Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 962974, 9 pages
http://dx.doi.org/10.1155/2015/962974

Research Article

Capturing Uncertainty Information and Categorical Characteristics for Network Payload Grouping in Protocol Reverse Engineering

Jian-Zhen Luo,[1,2] Shun-Zheng Yu,[1] and Jun Cai[2]

[1]*School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510275, China*
[2]*School of Electronic and Information, Guangdong Polytechnic Normal University, Guangzhou 510665, China*

Correspondence should be addressed to Shun-Zheng Yu; syu@mail.sysu.edu.cn

Received 21 January 2015; Revised 11 May 2015; Accepted 12 May 2015

Academic Editor: Filippo Ubertini

As a promising tool to recover the specifications of unknown protocols, protocol reverse engineering has drawn more and more attention in research over the last decade. It is a critical task of protocol reverse engineering to extract the protocol keywords from network trace. Since the messages of different types have different sets of protocol keywords, it is an effective method to improve the accuracy of protocol keyword extraction by clustering the network payload of unknown traffic into clusters and analyzing each clusters to extract the protocol keywords. Although the classic algorithms such as K-means and EM can be used for network payload clustering, the quality of resultant traffic clusters was far from satisfactory when these algorithms are applied to cluster application layer traffic with categorical attributes. In this paper, we propose a novel method to improve the accuracy of protocol reverse engineering by applying a rough set-based technique for clustering the application layer traffic. This technique analyze multidimension uncertain information in multiple categorical attributes based on rough sets theory to cluster network payload, and apply the Minimum Description Length criteria to determine the optimal number of clusters. The experiments show that our method outperforms the existing algorithms and improves the results of protocol keyword extraction.

1. Introduction

Network protocol reverse engineering [1–4] is a promising approach to address the problem of recovering detailed specifications of unpublished or undocumented network protocols from the network trace. The specifications of protocols play an important role in the network security and management oriented issues, such as intrusion detection, fuzzing test [5], recovering and understanding command-and-command (C&C) protocols [6], and building intelligent honeypot [7]. The specifications of open protocols such as HTTP are available from the published document. However, the specifications of some protocols (called proprietary protocols) used by enterprises or hackers are not open to public for the reason of commercial or security. Researchers deem that protocol reverse engineering is the only option available to build the understanding of proprietary protocol from network trace.

The extraction of protocol keywords from network trace is a critical task of protocol reverse engineering. The protocol keywords [4] are referred to as the constant strings or command characters used by the protocol. For example, the HTTP protocol uses "GET" as the request method and the FTP protocol uses "QUIT" as the command to quit a session. If the messages are not grouped into clusters to make sure each cluster belongs to a unique message type, some protocol keywords (associated with a particular message type) with low occurrence probability will be missed due to the interference of miscellaneous types of messages. Thus, it is critical that the messages input to a protocol reverse engineering system belong to a single type. In practice, it is a challenge to cluster messages of unknown traffic according to message types, since we have no prior knowledge about the unknown protocols. An adapted solution to these issues is to apply unsupervised clustering methods to group messages of the unknown traffic.

Clustering unlabeled data is also important for many other applications, such as building an automatic method to generate signatures for unknown or new network applications in network management. Such issues have been studied by many works [8–12] most of which are applicable for clustering data having attributes with numerical values. The reason why most clustering algorithms are focused on attributes with numerical value is that it is easy to define similarities of the numerical data using geometric concepts. However, much data contained in today's databases and much information which is valuable for clustering is categorical in nature. Specifically, in our problem domain, the protocol messages analyzed by reverse engineering systems expose mainly categorical characteristics, such as message direction, transport layer number used by the protocol, words, or delimiters used by the messages. On the other hand, most clustering algorithms are crisp clustering algorithms and they are inclined to classify an object into one and only one cluster [13]. However, it is hard to define the crisp boundaries of clusters in many practical cases and the crisp clustering algorithms lack the ability to handle the uncertain information hidden in the data set [14]. The messages that belong to different types reflect different string patterns. For instance, some messages start with a string of "M-SEARCH," while others start with "GET." Hence, the remarkable characteristic is valuable for clustering application layer traffic. However, some attributes in different clusters have the same value. For example, the first 4 bytes of some messages of both HTTP and SSDP are "HTTP." Thus, when we have a message whose first 4 bytes are "HTTP," it is difficult to determine which of HTTP and SSDP dose the message belong to. In other words, each attribute of a message has different degree of confidence to indicate the message's belonging to different types. We have to deal with the uncertain information of message attributes in the clustering process.

