IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 11, 2021, accepted December 4, 2021, date of publication December 10, 2021,
date of current version December 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3134697

Association Analysis and Identification of Unknown Bitstream Protocols Based on Composite Feature Sets

SHUCHENG WANG[1], **FAN GUO**[1], **YONG FAN**[2], **AND JING WU**[2], **(Member, IEEE)**
[1]Electronic Information School, Wuhan University, Wuhan 430072, China
[2]China Shipbuilding Industry Corporation 722nd Research Institute, Wuhan 430079, China

Corresponding author: Yong Fan (fanyong19932006@163.com)

This work was supported by the Research Fund of China Shipbuilding Industry Corporation 722nd Research Institute (Bitstream-oriented wireless network protocol analysis and recognition technology).

ABSTRACT Concomitant with the rapid development of network communications technology, the analysis of communication protocols has become indispensable in the maintenance of daily network security. Common protocol analysis methods predominantly analyze protocols using known information, such as fixed port numbers; however, these methods have significant limitations. In the current network environment, the proportion of undisclosed protocols is increasing daily, and the information related to such protocols is difficult to obtain and sometimes fails because of the particularity of the unknown protocol format. Therefore, it is crucial to analyze unknown protocols in the context of less prior knowledge. To solve this problem, this paper proposes a novel protocol identification method in which association analysis and identification of unknown bitstream protocols are first carried out based on composite feature sets. Furthermore, data mining and statistics-related knowledge are applied to realize protocol message-type identification and protocol message-format analysis. The results of experiments conducted on the bitstream protocol dataset verify that the proposed method can accurately identify different message types. Specifically, taking the ICMP and ARP protocols as examples, the proposed method could effectively infer the main features, which is helpful for further protocol information extraction and analysis.

INDEX TERMS Composite feature set, network communication technology, protocol message-format analysis, protocol message-type identification.

I. INTRODUCTION

A. MOTIVATION AND BACKGROUND

With the rapid development of communication and network technology, network security protection and maintenance have corresponding become increasingly more crucial. Analysis of network protocols is the basis and premise of information security [1]. In the electronic information warfare environment, the common method to acquire information from a target network is to capture its communications signal and then analyze the acquired bitstream protocol data to obtain any desired intelligence. Therefore, it is important to analyze and recognize unknown bitstream protocols from captured communication data; however, there is a dearth of efficacious research in this area.

Early methods were often realized by identifying fixed features of the protocol, with the core idea of such methods being to use static features for matching [2]. Protocol identification technology based on port number is the earliest studied protocol identification method. For traditional Internet protocols, the use of port numbers for protocol identification is characterized by high accuracy and efficiency [3], [4]. However, with the continuous development of the Internet, an increasing number of new protocols have emerged, some of which use registered ports and dynamic ports. To solve the limitations of the above methods, protocol analysis and identification are used to analyze protocol data at the application layer according to the similarity between protocol data [5], and the analysis of such protocol data generally requires certain prior knowledge [6].

The identification of protocols at the application layer depends on their unique features, which are compared with

The associate editor coordinating the review of this manuscript and approving it for publication was Paulo Mendes.

prior knowledge; however, this technique has significant limitations [7]. In an electronic information warfare environment, the target network typically uses wireless communication for data transmission, and most of the communication protocols are customized [8]. It is difficult to obtain relevant information on unknown protocols to analyze them. Therefore, protocol identification and analysis against a background of zero knowledge is an important research topic in the field of network security and electronic information countermeasures.

B. PROBLEM STATEMENT

Protocol reverse engineering technology in the field of unknown protocol identification primarily includes application-based, execution-trace-based, and network-trace-based methods [9]–[12]. The shortcomings of existing protocol reverse engineering methods are summarized in Section II of this paper.

There is currently a distinct lack of knowledge in this area. First, existing algorithms generally require prior knowledge of the protocol data as data input. Second, the data size of the protocol analysis is large and complicated. The analysis of unknown protocols depends on relatively complete datasets, and also introduces the problem of excessive computation. Third, the proportion of manual analyses is too large. Fourth, most of the existing research on protocol reverse engineering is focused on the unknown protocol of the application layer, with less focus being placed on the unknown bitstream protocol in the data link layer.

C. CONTRIBUTION

To solve the existing problems, this paper proposes a method for the association analysis and identification of unknown bitstream protocols based on composite feature sets, introduces data mining and statistics-related knowledge, and realizes protocol message-type identification and protocol message-format analysis. This study focuses on the following aspects:

1) Composite feature sets of unknown protocols were obtained in this study and a feature library constructed. Some unknown protocols located in the link layer have a feature library and, in their protocol recognition, the fields in the feature library are searched and matched using a pattern-matching algorithm. However, most of the unknown protocols have no public feature library, and protocol information cannot be obtained. Therefore, this study imitates the feature library of the protocol of the part link layer, establishes the feature library for an unknown protocol, and completes the protocol recognition based on the feature library.
2) For feature extraction, an improved FP-growth (Frequent Pattern Growth) algorithm, which is a commonly used algorithm in association rules, is used. Based on the FP-growth algorithm, this study improves the frequent pattern tree to effectively reduce the size of the tree, and the frequent items are effectively further filtered to reduce many redundant false rules produced when the system adopts the FP-growth algorithm.
3) Compared with previous research, this research uses a sequence alignment algorithm in bioinformatics to analyze the protocol message format, which can not only identify the address field, but also identify other field information in the message format of the protocol, including the length, check, and sequence number fields.

The remainder of this paper is organized as follows. In Section II, related work is introduced. In Section III, the entire method is introduced, including the overall framework. Then, the composite feature extraction, message-type recognition, and protocol format analysis are introduced. Finally, in Section IV, the experimental analysis is discussed and Section V presents the conclusions of this study.

II. RELATED WORK

Early manual protocol reverse engineering methods can extract all elements of the protocol structure clearly; however, these methods time-consuming and error-prone [13]. Furthermore, they are not compatible with the rapid increase in the number of new applications and the vast amount of traffic in today's network environment [14], [15]. Therefore, automatic protocol reverse engineering methods have been proposed to address these problems. Automatic protocol reverse engineering approaches can be divided into three categories: application-based, execution-trace-based, and network-trace-based methods.

The application-based approach uses program binaries or source codes. Caballero and Song [16] proposed a novel approach for automatic protocol reverse engineering based on a dynamic program binary analysis. In practice, however, it is difficult to obtain program binaries or their source codes. The execution-trace-based approach must set up an execution monitoring system to keep track of how the program handles messages of unknown protocols. This approach is only made possible by acquiring a program that uses an unknown protocol [17]. However, access to the program of an unknown protocol is rarely possible because of concealment and obfuscation undertakings.

Compared to the above two approaches, the network-trace-based approach is more realistic because it only analyzes the network traces captured by network packets that monitor the target protocol without accessing the program binaries. The primary research in this paper is based on network trace-based methods, which include natural language processing, bioinformatics, and data mining. The natural language processing method identifies protocol keywords by looking for tags that frequently appear together in messages [18], [19]. However, because binary protocols typically pack data more densely, this method is not suitable for inferring binary protocol information. Based on bioinformatics, Netzob [20] has been used for sequence alignment to determine the similarity of messages and to cluster them. The messages were divided into fields; however, multi-sequence alignment is exponentially complex because sequence alignment algorithms always use only two messages as inputs simultaneously [21].

