protocols. Wang et al. [18] applied three constrained variants of the K-means algorithm to merge background information by performing hard or soft constraint satisfaction and distance metric learning during clustering. Ran et al. [19] used an alternative approach to clustering unknown streams by dynamically adding cluster centres and iteratively semi-supervised K-means to select optimal system parameters. However, these above semi-supervised methods must have labelled training data, and their implied premise is that the unknown traffic and the training data are at least from the same protocol at the data link layer, network layer, or even transport layer.

In the absence of prior knowledge and training data, some studies used unsupervised learning methods, such as clustering, for the classification of unknown protocol data. Clustering works suffer from two main problems: model selection and appropriate grouping [20]. The determination of cluster number is the key of the clustering method and it will directly affect the quality of clustering. Classical clustering methods include expectation maximisation-based [21–23], K-means algorithm-based [24, 25], and DBSCAN algorithm-based [26]. These methods aggregate clusters according to the similarity of traffic statistics and then establish the mapping relationship between clusters and classes. Some new methods have emerged in recent years. Wei et al. [27] proposed the optimised affinity propagation algorithm, which focused on calculating the similarity between samples. Although they improved the Gaussian kernel function to reduce the influence of the scale parameter σ, this algorithm was essentially the same as the density-based statistics. By setting the threshold in advance, samples with similarity smaller than the threshold would not be divided into the same cluster. Wang et al. [28] proposed an automatic unknown application protocol classification method by extracting the characteristics of unknown protocols and using an extended ROCK algorithm. Yu et al. [29] used service-based statistical features for cluster analysis and used the payload analysis tool (nDPI) to identify unknown protocols. There are other methods designed for malicious network traffic detection and analysis, such as the UNIDS proposed by Pedro et al. [30], which integrated subspace-based clustering, density-based clustering, and evidence accumulation. UNIDS can be used to identify unknown malware.

On the one hand, these unsupervised learning algorithms need to abstract the network traffics as vectors in multidimensional space from some important features, such as IP addresses, ports, session time and so on, which led to introduce certain prior knowledge in the actual application process. This also means that only the ethernet protocols can be classified and the unknown part of the protocol is mainly concentrated on the application layer. For protocols that are completely unknown from the link layer to the application layer, such as satellite communications, private custom wireless communications etc., the above methods are impossible to determine whether there is a five-tuple like Ethernet. Since these protocols are completely unknown, it is difficult to represent the feature vectors for calculation without prior knowledge. On the other hand, the most important thing is that these unsupervised methods must set the number of clusters large enough to obtain high-purity clusters, and until now there do not exist in-depth study on the determination of the number of clusters.

For the calculation of the number of clusters, Wen et al. used GMM [31] to obtain the clusters by continuously approximating the distribution of the original data. The number of GMM components was the number of clusters. However, the GMM tended easily to overfit, as a result, the data in the same cluster would be divided into different clusters. Some studies used graph-based clustering methods. They focused on how to construct effective affinity matrices, such as constructing affinity matrices through representations of low rank spaces [32, 33], and calculating affinity matrices by strengthening global constraints [34]. These methods were used to calculate clusters. However, affinity matrix solving processes were inherently time consuming and there was no clear indicator when solving the number of clusters.

3 Problem statement

Mixed binary communication protocol data is unstructured and the lengths of protocol frames are different, which is difficult to represent in an unknown unified multi-dimensional space. It is also hard to determine whether it has convexity in the multi-dimensional space or not and to know what the data density distribution is. Furthermore, the spectral clustering can analyse the non-convex data. In this study, we apply the spectral clustering method to cluster the unknown mixed protocol data and separate the data. This method is achieved by constructing the Laplace matrix and solving the matrix eigenvector to complete the clustering of the protocol. Then, three kinds of CVIs are used to evaluate the final clustering results. According to the definition of the spectral clustering method and CVI, the mathematical model is established as

$$\arg \max_{\mathbf{h}(k)} \left( \min \sum_{i=1}^{n} h_i L_i \right) \quad \text{s.t.} \quad h_i h_j = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

where \(f(k)\) is the computational function of CVI including \(f_{\text{CVI}}(k)\), \(f_{\text{DAM}}(k)\) and \(f_{\text{SAD}}(k)\) in Section 4.2. Unlike \(f_{\text{CVI}}(k)\) and \(f_{\text{DAM}}(k)\), \(f_{\text{SAD}}(k)\) needs to solve the minimum in Eq. 1. In order to unify the form, we add a minus sign before \(f_{\text{SAD}}(k)\). \(L\) is the Laplace matrix and \(L = D - S\), where \(S\) is the affinity matrix and \(D\) is the degree matrix, which can be obtained via \(S\) easily. \(h_i, h_j, \ldots, h_k\) are the first \(k\) eigenvectors of matrix \(L_{\text{sym}} = D^{-1/2}LD^{-1/2}\).