In this paper, the main innovative contribution is to improve the accuracy of protocol reverse engineering by applying a rough sets theory- (RST-) [14–16] based method to tackle the problems of clustering application layer traffic and grouping the messages according to the message types. In the process of clustering, we propose to select cluster for further clustering according to the degree of uncertainty within the cluster instead of the number of objects within the cluster instead of selecting the subclusters with more objects for further splitting [17]. The approach is implemented on a system called RSCluster and applied to cluster real-world application layer network trace according to the protocols and group protocol messages according to the message types in order to improve accuracy of the protocol keyword extraction.

Besides the application of improving accuracy of protocol reverse engineering, the RSCluster is supposed to be an efficient and accurate traffic clustering tool for classifying and identifying newly emergent applications which are most likely unknown to network administrators. Based on the clustering results, one can generate models or signatures to represent the profile of each unknown application so as to distinguish them from each other in the future.

The remainder of this paper is arranged as follows. Section 2 studies the related work. Section 3 presents an overview of rough sets theory and outlines some basic definitions. Section 4 presents our methodology of traffic clustering. Section 5 discusses the Minimum Description Length criteria for model selection. The computation complexity is discussed in Section 6 and the approach is evaluated in Section 7. Finally, a conclusion is made in Section 8.

2. Related Work

In the last decade, numbers of techniques have been applied to cluster Internet traffic. The simplest approach is to cluster flows according to the well-known ports assigned by IANA [18]. However, more and more applications exceedingly use dynamic port numbers during the communication in the Internet. Furthermore, considerable applications were exposed to hide their traffic behind some well-known traffic (e.g., HTTP) to be transmitted over well-known port so as to bypass the detection of firewalls. Therefore, the port based approaches are insufficient in many cases.

Recently, a number of algorithms have been proposed to apply the classic clustering methods such as K-means and EM methods to address these issues. These techniques assume that traffic has statistical attributes (e.g., packet lengths, average length of packets, and packet interarrival time) which are unique for certain classes of applications. Hence, one could distinguish different kinds of applications from each other by leveraging the flow's statistical characteristics. Erman et al. [8] propose two unsupervised clustering algorithms (i.e., K-means and DBSCAN) to classify traffic by exploiting the distinctive characteristics of applications when they communicate on a network. In [9], Erman et al. apply unsupervised EM clustering technique for the Internet traffic classification problem. The unsupervised clustering approach uses an EM algorithm to classify unlabeled training data into groups based on similarity. Although these techniques have improved the accuracy in a certain extent, the reported results were still far from satisfactory.

The difference between ours and [8, 9] is that we cluster traffic by considering categorical value attributes and analyzing the multidimension of uncertainty between attributes, instead of quantifying flow attributes into numerical value attributes and applying geometric similarity. As we know, inaccurate or inappropriate quantification will lose some valuable information of data or distort the substantial truth. As a result, it is very important to keep the data as it is as far as possible in order to preserve more information.

In order to build up the categorical data clustering, Mahmood et al. [19] develop a framework to deal with mixed type attributes including numerical, categorical, and hierarchical attributes for a one-pass hierarchical clustering algorithm. However, they focus on analyzing network flow feature such as protocols (UDP, TCP, and ICMP) to identify interesting traffic patterns from network traffic data, while we aim to analyze the categorical features in application layer to cluster network traffic according to protocols and group protocol messages according to message types. Other researchers propose to take advantage of RST to cluster

qualitative-value data [20]. Mazlack et al. [21] propose RST-based technique to choose effective attributes for data partitioning in the procedure of clustering. Parmar et al. propose an novel algorithm, namely, Min-Min-Roughness (MMR) [17], for categorical data clustering. They show that MMR performs well in handling the uncertainty of multidimension categorical attributes of data. In each iteration of the clustering process, the MMR algorithm selects a subcluster with more objects for further splitting. However, our approach chooses cluster for further splitting according to the degree of uncertainty instead of the number of objects within the cluster. The rationality of this alternative will be explained in the next section.