In contrast to sequence alignment in bioinformatics, data mining techniques may use all messages as inputs simultaneously. In addition, it is vital to know how to optimize the results, so that the results are intuitive and clear. Common data mining algorithms include classification, clustering, and association rule algorithms. For supervised learning, the classification algorithm needs to know some prior knowledge of the classification model. Common classification models include the Bayesian method [22], genetic algorithm, decision tree algorithm [23], [24], and neural networks [25].

Protocol recognition based on classification algorithms requires a known protocol class; however, it is difficult to identify unknown protocols. The clustering algorithm requires no prior knowledge and can directly calculate the original sample data [26]. Association rule mining is one of the most mature, important, and active research topics in data mining. Liu *et al.* [27] used the Apriori algorithm to extract protocol keywords from network traces based on their support rates and variances of positions, reconstructed message formats, and inferred protocol state machines. Ji *et al.* [28] used a multipattern-matching algorithm to find frequent sequences, extracted keywords based on frequency, and extracted message formats using FP-growth.

The network trace-based approach mentioned above can only analyze the application layer protocol, whereas the unknown bitstream protocol is located in the data link layer. There is little prior knowledge of unknown bitstream protocol data, and several link layer protocol identification problems need to be solved in commercial applications or electronic information warfare. There are no significant achievements in the analysis of such protocols in current studies. Zhang *et al.* [29] only recognized the address field in the message format, and Zheng [30] could not identify the message types of the protocol.

In contrast to other studies, this study proposes association analysis and identification of unknown bitstream protocols based on composite feature sets, which uses an association rule algorithm to identify and analyze protocols based on composite feature sets. It can identify the message types of the protocol and analyze the message format, including the address field, length field, and verification field.

III. METHOD

This section discusses the traits of the unknown bitstream protocol, proposes an identification and analysis method for the unknown bitstream protocol, and describes the process of composite feature extraction, message-type identification, and message-format analysis.

A. GENERAL FRAMEWORK

The steps in the analysis and identification of the unknown protocol mainly include three parts: composite feature extraction, message-type identification, and message-format analysis. The core of the unknown protocol recognition model is shown in Fig. 1. First, the unknown bitstream protocol data are used as input. The features constructed by frequent sequences and their offset positions are considered as composite features. After extracting the compound features of the protocol, a feature library is then built. Then, based on the composite feature set in the feature library, the clustering algorithm is used for protocol message clustering, and the single message type of the protocol data is obtained. A message-format analysis is then performed. The ClustalW [31] algorithm is used to infer the message format of the protocol. Finally, the different fields of the message format are obtained, and the field identification results are taken as the output.

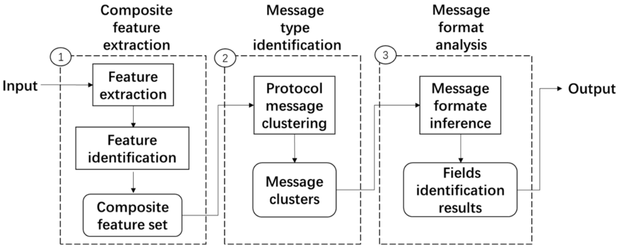

Composite feature extraction
Message type identification
Message format analysis
1
2
3
Input
Feature extraction
Feature identification
Composite feature set
Protocol message clustering
Message clusters
Message formate inference
Fields identification results
Output

FIGURE 1. Basic process of recognition for unknown protocol.

B. COMPOSITE FEATURE EXTRACTION

Compared with text data, bitstream protocol data have a single data form with only two values of 1 and 0, which makes it difficult to obtain semantic information. In addition, the offset position information should be considered in the composite feature extraction. In the context of bitstream protocol recognition, the frequently occurring sequences and their offset positions are considered as the compound features of the protocol. The composite feature extraction method proposed in this paper is shown in Fig. 2. It comprises feature extraction, feature identification, and the construction of the feature library.

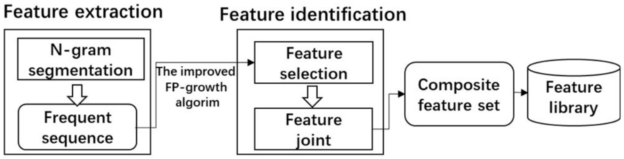


FIGURE 2. Proposed composite feature extraction method.

1) FEATURE EXTRACTION

a: N-GRAM SEGMENTATION

First, the data are divided, and then frequency statistics analysis of the divided unit length sequence is carried out. The final purpose is to analyze the unit length sequence. The main implementation principle of the N-gram model is based on string statistics, where N denotes the length of the unit string of segmentation, whose value has a significant impact on the effectiveness and integrity of the algorithm. For bitstream data of length m, $C_1, C_2, \ldots, C_m$, the N-gram model is used for segmentation, and the partition length is N bytes. The size of N directly affects the accuracy and efficiency of feature extraction; therefore, this study utilizes Zipf's law as the basis for selecting the N value [32].

b: FREQUENT SEQUENCE

This study analyzed the traits of the bitstream protocol. From a statistical point of view, the feature of the protocol bitstream sequences contains two attributes; one is that attributes occur frequently, and the other is that features must have a specific meaning. Therefore, the segmented sequences of the protocol units are not all protocol features. The position information in the protocol is also an important piece of information that can be used for feature extraction. The frequent sequence and the feature pair constructed by the frequently occurring sequence, as well as its offset position, are taken as the protocol composite features.

$$T_{l-s} = \{T_1, T_2, \ldots, T_n\} \tag{1}$$

Therefore, T_{l-s} denotes the protocol set with the same offset position sequence unit, l denotes the offset position, and s denotes the frequent sequence.

2) FEATURE IDENTIFICATION

The redundancy of data segmentation by the N-gram model is too high. The features of the protocol should be more frequent and have specific meanings. Therefore, this paper proposes the concept of feature selection and feature joints to filter and splice the sequence of compound elements.

a: FEATURE SELECTION

In feature selection, an improved FP-growth algorithm is proposed. Based on the FP-growth algorithm, the frequent pattern tree is improved, which effectively reduces the size of the tree, and reduces the storage space of the system; the search space of the algorithm is also effectively compressed, and the frequent items are further filtered to reduce a large number of redundant false rules produced by the system when adopting the FP-growth algorithm.

The association rules of frequent sequences were mined using the improved FP-growth algorithm. Let I $= \{i_1, i_2, \ldots, i_n\}$ be a set of n feature sequences, and i_n describes the ith feature sequence. The association rule is defined as an implication of the form: X $\Longrightarrow$ Y,Where X is called antecedent or left-hand-side (LHS) and Y is called consequent or right-hand-side (RHS).