This model is a second-order constraint optimisation model. If \(f(k)\) can be expressed by \(h_i\), the second-order optimisation problem will be transformed into a first-order optimisation problem, and the problem can be solved by the gradient descent method. Unfortunately, the value of \(f(k)\) is determined by the row vectors of matrix \(V\), which is constructed by indication vectors such as \(\{h_1, \ldots, h_n\}\), rather than the column vectors. Also, the row vectors are reclassified to determine the value of \(f(k)\). Therefore, it is very difficult to address this problem. In this study, we split the calculation process into two phases when using the model: firstly, the value of \(h_i\) is calculated, then \(f(k)\) is calculated according to \(h_i\).

4 Mixed unknown binary protocols classification method

In this section, we present a detailed solution to the mathematical model, which we have established. We propose an algorithm to compute the number of protocol types.

4.1 Protocol similarity measure and distance calculation

We can infer that the affinity matrix \(S\) is the key to calculate the indication vectors \(\{h_1, \ldots, h_n\}\) from Section 3. However, the

IET Commun., 2019, Vol. 13 Iss. 12, pp. 1769-1775
This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)
construction of \( S \) is troublesome whose elements are the similarities between binary communication protocol frames. These frames are unstructured data, and the length of each frame is usually different. Thus, it is difficult to express in an unified multi-dimensional space when the protocol is unknown. We cannot directly use the general calculation method (such as Euclidean distance, cosine distance, and so on) to calculate the distance between two frames. It is also difficult to extract and use the frequent item sets, and characterise their various features because there are few printable characters in the binary protocols. Also, the probability of occurrence of each frequent term in the frame is different. If feature vectors are used to represent the binary frames, these frames will show high-dimensional and sparse characteristics, which make the similarity between the frames and difficulties in clustering. Therefore, we introduce Smith–Waterman algorithm to measure the similarity between unknown protocols. Smith–Waterman algorithm is described as follows:

Let \( A = a_1, a_2, \ldots, a_n \) and \( B = b_1, b_2, \ldots, b_m \) be the sequences to be aligned, where \( n \) and \( m \) are the lengths of \( A \) and \( B \), respectively. Construct a scoring matrix \( F \) and initialise its first row and first column. The size of the scoring matrix is \((n+1) \times (m+1)\) and \( F_{00} = F_{0l} = 0(0 \leq k \leq n \text{ and } 0 \leq l \leq m) \). Fill the scoring matrix using \( F(i,j) \) as shown in (2) where \( \sigma(a,b) \) is the similarity score of the elements that constituted the two sequences. Finally, starting at the highest score in the scoring matrix \( F \) and ending at a matrix cell that has a score of 0, trace back based on the source of each score recursively to generate the best local alignment.

\[
F(i, j) = \max \begin{cases} 
0, & \text{if } i=0 \\
F(i-1, j-1) + \sigma(a_i, b_j), & \text{if } i,j \neq 0 \\
F(i-1, j) + \sigma(a_i, -), & \text{if } i \neq 0, j = 0 \\
F(i, j-1) + \sigma(-, b_j), & \text{if } i = 0, j \neq 0 \\
F(i, j-1) + \sigma(-, -), & \text{if } i \neq 0, j = 0 
\end{cases}
\]  

(2)

So affinity matrix \( S \in \mathbb{R}^{n \times n} \) of protocol frames can be obtained by Smith–Waterman algorithm.

In order to quantify the clustering effect, most CVI calculations need to calculate the data centroid, but it is difficult to calculate the centroid of protocol frames when it is not expressed as a point in the multidimensional space. Therefore, in the calculation of \( f(k) \), different index calculation requires distance matrix and dot product matrix. The spectral clustering uses an affinity matrix, which can directly use the general calculation method (such as Euclidean distance, cosine distance, and so on) to calculate the distance between two frames. It is also difficult to extract and use the frequent item sets, and characterise their various features because there are few printable characters in the binary protocols. Also, the probability of occurrence of each frequent term in the frame is different. If feature vectors are used to represent the binary frames, these frames will show high-dimensional and sparse characteristics, which make the similarity between the frames and difficulties in clustering. Therefore, we introduce Smith–Waterman algorithm to measure the similarity between unknown protocols. Smith–Waterman algorithm is described as follows:

Let \( A = a_1, a_2, \ldots, a_n \) and \( B = b_1, b_2, \ldots, b_m \) be the sequences to be aligned, where \( n \) and \( m \) are the lengths of \( A \) and \( B \), respectively. Construct a scoring matrix \( F \) and initialise its first row and first column. The size of the scoring matrix is \((n+1) \times (m+1)\) and \( F_{00} = F_{0l} = 0(0 \leq k \leq n \text{ and } 0 \leq l \leq m) \). Fill the scoring matrix using \( F(i,j) \) as shown in (2) where \( \sigma(a,b) \) is the similarity score of the elements that constituted the two sequences. Finally, starting at the highest score in the scoring matrix \( F \) and ending at a matrix cell that has a score of 0, trace back based on the source of each score recursively to generate the best local alignment.

\[
F(i, j) = \max \begin{cases} 
0, & \text{if } i=0 \\
F(i-1, j-1) + \sigma(a_i, b_j), & \text{if } i,j \neq 0 \\
F(i-1, j) + \sigma(a_i, -), & \text{if } i \neq 0, j = 0 \\
F(i, j-1) + \sigma(-, b_j), & \text{if } i = 0, j \neq 0 \\
F(i, j-1) + \sigma(-, -), & \text{if } i \neq 0, j = 0 
\end{cases}
\]  

(2)

So affinity matrix \( S \in \mathbb{R}^{n \times n} \) of protocol frames can be obtained by Smith–Waterman algorithm.

In order to quantify the clustering effect, most CVI calculations need to calculate the data centroid, but it is difficult to calculate the centroid of protocol frames when it is not expressed as a point in the multidimensional space. Therefore, in the calculation of \( f(k) \), different index calculation requires distance matrix and dot product matrix. The spectral clustering uses an affinity matrix, which typically uses a radial basis function to convert between distance and similarity. This conversion requires pre-setting of parameters and affects the clustering results. In this study, the Smith–Waterman algorithm is used to calculate the similarity of frames to generate an affinity matrix. Since dot product can also express similarity measurement, affinity matrix is regarded as a dot product matrix directly.

Let \( X = \{x_1, x_2, \ldots, x_n\} \), \( x_i \) represent a protocol frame. In order to facilitate the conversion between the distance matrix and the affinity matrix, let \( x_i \) be a vector in a multidimensional space, and the length of the \( x_i \) is unified as a constant value \( \sqrt{1/2} \). Let \( s_{ij} = x_i^T x_j \), similarity measure can be described as

\[
d_{ij} = \frac{1}{2} \| x_i - x_j \|_2^2 = 1 - x_i^T x_j = 1 - s_{ij}
\]  

(3)

and

\[
d_{ij} = \sqrt{1 - s_{ij}}
\]  

(4)

where \( s_{ij} \) is the dot product of \( x_i \) and \( x_j \), and the affinity matrix is formed by \( s_{ij} \).

4.2 Clustering validity indices

Almost all of the CVIs are calculated by intra-class and inter-class distances with a number of other factors, such as square error, statistical attributes etc. We determine the number of clusters by the computational function \( f(k) \) of CVI, however, three CVIs mentioned in Section 1 need to use clustering centroid. For the special data in this study, we use the dot product to calculate CVIs in the case of difficult to obtain the centroid and give the intuitive explanation of three CVIs.

### 4.3 Calinski–Harabasz index

The first CVI is Calinski–Harabasz index whose expression is shown in (5)

\[
f_{CH}(k) = \frac{n - k}{k - 1} \frac{\text{tr}(S_W)}{\text{tr}(S_B)}
\]  

(5)

where \( S_B \) is an inter-class scatter matrix and \( S_W \) is an intra-class scatter matrix where they can be described as

\[
S_W = \sum_{i=1}^{n} (x_i - \mu)(x_i - \mu)^T
\]  

(6)

\[
S_B = n \Sigma - S_W
\]  

(7)

where \( \Sigma \) is the covariance matrix of all samples as shown in (8)

\[
\Sigma = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)(x_i - \mu)^T = \frac{1}{n} X (I - \frac{1}{n} ee^T) X^T.
\]  

(8)

Therefore

\[
\text{tr}(S_W) = \text{tr}(X^T X ) - \text{tr} \left( \sum_{i=1}^{n} \frac{1}{n_i} e_i e_i^T \right) X^T
\]  