Wang et al. [22, 23] demonstrate an approach to cluster unknown traffic and determine the number of clusters. Since the exact number of classes is unknown in advance, they apply the X-means algorithm to efficiently estimate the best value of cluster number K by integrating Bayesian information criterion with basic K-means. Georgieva et al. [24] propose an approach to determine the number of cluster based on the Minimum Description Length (MDL) criteria [25, 26]. Toward the number of clusters, we also explore MDL criteria to capture the optimal number of clusters. We determine the cluster number by choosing the clustering model which minimizes the total description lengths of describing the model and encoding the data set with the help of the model.

3. Traffic Clustering Using Rough Sets Theory

The concept of rough sets theory (RST) [15] was developed by Pawlak in 1982. To date, RST has received considerable attention of research in the computational intelligence literature for its excellent capability of handling imprecision, vagueness, and uncertainty in data analysis [14, 16, 17, 21]. Parmar et al. [17] show that the RST-based algorithm is appropriate to cluster categorical data and handle uncertainty in the clustering process. In what follows, we will introduce the basic concepts of rough sets theory.

3.1. Information System for Clustering. In order to formulate the problem of traffic clustering or message grouping, an information system S is defined as an ordered quadruple $S = (U, Q, V, \rho)$. The universe $U = \{x_1, x_2, \ldots, x_n\}$ contains all objects (sessions or messages) in the network trace. The attribute set is denoted by $Q = \{a_1, a_2, \ldots, a_m\}$, where every a_i is an attribute of the objects in U. $V_{a_i \in Q}$ stands for the domain of a_i, whereas $V = \cup_{\forall a_i \in Q} V_{a_i}$. Finally, the description function $f : U \times Q \to V$ maps an object $x \in U$ and an attribute $a_i \in Q$ to the value domain V.

3.2. Indiscernibility Relation. For any two objects $x_i, x_j \in U$, they are indiscernible with respect to attribute a (denoted by $\sim_a$) in S if and only if $f(x_i, a) = f(x_j, a)$. More generally, given a subset $A \subset Q$, if x_i and x_j are indiscernible with respect to every $a' \in A$, then x_i and x_j are defined to be indiscernible by the set of A; that is, $\forall a' \in A, f(x_i, a') = f(x_j, a')$. The indiscernibility relation on the set A in S, in symbols $\sim_A$, is defined as follows: $x_i \sim_A x_j$ if and only if x_i and x_j are indiscernible by the set of A.

Importantly, the elementary set, denoted by E, in U with respect to A, is defined as the equivalence class of relation $\sim_A$. The family of all elementary sets with respect to A is denoted by $U/A = \{E_1, E_2, \ldots, E_{|U/A|}\}$. For any element x_i of U, the equivalence class of x_i of relation $\sim_A$ is represented as $[x_i]_{\sim_A}$.

3.3. Lower and Upper Approximation. In what follows, we introduce two important concepts in rough sets-based approach for data analysis, namely, the lower approximation and upper approximation.

Let $A \subset Q$ be a set of attributes, and let X be a subset of objects in universe U (i.e., $X \subset U$). The lower approximation of X with respect to A in U, denoted by $\underline{\Delta}_A(X)$, is defined as the union of all those elementary sets which are contained in X. That is, given $U/A = \{E_1, E_2, \ldots, E_{|U/A|}\}$, we have

$$\underline{\Delta}_A(X) = \bigcup_{E_i \subseteq X} E_i = \left\{ x \mid x \in U \wedge [x]_{\sim_A} \subseteq X \right\}. \tag{1}$$

By this notation, it is easy to have that an object $x \in U$ belongs to X doubtlessly, if $x \in \underline{\Delta}_A(X)$.

On the other hand, the upper approximation of X with respect to A in U, denoted by $\overline{\nabla}_A(X)$, is defined as the union of all those elementary sets which have a nonempty intersection with X. More formally, given $U/A = \{E_1, E_2, \ldots, E_{|U/A|}\}$, we have

$$\overline{\nabla}_A(X) = \bigcup_{E_i \cap X \neq \phi} E_i = \left\{ x \mid x \in U \wedge [x]_{\sim_A} \cap X \neq \phi \right\}. \tag{2}$$

We note that if an object $x \in \overline{\nabla}_A(X)$, it implies that x possibly belong to X.