Support is the probability of X and indicates the frequency of the frequent sequence.

$$\mathrm{sup} = P(X) \tag{2}$$

Confidence is the conditional probability P $(X \mid Y)$ and is an indication of how often the rule has been found to be true.

$$\text{Conf}\,(\text{X} \Longrightarrow \text{Y}) = \frac{sup\left(X \bigcup Y\right)}{sup\,(X)} \tag{3}$$

The lift of a rule is the ratio of the observed support to the expected value if X and Y are independent.

$$\text{lift}\,(\text{X} \Longrightarrow \text{Y}) = \frac{sup\left(X \bigcup Y\right)}{sup\,(X) \times sup\,(Y)} \tag{4}$$

The viction of a rule is defined as follows. This can be interpreted as the ratio of the expected frequency that X appears without Y.

$$\text{conv}\,(\text{X} \Longrightarrow \text{Y}) = \frac{1 - sup\,(Y)}{1 - conf\,(\text{X} \Longrightarrow \text{Y})} \tag{5}$$

In the improved FP-growth algorithm, after the construction of the FP-tree, FP-tree mining is performed. Starting with the frequent pattern of length 1, a conditional pattern base is constructed. Then, the FP-tree is constructed and recursively digs into the tree. Pattern growth is achieved by linking the postfix pattern to the frequent pattern generated by the conditional FP-tree. This algorithm establishes FP-tree by scanning the frequent sequence database, explains the association between frequent sequences, and filters out infrequent sequences through min_sup (The Minimum Support Value) [33].

b: FEATURE JOINT

After the feature selection, the length of the bitstream protocol features is not necessarily fixed, and the filtered composite unit sequence also needs to be spliced according to the position relationship. In the splicing process, association rules are used to determine the possibility of splicing. The splicing of the frequent sequence is completed according to the algorithm flow of the association rules. By analyzing the position difference between unit field sequences and according to association rules, the following definitions are provided during stitching:

Definition 1: Offset position of sequence P_i. The offset length bit that conforms to the first character in the sequence P_i from the first part is defined as the position of P_i, POS (P_i). After adding the offset position limitation, the Pi of the same sequence is treated as a different sequence owing to its different positions.

Definition 2: The expression P⇒P' is introduced. In the context of this paper, the correlation between two sequences P and P' satisfies POS(P)<POS(P'), with the position of P' in the position following P.

Definition 3: Splicing Confidence For P⇒P', the definition of confidence is changed in the context of this paper to the conditional probability of the subsequent occurrence in the adjacent positions under the condition of the presence of the leader, where the length of the leader P is expressed by Len (P).

$$\text{conf}\left(\text{P} \Longrightarrow \text{P}'\right) = \frac{sup\,(len(P))}{min\,(sup\,(P)\,, sup\,(P'))} \tag{6}$$

The input of the feature joint is the segmented and filtered composite feature, and the minimum confidence threshold. The composite feature contains two parts of information, namely: the frequent sequence and the offset position. According to the above definitions of association rules, if the confidence of two composite feature sequences P and P' is greater than the threshold value, then the association rule between them is valid, and the two are spliced into a long string. Finally, the composite feature set obtained after feature selection and the feature joint is put into the feature library.

C. MESSAGE-TYPE IDENTIFICATION

Each protocol typically contains a sequence of messages. Each message has a message type. After building the feature library in the previous section, the message type of the protocol needs to be recognized based on the composite feature set in the feature library. By extracting the protocol features from the feature library, a clustering algorithm is used to identify and determine the Dunn index of the protocol features [34]. After the vectorization of protocol data is completed, it is used as a variable to complete clustering by setting the number of different message types K, and the Dunn index is introduced to select the final K value to complete the differentiation of protocol message types. Finally, a single message type is obtained.

1) PROTOCOL VECTORIZATION

The variable selection of the clustering algorithm is generally performed in two ways. The first method involves the direct selection of continuous attributes. The second method is for some attributes that can only be represented by "have" or "none," corresponding to "1" or "0" respectively. Based on the compound features selected in this study, the second method was chosen to determine the variables of the protocol data frame. Use "1" or "0" to indicate whether the composite features appear or not, mark the composite features, and finally select the variables to complete the clustering.

When the vectorization operation is performed on the protocol data frame, the offset position in the composite feature is used to perform sequence alignment on the corresponding position of the protocol. For each composite feature, the value is assigned according to whether the corresponding offset position of the composite feature appears in the protocol data frame to obtain the vector $M = \{m_1, m_2, \ldots, m_n\}$, where the value of m_i can only be 0 or 1.

2) THE MEASURE OF SIMILARITY

After vectorization of the protocol data frames is completed, an appropriate similarity measurement method is selected. In this study, the Jaccard distance was chosen as the similarity measure for the two data frame vectors. When Jaccard's similarity coefficient is used, the processing object is usually a binary variable, without considering the size of the actual value, and the calculation efficiency is high [35].

$$\text{Jaccard}\,(\text{A, B}) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{7}$$

3) CLASS CLUSTER EVALUATION

The Dunn index is also used to select the most reasonable clustering result to obtain the number of message types and is defined as follows:

$$\text{D}\,(K) = \frac{min_{1 \le i \le K} \left\{\delta\left(C_i, C_j\right)\right\}}{max_{1 \le i \le K} \left\{\Delta\left(C_i\right)\right\}} \tag{8}$$

where C_i and C_j are any two clusters in the clustering result, $\Delta(C_i)$ represents the furthest distance between samples in the cluster of class cluster C_i, and $\delta\left(C_i, C_j\right)$ represents the distance between the two clusters of C_i and C_j.

The goal of message clustering is to assign a type to each message. To this end, a metric of similarity is defined between messages and is used to cluster similar messages together. Once all similar messages are clustered, each cluster (and all the corresponding messages) is labeled with a type. As shown in Fig. 3, the message type in message cluster 1 is one type, and there are a total of K message clusters. It is known from the previous section that there are a total of K different types of messages.

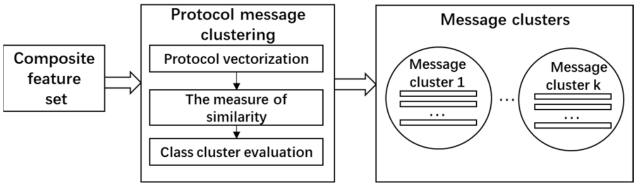


FIGURE 3. Overall process of protocol message-type identification.

D. MESSAGE FORMAT ANALYSIS

The message type is defined by a message-format specification. The message format specifies the structure of a message, typically in a number of fields. After the message-type identification of the protocol in the previous section and obtaining the single message type, relevant information needs to be extracted from the protocol by analyzing the message format of the protocol. As shown in Fig. 4, on the premise that message clusters are taken as input, a multi-sequence contrast algorithm and information entropy correlation theory [36] are used to protocol alignment and field partitioning to infer message formats, including fixed-length and variable-length fields. The ClustalW algorithm is used to identify the different fields of the protocol message format and divide the length, address, sequence number, and check fields of the protocol. This method divides the field area of the protocol under the condition of less prior knowledge.

1) MESSAGE FORMAT INFERENCE

The fields of the message format mainly include fixed-length and variable-length fields and each field is either a fixed

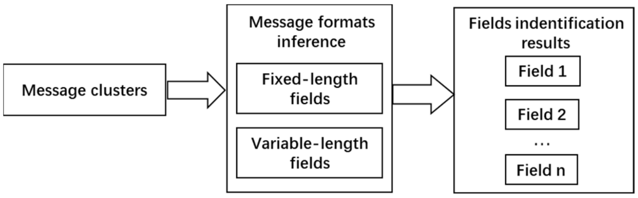


FIGURE 4. Overall process of message-format analysis.

length or variable length. The length value of a fixed-length field is static, and it does not change across multiple instances of the same field. The length value for a fixed-length field is part of the protocol specification and is known a priori for the implementation of the protocol. In contrast, the length of a variable-length field is dynamic; that is, it can change across multiple instances of the same field. The message-format analysis in this study focuses on variable-length fields.

The protocol region should be divided to determine the length of the field. Taking byte as the minimum division unit, the bytes of the same field have certain similarities in the statistical law. The value of each position is regarded as a discrete random variable, and its value is not unique. According to relevant statistical knowledge, information entropy can be used to represent the distribution relationships among variables. This study used the statistical distribution of different bytes as the standard for the region merging between bytes according to the offset position. Finally, according to the correlation coefficient between different bytes, the region of the field was divided.