(9)

\[
\text{tr}(S_B) = \text{tr}(X^T X ) - \text{tr} \left( \sum_{i=1}^{n} \frac{1}{n_i} e_i e_i^T \right) X^T - \text{tr} \left( \frac{1}{n} e e^T \right) X^T
\]  

(10)

The affinity matrix is a symmetrical matrix of size \( n \times n \). It can be treated as a square. The protocol frames have been clustered into \( k \) classes and \( n_1, n_2, \ldots, n_k \) are the number of samples of each classes. Assuming that the affinity matrix of protocol frames is moved after clustering, the same type of frame is reordered and displayed graphically as shown in Fig. 2. The pseudo area of a block graph is the sum of affinity matrix elements covered by it as shown in (11). For example, the pseudo area of the top left corner of the figure is the sum of affinity elements of \( n_i \) class. In the same principle, the line segment in the figure represents the sum of the elements covered by it, such as line \( i \).

\[
\text{pseudo \_area} = \sum_{i, j \in K} s_{ij}
\]  

(11)
4.4 Davies–Bouldin index

The second CVI is Davies–Bouldin index whose expression is shown in (12)

\[ f_{DB}(k) = \frac{1}{k} \sum_{i=1}^{k} \left( \frac{\max_{j \neq i} \{ d_{ij} / s_i \}}{\min_{j \in \{i, k \}} \{ d_{ij} / s_j \}} \right)^{\frac{1}{q}} \]

where \( S_i \) represents the value of intra-class dispersion when \( q = 2 \) is specified. According to the calculation rule of (4), we can obtain

\[ S_i = \left( \frac{1}{n_i} \sum_{j=1}^{n_i} \left( x_j - A_i \right)^{q} \right)^{\frac{1}{q}} \]

\[ d_{ij} = \left\| A_i - A_j \right\| \]

\[ d_{ij} = \sqrt{\sum_{s=1}^{l} s_{ij}^2 + \sum_{s=1}^{m} s_{il}^2 - 2 \sum_{s=1}^{l} \sum_{t=1}^{m} s_{st} s_{ij}} \]

4.5 Silhouette index

The third CVI is Silhouette index whose expression is shown in (15)

\[ f_{Sil}(k) = \frac{1}{n} \sum_{i=1}^{n} \frac{b_i - a_i}{\max \{ a_i, b_i \}} \]

Also, due to the calculation rule from (4), we can obtain

\[ a_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \sqrt{1 - s_{ii}} = \frac{1}{n_i} \sum_{j=1}^{n_i} d_{ij} \]

\[ b_i = \min_{j \neq i} \frac{1}{n_j} \sum_{k=1}^{n} \sqrt{1 - s_{kj}} = \min_{j \neq i} \frac{1}{n_j} \sum_{k=1}^{n} d_{kj} \]  

Let us consider Fig. 2 as a distance matrix. At the red part of the \( i \)th row indicate the distance between frame \( i \) and other frames in the same cluster. \( a_i \) represents the mean of elements of the red part. The black parts indicate the distance between frame \( i \) and frames in other clusters. \( b_i \) represents the minimum value of the mean of other clusters.

4.6 Multi-layer refinement classification

The majority of communication protocols are designed for multi-layers such as internet’s five-layer protocol. In this study, protocol frames were obtained from the link layer. So these frame contains different layers of the protocol, which bring great trouble for the classification. It needs a hierarchical classification. The purpose of classification is to use the same type of data in PRE. Since there is no prior knowledge, it is impossible to get the same data as the protocol specification. We try to design refinement classification and achieve high quality data cluster, which reduces input error in PRE.

The data is from the link layer to the application layer even including a physical layer for data synchronisation. Usually, high-layer protocol types are the most and low-layer types are the least. For example in internet protocol (Fig. 1), there is one kind of protocol in layer 1 while five in layer 3. However, the determination of classification number is a hard problem because layer 1 and layer 2 exist, some frames are likely separated into the same class despite different layer 3. We propose a multi-layer optimal clustering algorithm for the problem.

Step 1: Given a set of binary frames \( X = \{ x_1, x_2, ..., x_n \} \). We compute the pairwise similarity values of the binary frames as described in Section 4.1 to obtain the affinity matrix \( A \). The \( i \)th row and \( j \)th column element of the matrix \( A \) is denoted by \( a_{ij} \).

Step 2: After the affinity matrix obtained by Step 1, we cluster the protocol frames based on the spectral clustering algorithm. Firstly, \( L_{sym} \) is computed by normalised transformation of matrix \( A \). Then the first \( k \) eigenvectors of \( L_{sym} \) are solved. These eigenvectors are constructed into a new matrix \( V \). Finally, the \( K \)-means algorithm is used to cluster the row vectors of \( V \) to obtain the clustering results in one iterative calculation.

Step 3: In order to determine \( k \) automatically, we use the CVI described in Section 4.2 to help determine the number of clusters in the spectral clustering process. The three CVIs are weighted (denoted by \( f_{mul}(\cdot) \)) to determine the excellent CVI value and reduce the one-sidedness of a single index. If the feature vectors are continuously clustered CVI will inevitably have multiple local optimal values to determine different layers of clustering. This also means achieving the goal of multi-layer classification. Since CVI values are discrete, we introduce the concept of stagnation and inflection point like a continuous function to judge CVI and determine \( k \) as (18). Different clustering layers are determined by queue \( K \), where \( k \in K \).

\[ f_{mul}(k + 1) - f_{mul}(k) \leq \lambda \quad \text{and} \quad f_{mul}(k) \geq f_{mul}(k - 1). \]

Therefore, we designed a multi-layer optimal clustering method as shown in Algorithm 1 (see, Fig. 3), where \( \alpha \), \( \beta \), and \( \gamma \) are weights of three CVIs, \( \lambda \) is a balance parameter used for determining the inflection point, \( c \) is the maximum possible value of \( K \).

We can see that the main computational process of Algorithm 1 (Fig. 3) is divided into the calculation of affinity matrix, symmetric matrix eigenvector, row vector clustering and three CVIs in which the runtime complexity of Davies–Bouldin index is higher than that of the other two CVIs. Therefore, the runtime complexity of Algorithm 1 ( Fig. 3) can be obtained by four processes as shown in Table 1. where \( n \) is the number of frames, \( l \) is the average
length of frames, $m$ is the rank of the affinity matrix, $d$ is the average number of non-zero elements in a row of the matrix, $k$ is the number of clusters and $s$ is the number of iterations. So the complexity of the Algorithm 1 (Fig. 3) is $O(l^2n^2)$.

### 5.2 First clustering

To determine the parameters, we set $c = 14$. This value is not obtained based on prior knowledge. However, it can be estimated according to the size of the protocol packets because protocol types should not be large in a small packet. In order to calculate $f_{mul}(k)$, we set $\alpha = 0.5$, $\beta = 0.3$, $\gamma = 0.2$ and $\lambda \leq 0.2$.

According to Table 2, Fig. 4–6 indicate that the first determined cluster number can be obtained. The results show that the correctness of the first determined cluster number is 100%. In Fig. 4, the CVI, $f_{mul}(k)$, is monotonically increased or decreased before the best clustering results achieved. The first clustering number is determined just in the case of the global optimal value. In Fig. 5, the number of clusters is determined based on the first local optimal value. The above two cases are $\lambda > 0$. In Fig. 6, the values of CVI are compared within adjacent clusters, and the number of first clusters is determined if the inflection points are at $\lambda > 0$. In any case, the number of the first clustering can be accurately obtained by CVI values of different cluster numbers.

After determining the number of clusters for the first time, it is important to verify the quality of clusters in detail. We validate our algorithm by ratios of accuracy and recall which are used to evaluate most classification methods. In our algorithm, the frames identified as the same type of protocol in Wireshark are also clustered into the same cluster. Fig. 7 compares the results of our algorithm with a benchmark provided by Wireshark. We find that the ratios of accuracy and recall are above 93% for most data after the first clustering of our algorithm. Some clustering results are inconsistent with the benchmark, such as DarkComet. The 26th and 28th frames are HTTP protocols, but clustering results are inconsistent with the benchmark.

![First determined cluster number via global optimal value](image)

(a) Newcap, (b) Shutdown

Wireshark, which is a useful and open-source tool for grabbing analytical protocol data, to parse and view our data as a benchmark. In the testing process, we treated all data as completely unknown data. The experimental results are compared with benchmark provided by Wireshark to verify the validity of our algorithm. This study mainly focuses on the classification of various protocol frames and does not infer the type name of the protocol. So the test results are only compared with classification results indicated by Wireshark.