2) FIELD IDENTIFICATION RESULTS

The data input is the protocol data with a single message type. By distinguishing the length and region of the fields of the same message-type protocol data, a message-format analysis of this message type is realized.

With less prior knowledge, the offset position was used to identify the address fields. This method is defined as follows:

Definition 1: Address the field candidate set. The set $\mathrm{U}(L_i) = \{S_1, S_2, \ldots, S_n\}$ is defined as the set of all sequences at the offset L_i.

Definition 2: Similarity coefficients $\mathrm{Sim}\left(L_i, L_j\right)$ was defined to represent the similarity of sequences at two offset positions.

$$\mathrm{Sim}\left(L_i, L_j\right) = \frac{\left|U(L_i) \cap U(L_j)\right|}{\left|U(L_i) \cup U(L_j)\right|} \tag{9}$$

The entire protocol dataset can be regarded as a two-dimensional matrix of n*m field regions, where n is the number of protocols and m is the number of field regions. After the two-dimensional matrix of the field area is obtained, the set of address fields at each position is obtained. Finally, the threshold of the address field is set as follows: if the similarity coefficient is greater than the address field threshold, show in two positions of sequence where the similarity is higher, the collection of the set of similarity coefficients is higher fields, as the address field of the optimal solution.

Identification flowchart of other variable-length fields is shown in the Fig. 5. Other variable-length field identification involves extracting the value of the same variable-length field, arranging the time sequence of these data frames, and counting the change rule of the value. When a field is related to the length of the data frame, it is recognized as a length field; when the value of a field is increasing or decreasing, it is identified as a sequence number field, and when the value types of information entropy and bytes exceed the threshold value, they are identified as check fields.

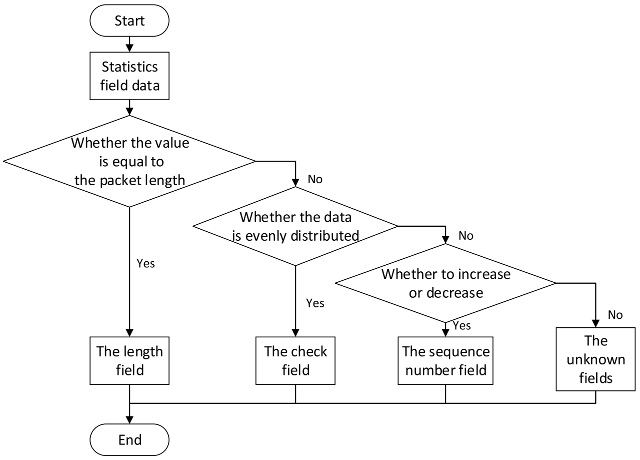

Start
Statistics field data
Whether the value is equal to the packet length
No
Yes
Whether the data is evenly distributed
No
Yes
Whether to increase or decrease
No
Yes
The length field
The check field
The sequence number field
The unknown fields
End

FIGURE 5. Identification flowchart of other variable-length fields.

IV. EXPERIMENT AND RESULT

To express the accuracy of the unknown bitstream protocol identification method, appropriate protocol datasets and design experimental evaluation metrics were selected and the parameter selection and algorithm effect of the above algorithm were verified. In Section IV, the unit segmentation length and frequent sequence filtering threshold are first determined, then experimental indexes are designed to analyze the clustering results, and finally, the field identification results of protocol message formats are presented.

A. PROTOCOL DATA AND DEVELOPMENT ENVIRONMENT

Two types of datasets, both of which were captured using the Wireshark software, were used in this study—namely, the ICMP and ARP datasets. The required configuration and relevant environment for algorithm realization in this study were as follows: The integrated development environment was IDEA with PyCharm. Java and Python were used to realize the algorithms used in this study. The operating system was Windows 10 on an Intel (R) Core (TM) I5-8250U CPU @1.60 GHz.

B. EVALUATION METRICS

In this study, the datasets were used to conduct protocol analysis so that the classification effect could be verified after

TABLE 1. Indicator definition table.

	Positive class	Negative class [a]
Recognize as a positive class	TP	FP
Recognize as a negative class	FN	TN

the completion of association rule classification. First, the following indicators are introduced. Protocols that belong to such a cluster are called positive classes, and protocols that do not belong to such a cluster are called negative classes.

Precision Rate: This is used to characterize the proportion of the actual positive class in the instance that is classified into the positive class.

$$\mathrm{PR} = \frac{TP}{TP + FP} \times 100\% \tag{10}$$

where TP is the number of protocols belonging to such a cluster and FP is the number of protocols that do not belong to such a cluster.

Recall Rate: This indicates that there are multiple positive classes that are classified into positive classes.

$$\mathrm{RR} = \frac{TP}{N} \times 100\% \tag{11}$$

N is the total number of protocol frames.

C. EXPERIMENTAL RESULT

1) SEGMENTATION UNITS

The protocol data were first divided according to the unit length. In this study, the N-gram model was used for the protocol segmentation. In the process of the experiment using two datasets, in bytes for the basic unit, different lengths were selected to shard and unit series frequency statistics were completed at the same time. Then, natural numbers were used to sort by frequency to obtain the ranking. Finally, the logarithm of the frequency was used as the ordinate, and the logarithm of rankings as the ordinate to draw a line chart. If a certain segmentation length made the result conform to the Zipf distribution, this would indicate that the segmentation length was reasonable.

The Zipf distribution curve of the ICMP protocol dataset is shown in Fig. 6; it varies with the value of the unit length n. When n = 1, the curve exhibits the smallest fluctuation and, with an increase in the value of n, the fluctuation of the curve increases. Therefore, for the ICMP protocol, one byte was selected as the unit segmentation length in this study. The Zipf distribution curve of the ARP protocol data is also shown in the figure; the unit division length is 1.

2) SCREENING THRESHOLDS

Following completion of the unit segmentation of the protocol data, the unit sequences need to be screened to filter out data with a lower frequency. The method proposed in this paper is to randomly divide the unit sequence set into two subsets, A and B, set different screening thresholds, and compare the similarity of the two subsets A and B after filtering the lower frequency sequence. Section III introduced the improved FP-growth algorithm and the minimum support value. In this experiment, min_sup was used as the screening threshold. As shown in the figure, the screening threshold that makes the two sets most similar for the first time is the final threshold. With an increase in the threshold, the similarity between the two sets increases gradually after filtering this type of sequence. The frequency with the highest similarity between the two subsets for the first time was selected as the screening threshold. As shown in Fig. 7, for the ICMP protocol, the filter threshold was 0.016, whereas the ARP protocol threshold was 0.019.