### 5. Experimental results and analysis

#### 5.1 Dataset

In order to verify the validity of our algorithm, we have got some real world data from pcapr [36], an open web site, which provides packets for various protocols. We used six datasets to test, as shown in Table 2. Each dataset contains several protocols. We use

#### 5.2 Complexity of each process

| Process            | Algorithm          | Complexity          |
|--------------------|--------------------|---------------------|
| affinity matrix calculation | Smith–Waterman | $O(l^2n^2)$          |
| symmetric matrix eigenvector | Lanczos & QR | $O(\text{dn}^2)$ & $O(\text{m}^2)$, $m < n$ |
| row vector clustering | $k$-means | $O(nks)$           |
| CVIs                | Davies–Bouldin index | $O(k^2n^2)$        |

length of frames, $m$ is the rank of the affinity matrix, $d$ is the average number of non-zero elements in a row of the matrix, $k$ is the number of clusters and $s$ is the number of iterations. So the complexity of the Algorithm 1 (Fig. 3) is $O(l^2 + k + d)n^2 + ksn$. The complexity of Algorithm 1 (Fig. 3) is relatively high, which is mainly reflected in the similarity calculation. However, this study is aimed at the binary data frames of unknown protocol, which must be accurate in the similarity calculation. Therefore, the Smith–Waterman algorithm with high complexity can be adopted. In addition, how to solve this problem is also our follow-up research.

#### 5.3 First clustering

The number of first clustering can be obtained. The results show that the correctness of the first determined cluster number is 100%. In Fig. 4, the CVI, $f_{mul}(k)$, is monotonically increased or decreased before the best clustering results achieved. The first clustering number is determined just in the case of the global optimal value. In Fig. 5, the number of clusters is determined based on the first local optimal value. The above two cases are $\lambda > 0$. In Fig. 6, the values of CVI are compared within adjacent clusters, and the number of first clusters is determined if the inflection points are at $\lambda > 0$. In any case, the number of the first clustering can be accurately obtained by CVI values of different cluster numbers.

After determining the number of clusters for the first time, it is important to verify the quality of clusters in detail. We validate our algorithm by ratios of accuracy and recall which are used to evaluate most classification methods. In our algorithm, the frames identified as the same type of protocol in Wireshark are also clustered into the same cluster. Fig. 7 compares the results of our algorithm with a benchmark provided by Wireshark. We find that the ratios of accuracy and recall are above 93% for most data after the first clustering of our algorithm. Some clustering results are inconsistent with the benchmark, such as DarkComet. The 26th and 28th frames are HTTP protocols, but clustering results are inconsistent with the benchmark.
clustered with TCP frames together. HTTP is the upper application protocol of TCP originally and HTTP frames can be easily clustered into TCP frames set because the similarity among the frames is calculated from the link layer to the application layer. In the previous data link layer, the network layer and transport layer data will affect the results.

5.3 Second clustering

Owing to the layered design of protocol families, we can determine the type of low-layer protocol according to our algorithm after the first clustering. In further clustering, we can get a more detailed protocol type at higher layers. We can illustrate the process of two clusters as shown in Fig. 8. Assuming that the first clustering divides the data into three types, such as A, B and C, there are different similarities of the data in the A class and the second clustering process divides A into A₁ and A₂. However, the different values of the data in B and C are relatively smaller, they remain in the same type. We show the results of Binetravler and Darkcomet by the second clustering in Table 3. We can see that frames are subdivided into more high-layer protocol types by the second clustering. Our datasets are mixed with a variety of frames. Lengths and payloads of frames are not the same. In the second clustering, some frames are further subdivided, and other frames still maintain the same type. Since similarity values of frames are relatively large, they are maintained at the same type from bottom to upper layer of the protocol. No new error clusters had been introduced which makes the protocol clustering maintain the hierarchical characteristics. We also find that the numbers of protocol types are correct after the second clustering (seven as shown in Fig. 5a, 12 as shown in Fig. 6a), which is consistent with Table 3, respectively.

6 Conclusion

This study aims to solve the problem of multi-layer clustering of mixed binary protocol frames. We use the spectral clustering method combined with CVIs to establish the second-order optimisation mathematical model of the problem. Although this model is difficult to be converted to a first-order optimisation model, this study gives a feasible calculation method to calculate the model. Afterwards, we give a concrete algorithm of CVI for the protocol data in the absence of centroid and an interpretation to facilitate calculation. Based on that, a clustering algorithm of mixed binary protocol frames is proposed, which can achieve multi-layers clustering of frames without any prior knowledge. The results of experiments show that our algorithm can work well. Good quality and high similar data sets can be provided to format inference, which is another important task of PRE by multi-layers clustering.
7 Acknowledgments

This work was supported by CAEP Foundation (CX2019040) and the Defense Industrial Technology Development Program (JCKY2018212C020, JCKY2016212C005) and the National Key Research Planning Program (2016YFB08002).

8 References

[1] Helen, J.W., Chuanxiong, G., Daniel, R.S., et al.: ‘Shield: vulnerability-driven network filters for preventing known vulnerability exploits’, ACM Sigcom Comput. Commun. Rev., 2004, 34 (4), pp. 193–204
[2] Patel, S.C., Bhatt, G.D., Graham, J.H.: ‘Improving the cyber security of scada communication networks’, Commun. ACM, 2009, 52 (7), pp. 139–142
[3] Siyu, T., Hongyi, Y., Qing, L.: ‘Bit-oriented format extraction approach for automatic binary protocol reverse engineering’, IET Commun., 2016, 10 (6), pp. 709–716
[4] Narayan, J., Shukla, S.K., Clancy, T.C.: ‘A survey of automatic protocol reverse engineering tools’, ACM Comput. Surv., 2016, 48 (3), pp. 1–26
[5] Sija, B.D., Young-Hoon, G., Shin, K.S., et al.: ‘A survey of automatic protocol reverse engineering approaches, methods, and tools on the inputs and outputs view’, Secur Commun. Netw., 2018, pp. 1–17
[6] Marshall, B.: ‘The protocol informatics project’. Toorcon. Available at http://www.4tphi.net/awalters/PI/PI.html, accessed 2004.
[7] Weidong, C., Kannan, J., Helen, J.W.: ‘Discoverer: automatic protocol reverse engineering from network traces’. USENIX Security Symp., Boston, USA, 2007, pp. 1–14

Table 3  Results of first and second clustering for some datasets

| Datasets     | First clustering | Protocol types | Second clustering | Protocol types |
|--------------|-----------------|----------------|-------------------|----------------|
| Binnettravelr | 3, 4, 5, 6, 7, 8 | IGMPv3         | 3, 4, 5, 6, 7, 8  | IGMPv3         |
|              | 16, 10, 13      | ARP            | 16, 10, 13        | ARP            |
|              | 1, 2, 11, 12, 17| DHCP           | 1, 2, 12, 18      | DHCP ACK       |
|              | 18              |                | 17, 11            |                |
|              | 14, 15, 19, 20  | DNS            | 14, 19, 20        | DNS query      |
|              | 21, 22          |                | 21, 22            | DNS query response 1 |
|              |                  |                | 15               | DNS query response 2 |
| Darkcomet    | 1, 2, 3, 4, 5, 6| IGMPv3         | 1, 2, 3, 4, 5, 6  | IGMPv3         |
|              | 7               |                | 6, 7             |                |
|              | 8, 9, 10, 20, 29| DHCP           | 8, 9, 10, 20, 29 | DHCP ACK       |
|              | 12, 13          | ICMP           | 12, 13           | ICMP           |
|              | 25, 26, 31, 33  | TCP SYN        | 31, 33           | TCP ACK        |
|              | 23              |                | 25, 26           | TCP SYN        |
|              | 17, 18, 19, 21  | ARP            | 17, 18, 19, 21   | ARP request    |
|              | 22              |                | 22              | ARP reply      |
|              | 11, 14, 15, 16  | DHCP           | 14, 16           | DHCP ACK       |
|              | 32              |                | 11, 15           | DHCP request   |
|              | 24, 27, 28, 30  | TCP ACK        | 28, 30           | TCP ACK        |
|              | 27              |                | 24              | TCP SYN, ACK   |