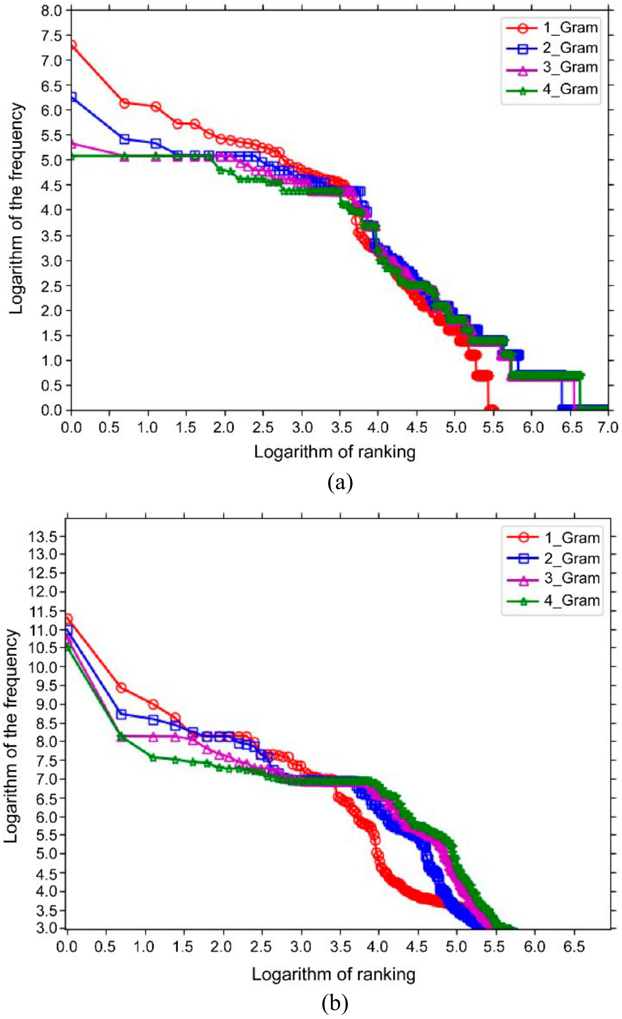


FIGURE 6. Two protocol unit sequence statistics. (a) ICMP. (b) ARP.

3) NUMBER OF MESSAGE TYPES

The physical meaning of the Dunn index is the ratio of the minimum value between any protocol feature set and the maximum value within all protocol feature sets. The larger the

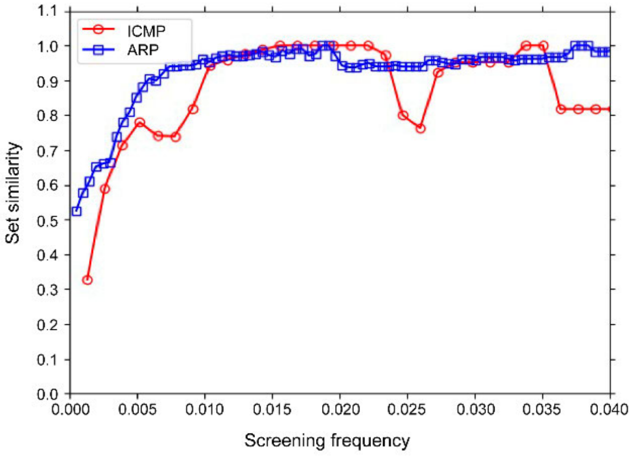


FIGURE 7. Protocol frequent sequence screening threshold.

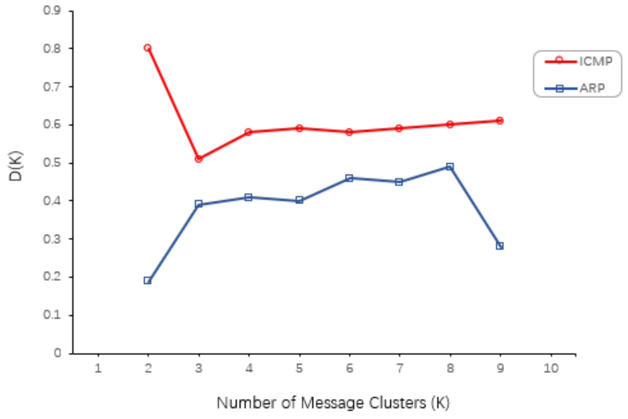


FIGURE 8. Dunn coefficient line chart.

value, the better is the clustering effect. The Dunn coefficients for different message clusters are shown in the figure.

As can be seen from Fig. 8, for the ICMP protocol, when the number of message clusters is two, the Dunn coefficient in the clustering result reaches the maximum. The hierarchical clustering protocol message data of all clusters were compared with the actual data of the dataset. Among them, the data in message cluster 1 belong to the inquiry message, and the data in message cluster 2 belong to the error report message. The number of corresponding messages in each cluster and the number of actual message protocol data frames were counted, and the precision and recall rates were both 100%, which indicates that the proposed algorithm can distinguish the ICMP protocol well. Similarly, the Dunn coefficient under different class clusters of the ARP protocol dataset was calculated. When the number of message clusters was eight, the Dunn coefficient reached its maximum value, as shown in the figure. The average recall rate and average precision rate for this agreement were 96.5% and 100%, respectively.

By analyzing the results of the two datasets, the compound feature extraction method demonstrated that it could effectively extract compound features that can identify different types of protocols. The improved FP-growth algorithm can classify different message types of protocols.

TABLE 2. Field identification instance table.

Offset position	Field values/frequency			Offset position entropy
0	208/0.54	100/0.4	24/0.06	0.376
1	199/0.54	110/0.4	240/0.06	0.376
2	192/0.54	105/0.4	228/0.06	0.376
3	62/0.54	238/0.4	15/0.06	0.376
4	237/0.54	111/0.4	136/0.06	0.376
5	122/0.54	159/0.4	13/0.06	0.376
6	100/0.6	208/0.4	-	0.292
7	110/0.6	199/0.4	-	0.292
8	105/0.6	192/0.4	-	0.292
9	238/0.6	62/0.4	-	0.292
10	111/0.6	237/0.4	-	0.292
11	159/0.6	122/0.4	-	0.292
12	8/1	-	-	0

4) FIELD IDENTIFICATION RESULTS

The protocol data of two protocol datasets were selected, and the connection between different bytes was analyzed using information entropy to obtain the specific region segment of the protocol. Then, the protocol format was inferred from the fixed-length fields and variable-length fields according to the statistical characteristics of different fields. We took the first 13 bytes of ICMP as an example to calculate the protocol data statistics and information entropy distribution.

In the first 13 bytes of ICMP data, the first six bytes have the same entropy value, the same field value, and corresponding frequency. Therefore, the first six bytes can be divided into one field. Similarly, 6 to 11 bytes can also be identified in the same field.

Table 3 compares the address fields recognized by the ICMP protocol and the actual address fields when different thresholds are set.

TABLE 3. Address field identification schematic table.

Offset position	Field values/frequency		Offset position entropy
0.6	[0-5,6-11] [26-29, 54-57] [26-29, 58-61]	[26-29, 30-33] [30-33, 54-57] [30-33, 58-61]	[0-5, 6-11]
0.7	[0-5, 6-11]	[26-29, 30-33]	[26-29, 30-33]
0.8	[0-5, 6-11]	[26-29, 30-33]	
0.9	[0-5, 6-11]	[26-29, 30-33]	
1.0	0-5, 6-11		

As can be seen from Table 3, when the similarity threshold is low, all the address fields cannot be effectively recognized. With an increase in the threshold, all the address fields can be recognized, and the recognition rate reaches 100%. However,

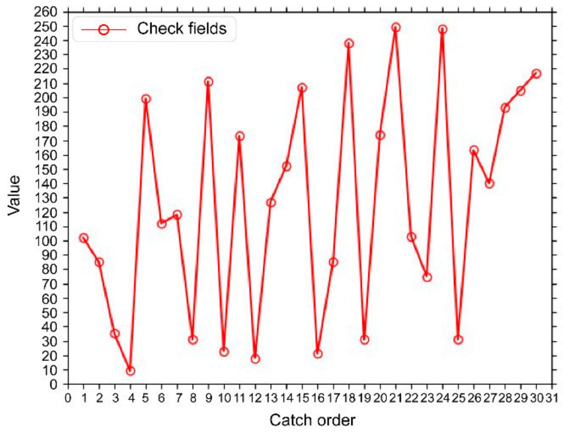


FIGURE 9. Check fields statistics result graph.