[9] Langner, R.: ‘Stuxnet: dissecting a cyberwarfare weapon’, IEEE Secur. Priv., 2011, 3 (9), pp. 49–51
[10] Goyal, R., Sharma, S., Bevinakoppa, S., et al.: ‘Obfuscation of stuxnet and flame malware’, Latest Trends Appl. Inf. Comput., 2012, 150, p. 154
[11] Glenn, W.M., Martin, C.C.: ‘An examination of procedures for determining the number of clusters in a data set’, Psychometrika, 1985, 50 (2), pp. 159–179
[12] Arbelaitz, O., Gururutxua, J., Muguerza, J., et al.: ‘An extensive comparative study of cluster validity indices’, Pattern Recognit., 2013, 46 (1), pp. 243–256
[13] Chosiuhi, H., Charrad, M., Ghazalli, N.: ‘A comparison study of clustering validity indices’. Computer & Information Technology, Sousse, Tunisia, 2015, pp. 1–4
[14] Erman, J., Mahanti, A., Arlitt, M., et al.: ‘Identifying and discriminating between web and peer-to-peer traffic in the network core’. Proc. 16th Int. Conf. on World Wide Web, Florence, Italy, 2007, pp. 883–892
[15] Erman, J., Mahanti, A., Arlitt, M., et al.: ‘Offline/online traffic classification using semi-supervised learning’, Perform. Eval., 2007, 64, pp. 1194–1213
[16] Jun, Z., Chao, C., Yang, Y., et al.: ‘An effective network traffic classification method with unknown flow detection’, IEEE Trans. Netw. Serv. Manage., 2013, 10 (2), pp. 133–147
[17] Rongqiang, L., Ou, L., Qing, L., et al.: ‘Unknown network protocol classification method based on semi-supervised learning’. IEEE Int. Conf. on Computer and Communications, ChengDu, China, 2015, pp. 300–308
[18] Yu, W., Yang, X., Jun, Z., et al.: ‘A novel semi-supervised approach for network traffic clustering’. IEEE 5th Int. Conf. on Network and System Security, Milan, Italy, 2011, pp. 169–175
[19] Jing, R., Xiaochen, K., Gan, L., et al.: ‘A self-adaptive network traffic classification system with unknown flow detection’. 3rd IEEE Int. Conf. on Computer and Communications, ChengDu, China, 2017, pp. 1215–1220
[20] Yimin, X., Dit-Yan, Y.: ‘Mixtures of arma models for model-based time series clustering’. IEEE Int. Conf. on Data Mining, Maebashi City, Japan, 2002, pp. 717–720
[21] Wanzeng, K., Zhihai, S., Can, Y., et al.: ‘Automatic spectral clustering based on eigengraph and orthogonal eigenvector’, Acta Electron. Sin., 2010, 8, pp. 1880–1885
[22] Zander, S., Nguyen, T., Armitage, G.: ‘Automated traffic classification and application identification using machine learning’. The IEEE Conf. on Local Computer Networks 36th Anniversary, Sydney, Australia, 2005, pp. 250–257
[23] Erman, J., Mahanti, A., Arlitt, M.: ‘QoP5-4: internet traffic identification using machine learning’. IEEE Globecom, San Francisco, USA, 2006, p. 1–6
[24] Bernaille, L., Teixeira, R.: ‘Implementation issues of early application identification’, Lect. Notes Comput. Sci., 2007, 4866, pp. 156–166
[25] Hirvonen, M., Laulajainen, J.P.: ‘Two-phased network traffic classification method for quality of service management’. IEEE Int. Symp. on Consumer Electronics, Kyoto, Japan, 2009, pp. 962–966
[26] Cailong, Y., Fei, W., Benxiong, H.: ‘Internet traffic classification using dbscan’. IEEE Wase Int. Conf. on Information Engineering, vol. 2, Taiyuan, China, 2009, pp. 163–166
[27] Wei, J., Jingbo, X.: ‘Methods for traffic clustering in network virtualization environment’, Electron. Opt. Control, 2015, 22 (8), pp. 54–59
[28] Bianqin, W., Shunhen, Y.: ‘Automatic extraction for the traffic of unknown network applications’, J. Commun., 2014, 35 (7), pp. 164–172
[29] Hongliang, Y., Yong, Z., Gang, X., et al.: ‘Poster: mining elephant applications in unknown traffic by service clustering’. Proc. 2014 ACM SIGSAC Conf. on Computer and Communications Security, Scottsdale, USA, 2014, pp. 1532–1534
[30] Casas, P., Mazel, J., Owezarski, P.: ‘Unsupervised network intrusion detection system: detecting unknown without knowledge’. Comput. Commun., 2012, 35 (7), pp. 772–783
[31] Yu, W., Weiping, W., Dan, M.: ‘Mining user cross-domain behavior patterns for insider threat detection’. Chin. J. Comput., 2016, 8, p. 605
[32] Vidal, R., Favarro, P.: ‘Low rank subspace clustering (LRSC)’. Pattern Recognit. Lett., 2014, 43, pp. 47–61
[33] Jun, W., Daming, S., Danong, C., et al.: ‘Lsr: low-rank-sparse representation for subspace clustering’. Neurocomputing, 2016, 214, pp. 1026–1037
[34] Daming, S., Jun, W., Danong, C., et al.: ‘A global-local affinity matrix model via eigengraph for graph-based subspace clustering’. Pattern Recognit. Lett., 2017, 89, pp. 67–72
[35] Smith, T.F., Waterman, M.S.: ‘Identification of common molecular subsequences’. J Mol Biol, 1981, 147 (1), pp. 195–197
[36] Zhigang, W., Changzhen, H., Ming, H.E.A.: ‘Cyber security datasets research’, Adv. Mater. Res., 2013, 659, pp. 191–195