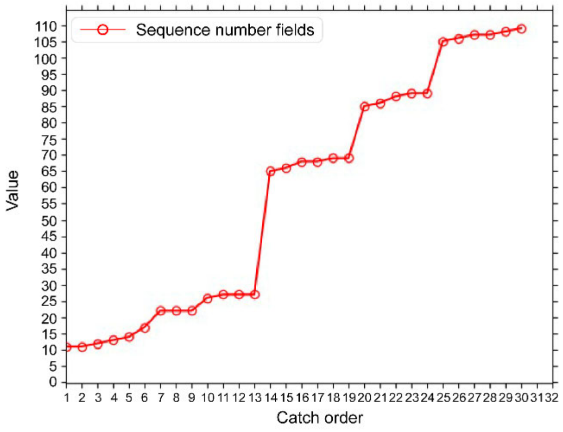


FIGURE 10. Sequence number fields statistics result graph.

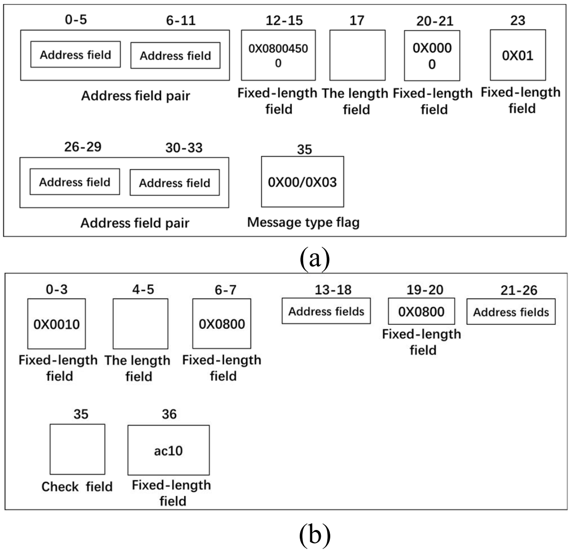


FIGURE 11. Protocol field identification results. (a) ICMP. (b) ARP.

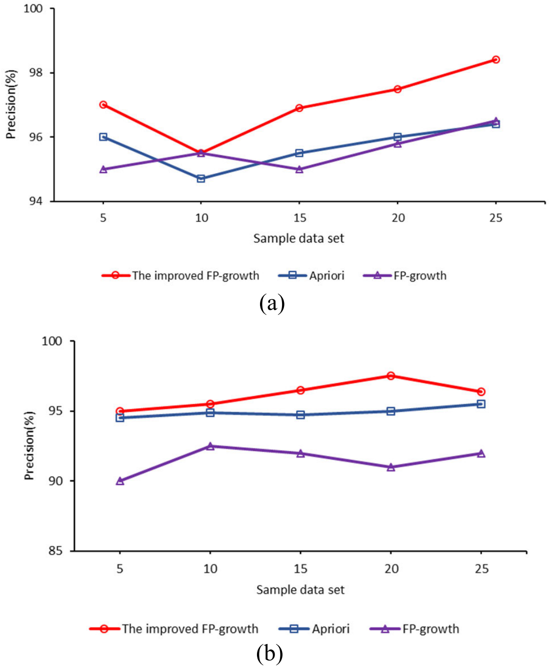


FIGURE 12. Comparison of precision and recall rates of the algorithm. (a) Precision. (b) Recall.

when the threshold is set to 1.0, the similarity requirement is too high, and more complex IP addresses cannot be identified.

The variation in the check fields is shown in Fig. 9. The value of the check fields was irregular and evenly distributed, and the entropy value was large. The changes in the sequence number fields are shown in Fig. 10, and the value of the sequence number field is arranged in an increasing or decreasing order.

The field identification results are presented in Fig. 11. In Fig. 11(a), two pairs of address fields, as well as the corresponding fixed-length field, are correctly identified in the protocol. However, for the length field, the offset in the actual format was 16–17. In the actual format, the offset position of the message-type identifier field ranges from 34 to 35. The main reason for this is that the value of the message-type identifier field is too high, resulting in inconsistent entropy values of the two bytes. In Fig. 11(b), for the control frame structure, the length is generally fixed, and the immutable feature fields of the head and tail of the frame are considered as the leading and ending frame markers. In this study, Ox0800 was determined as a fixed-length field, which is the network number in the actual frame format. It is inferred that the reason for the error is that there are fewer samples in the protocol dataset, and the network number communicating in the same network cannot be effectively identified. For the length fields, the actual format should be two independent length fields, and because the length of the control frame is fixed, the entropy values of the two fields are the same; thus, it is misjudged as one.

D. COMPARISON WITH EXISTING METHODS

1) ALGORITHM COMPARISON

To verify the effectiveness of the improved FP-growth algorithm, the performance of the message recognition method in this study was compared with that of the Apriori and FP-growth algorithms. As shown in Fig. 12, the FP-growth

TABLE 4. Transaction database D list.

Transaction	Frequent items
1	f, a, c, d, g, l, m, p
2	a, b, c, f, l, o
3	d, f, h, j, m, p
4	b, c, k, m, o, s
5	a, f, c, e, l, n, o, p

TABLE 5. Comparison of number of frequent pattern tree nodes of the two algorithms.

Frequent items	Number of nodes in the FP-growth algorithm tree	Number of nodes in the improved FP-growth algorithm tree
1	18	10
5	16	11

algorithm has the lowest efficiency and the proposed algorithm has a high precision rate and recall rate and can achieve the distinction of protocol message types.

The performance of the improved FP-growth algorithm was also compared with that of the original FP-growth algorithm. The two algorithms mine association rules for transaction database D (as shown in Table 4) to discover frequent sets, as shown in the figure, by scanning 1–5 transactions in the database.

By comparing the examples of the two algorithms above, it can be seen that the improved FP-growth algorithm effectively reduces the size of the tree when building the frequent pattern tree (as shown in Table 5). Consequently, the corresponding system storage space is also reduced, and the search space of the algorithm is also effectively compressed.

2) PRECISION AND RECALL

In the experimental section, the proposed protocol identification results were compared with the method proposed by Zhang *et al.* [29]. Zhang's method is also used for unknown bitstream protocols, which discovers protocols by frequent sequences and positions based on clustering and detects address fields based on the similarity of the unit set in different positions.

Tables 6 and 7 provide details of the frequent sequences and locations of the ICMP and ARP protocols. Frequent locations are listed in descending order of frequency. The frequencies of positions (12), (16), (20), (23), and (34) in Table 1 and positions (12), (28), and (38) in Table 2 were all observed to reach 1. These common sequences are the keywords used in the protocol and the experiment was conducted based on frequent sequences and positions whose frequencies did not achieve 1.

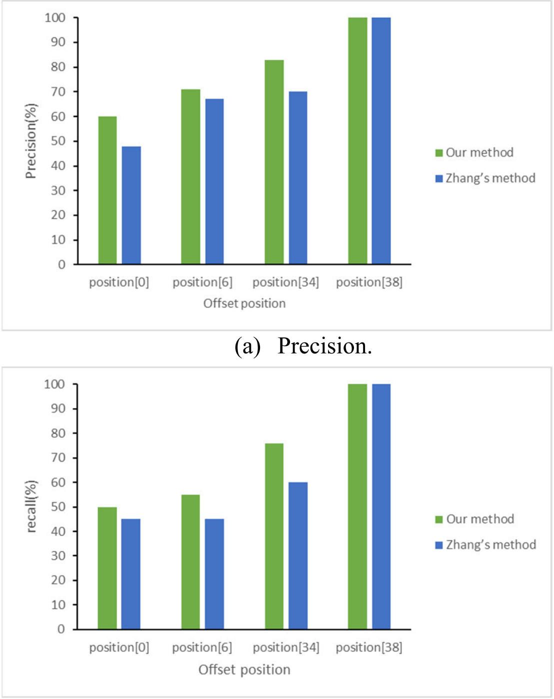
Figs. 13 and 14 show the precision and recall rates of the two methods for different positions in the ICMP and ARP frequent sequences. In Fig. 13, both methods find ICMP

TABLE 6. ICMP protocol frequent sequence and position.

Position	Frequency	Sequence
(12)	1	080045
(20)	1	0000
(34)	0.998	0800
(38)	0.978	0000000000000000000000 0000000000000000000000
(0)	0.841	0010
(16)	1	00
(23)	1	01
(6)	0.851	0010

TABLE 7. ARP protocol frequent sequence and position.

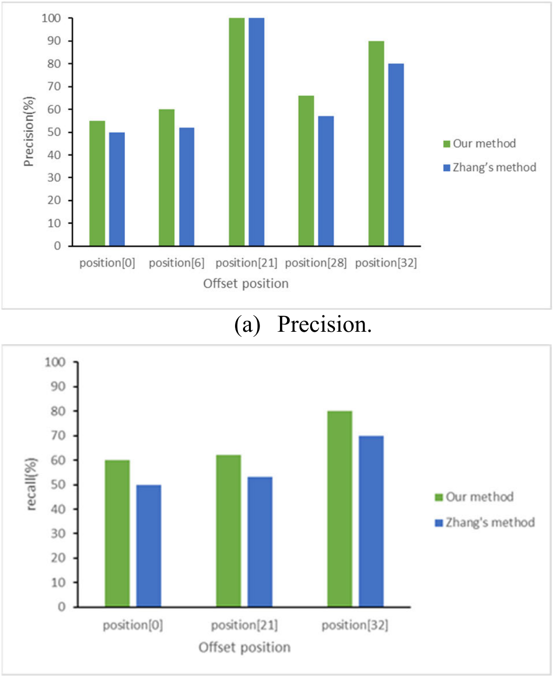
Position	Frequency	Sequence
(12)	1	080600010800060400
(28)	1	ac10
(22)	0.850	0010
(32)	0.789	0010
(6)	0.700	0010
(21)	0.5	01
(38)	1	ac10
(32)	0.789	000000000000ac10
(0)	0.665	0800



(a) Precision.

(b) Recall.

FIGURE 13. Precision and recall of ICMP message.



(a) Precision.

(b) Recall.

FIGURE 14. Precision and recall of ARP message.

TABLE 8. Summary of the time complexity.

Phase	Part of each phase	Time complexity
Composite feature extraction	N-gram generation	O(M*L)
	Feature identification	O(M*K)
Protocol message-type identification	Protocol vectorization	O(N*L)
	The measure of similarity	O(N*K)
	Class cluster evaluation	O(N*K^2)
Protocol message-format analysis	Message formats inference	O(K*L)
	Field identification	O(G*2^b)

messages with 100% precision and recall rates in the frequent sequence at position (34). However, the precision and recall rates of the proposed method are slightly higher than those of Zhang's method for other frequent sequences. In Fig. 14, both methods only find ARP messages with 100% precision and recall at sequence "01" in the 21st byte, which is the key word of ARP. For other frequent sequences, the precision and recall rates of the proposed method are higher than those of Zhang's method. Therefore, the proposed method is faster and more efficient.

A theoretical analysis of the computational complexity for each phase is presented as follows: (1) composite feature extraction. (2) Protocol message-type identification. (3) Protocol message-format analysis.

1) Composite feature extraction: It contains two important parts: N-gram generation and feature identification. The computational complexity of the two parts is presented in Table 8, where L is the first constant number of bytes of a protocol data frame, M denotes the size of the samples of the protocol dataset, and K is the number of attributes (i.e., features). As K $\gg$ L, the overall computational complexity of this phase is O $(M * K)$.

2) Protocol message-type identification contains three important parts: protocol vectorization, the measure of similarity, and class cluster evaluation. The time complexity of each part is presented in Table 8, where L is the first constant number of bytes of a protocol data frame, N is the size of the samples for clustering, and K is the number of attributes (i.e., features). Note that, in practice, the following relationship K $\gg$ L is present. Therefore, the overall computational complexity of the phase is O $\left(N * K^2\right)$.

3) Protocol message-format analysis: It contains two important parts: message-format inference and field identification, where L is the first constant number of bytes of a protocol data frame, and K is the number of attributes (i.e., features). The computational complexity of this phase is O $\left(G * 2^b\right)$, where G is the size of the field, and b is the number of bits per field.

V. CONCLUSION

This paper proposed a method for identifying and analyzing an unknown bitstream protocol. It solves the problem of difficulty obtaining protocol information and specifications against the background of zero knowledge. The proposed method identifies the protocol message type, analyzes the protocol message format, and obtains the results of field identification. Our experimental results showed that it achieved high accuracy and recall rates on the ICMP and ARP datasets. This is of great significance in the identification of unknown protocols; however, a more comprehensive analysis of the protocol's message-format information is required. We are currently working toward extending the proposed method to accurately infer semantic information about fixed-length fields.

REFERENCES

[1] G. Chen, "The solution of information network security in large scale enterprise," *J. Yueyang Normal Univ.*, to be published.

[2] X. Cai, R. Zhang, and B. Wang, "Machine learning and keyword-matching integrated protocol identification," in *Proc. 3rd IEEE Int. Conf. Broadband Netw. Multimedia Technol. (IC-BNMT)*, Oct. 2010, pp. 164–169.

[3] M.-S. Kim, Y. J. Won, and J. W.-K. Hong, "Application-level traffic monitoring and an analysis on IP networks," *ETRI J.*, vol. 27, no. 1, pp. 22–42, Feb. 2005.

[4] Y. Wang, L.-Z. Gu, Z.-X. Li, and Y.-X. Yang, "Protocol reverse engineering through dynamic and static binary analysis," *J. China Universities Posts Telecommun.*, vol. 20, pp. 75–79, Dec. 2013.

[5] H. Yang, P. Li, Q. Zhu, and L. Xu, "The application layer protocol identification method based on semisupervised learning," in *Proc. 12th ACIS Int. Conf. Softw. Eng., Artif. Intell., Netw. Parallel/Distrib. Comput.*, Jul. 2011, pp. 115–120.

[6] H. Meng, C. Song, Z. Chen, and Z. Meng, "Design of network mobility support in eXpressive internet architecture," in *Proc. Int. Conf. Connected Vehicles Expo. (ICCVE)*, Oct. 2015, pp. 169–174.

[7] W. Amei, D. Huailin, W. Qingfeng, and L. Ling, "A survey of application-level protocol identification based on machine learning," in *Proc. Int. Conf. Inf. Manage., Innov. Manage. Ind. Eng.*, Nov. 2011, pp. 201–204.

[8] D. W. Harper, J. Sabella, and W. H. Munch, "Method, apparatus and system for management of information content for enhanced accessibility over wireless communication networks," Tech. Rep., 2009.

[9] J. Caballero, H. Yin, Z. Liang, and D. Song, "Polyglot: Automatic extraction of protocol message format using dynamic binary analysis," in *Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS)*, 2007, pp. 317–329.

[10] W. Cui, J. Kannan, and H. J. Wang, "Discoverer: Automatic protocol reverse engineering from network traces," Tech. Rep., 2007.

[11] Y. Wang, X. Yun, M. Z. Shafiq, L. Wang, A. X. Liu, Z. Zhang, D. Yao, Y. Zhang, and L. Guo, "A semantics aware approach to automated reverse engineering unknown protocols," in *Proc. 20th IEEE Int. Conf. Netw. Protocols (ICNP)*, Oct. 2012, pp. 1–10.

[12] S. Tao, H. Yu, and Q. Li, "Bit-oriented format extraction approach for automatic binary protocol reverse engineering," *IET Commun.*, vol. 10, no. 6, pp. 709–716, Apr. 2016.

[13] N. Borisov, D. J. Nrumley, H. J. Wang, and C. Guo, "Generic application-level protocol analyzer and its language," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*. 2007, pp. 1–13.

[14] Y. H. Goo, K. S. Shim, B. M. Chae, and M. S. Kim, "Framework for precise protocol reverse engineering based on network traces," in *Proc. NOMS IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2018, pp. 1–4.

[15] Y.-H. Goo, K.-S. Shim, M.-S. Lee, and M.-S. Kim, "Protocol specification extraction based on contiguous sequential pattern algorithm," *IEEE Access*, vol. 7, pp. 36057–36074, 2019.

[16] J. Caballero and D. Song, "Automatic protocol reverse-engineering: Message format extraction and field semantics inference," *Comput. Netw.*, vol. 57, no. 2, pp. 451–474, 2013.

[17] Y. H. Goo, B. D. Sija, S. H. Lee, and M. S. Kim, "Analyzing the difference between network trace-based and execution trace-based protocol reverse engineering in three perspectives," in *Proc. Symp. Korean Inst. Commun. Inf. Sci.*, Jun. 2017, pp. 82–83.

[18] Y. Wang, X. Yun, M. Z. Shafiq, L. Wang, A. X. Liu, Z. Zhang, D. Yao, Y. Zhang, and L. Guo, "A semantics aware approach to automated reverse engineering unknown protocols," in *Proc. 20th IEEE Int. Conf. Netw. Protocols (ICNP)*, Oct. 2012, pp. 1–10.

[19] H. Li, B. Shuai, J. Wang, and C. Tang, "Protocol reverse engineering using LDA and association analysis," in *Proc. 11th Int. Conf. Comput. Intell. Secur. (CIS)*, Dec. 2015, pp. 312–316.

[20] G. Bossert, "Exploiting semantic for the automatic reverse engineering of communication protocols," Ph.D. dissertation, Univ. Gif-Sur-Yvette, Rennes, France, Dec. 2014.

[21] L. Wang and T. Jiang, "On the complexity of multiple sequence alignment," *J. Comput. Biol.*, vol. 1, no. 4, pp. 337–348, 1994.

[22] N. B. Amor, S. Benferhat, and Z. Elouedi, "Naïve Bayes vs decision trees in intrusion detection systems," in *Proc. ACM Symp. Appl. Comput.*, vol. 18, no. 6, pp. 420–424, 2004.

[23] A. A. Nagra, F. Han, Q. H. Ling, M. Abubaker, F. Ahmad, S. Mehta, and A. T. Apasiba, "Hybrid self-inertia weight adaptive particle swarm optimisation with local search using C4.5 decision tree classifier for feature selection problems," *Connection Sci.*, vol. 32, no. 1, pp. 16–36, Jan. 2020.

[24] R. Sambasivan and S. Das, "Classification and regression using augmented trees," *Int. J. Data Sci. Anal.*, vol. 7, no. 4, pp. 259–276, Jun. 2019.

[25] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield, "Class-of-service mapping for QoS: A statistical signature-based approach to IP traffic classification," in *Proc. 4th ACM SIGCOMM Conf. Internet Meas. (IMC)*, 2004, pp. 135–148.

[26] A.-A. Liu, Y.-T. Su, W.-Z. Nie, and M. Kankanhalli, "Hierarchical clustering multi-task learning for joint human action grouping and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 102–114, Jan. 2017.

[27] Q. Liu, J. J. Bu, and C. Chen, "Application of key words recommendation based on Apriori algorithm in theme-oriented personalized search," *Pattern Recognit. Artif. Intell.*, vol. 19, no. 2, pp. 186–190, 2006.

[28] R. Ji, H. Li, and C. Tang, "Extracting keywords of UAVs wireless communication protocols based on association rules learning," in *Proc. 12th Int. Conf. Comput. Intell. Secur. (CIS)*, Dec. 2016, pp. 310–313.

[29] F. Zhang, J. Zhang, and H. Zhou, *Unknown Bit Stream Protocol Message Discovery With Zero Knowledge*. Springer, 2015.

[30] J. Zheng, "An association analysis and identification for unknown protocol of bitstream oriented," *Concurrency Comput., Pract. Exper.*, vol. 28, no. 15, pp. 4067–4081, Oct. 2016.

[31] Clustal.org. *Clustal W/Clustal X Multiple Alignment Nucleic Acid Protein Sequences [EB/OL]*. [Online]. Available: http://www.clustal.org/clustal2/.2017-9-1

[32] T. H. Shosaku, "Estimation of the word frequency distribution based on n-gram pattern frequency distributions and the Zipf's law," in *Proc. Rec. Joint Conf. Elect. Electron. Eng.*, Kyushu, Japan, 2004.

[33] R. Prabamanieswari, "NCFP-tree: A non-recursive approach to CFP-tree using single conditional database," Tech. Rep., 2017.

[34] C.-E. Ben Ncir, A. Hamza, and W. Bouaguel, "Parallel and scalable Dunn Index for the validation of big data clusters," *Parallel Comput.*, vol. 102, May 2021, Art. no. 102751.

[35] H. Seifoddini and M. Djassemi, "The production data-based similarity coefficient versus Jaccard's similarity coefficient," *Comput. Ind. Eng.*, vol. 21, nos. 1–4, pp. 263–266, Jan. 1991.

[36] K. Frenken, *Entropy Statistics and Information Theory*. Toronto, ON, Canada: Chapters, 2007.

SHUCHENG WANG is currently pursuing the M.S. degree in communication engineering with the Electronic Information School, Wuhan University, Hubei, China.

His research interests include protocol reverse engineering and communication and information systems.

FAN GUO received the M.S. degree in communication engineering from the Electronic Information School, Wuhan University, Hubei, China.

His research interests include protocol reverse engineering and signal processing.

YONG FAN received the M.S. degree in computer science from the Communication University of China, Beijing, China.

His research interests include signal processing, big data science, and machine learning.

JING WU (Member, IEEE) received the B.Eng. degree in communication engineering and the Ph.D. degree in communication and information systems from Wuhan University, China, in 2002 and 2007, respectively. From 2004 to 2005, she undertook her postdoctoral research work at LIMOS, Clermont-Ferrand, France. She is currently an Associate Professor with Wuhan University. Her research interests include wireless communication networks, network simulation, and intelligence data processing.