An *accuracy* measure of the set X in $A \subseteq Q$ is defined as

$$\mu_A(X) = \frac{\operatorname{card}\left(\underline{\Delta}_A(X)\right)}{\operatorname{card}\left(\overline{\nabla}_A(X)\right)}, \tag{3}$$

where $\operatorname{card}(\underline{\Delta}_A(X))$ or $\operatorname{card}(\overline{\nabla}_A(X))$ is the number of objects contained in the lower or upper approximation of the set X with respect to A. Obviously, $0 \leq \mu_A(X) \leq 1$.

If $\mu_A(X) = 1$, then the set X is asserted to be definable in U with respect to A. Otherwise, X is undefinable in U.

For ease of exposition, we define the notion of *roughness* as follows:

$$R_A(X) = 1 - \mu_A(X) = 1 - \frac{\operatorname{card}\left(\underline{\Delta}_A(X)\right)}{\operatorname{card}\left(\overline{\nabla}_A(X)\right)}. \tag{4}$$

If $R_A(X) = 0$, X is crisp with respect to A. If $0 < R_A(X) \leq 1$, X is rough with respect to A.

3.4. Classification of Information System. Let $F = \{C_1, C_2, \ldots, C_n\}$, $C_i \subset U$, be a family of subsets of the universe U. If F is a partition of U, that is,

$$C_i \cap C_j = \phi \quad \forall i, j,\ 1 \leq i, j \leq n,\ i \neq j,$$
$$\bigcup_{i=1}^{n} C_i = U \tag{5}$$

then F is a *classification* of U, whereas C_is are called classes of F.

Suppose that A is subset of Q, the lower and upper approximation of F with respect to A in S are defined as

$$\begin{aligned}\underline{\Delta}_A(F) &= \left\{\underline{\Delta}_A(C_1), \underline{\Delta}_A(C_2), \ldots, \underline{\Delta}_A(C_n)\right\},\\ \overline{\nabla}_A(F) &= \left\{\overline{\nabla}_A(C_1), \overline{\nabla}_A(C_2), \ldots, \overline{\nabla}_A(C_n)\right\},\end{aligned} \tag{6}$$

respectively.

With these notations, the *quality* of the classification with respect to A is given as

$$\eta_A(F) = \frac{\operatorname{card}\left(\bigcup_{i=1}^{n} \underline{\Delta}_A(C_i)\right)}{\operatorname{card} U}, \tag{7}$$

and the *accuracy* of the classification with respect to A is given as

$$\mu_A(F) = \frac{\operatorname{card}\left(\bigcup_{i=1}^{n} \underline{\Delta}_A(C_i)\right)}{\operatorname{card}\left(\bigcup_{i=1}^{n} \overline{\nabla}_A(C_i)\right)}. \tag{8}$$

4. Traffic Clustering Using Rough Sets Theory

As suggested by Mazlack et al. [21], data clustering is a series of procedures to discover the intraitem dissonance of data and eliminate it within the resulting subpartitions by progressively partitioning the data set. Inspired by this rationale, traffic clustering could be achieved by recursively partitioning the data set to reduce the dissonance (uncertainty) within the resulting partitions. Obviously, the crisp partitioning is the most desired situation because there is no dissonance inside the partitions. However, it cannot always be achieved in real world. Thus, our proposed algorithm considers the partitioning leading to less uncertainty.

Following this heuristic thought, the procedure of traffic clustering should be performed in the following way: we firstly choose an effective partitioning attribute, then split the data set by searching for a partitioning point of the selected attribute so as to maximize the coherence of resulting partitions. These procedures are repeated until either the coherence is no longer changed or a predefined termination condition is satisfied.

4.1. Roughness and Average Roughness. In order to deal with the uncertain information in data set and quantify the degree of the uncertainty, we define the some related concepts based on roughness [17, 21] as follows.

Given an attribute $a_i \in A$ and the domain of its possible values $V_{a_i} = \{\alpha_1, \alpha_2, \ldots, \alpha_{|V_{a_i}|}\}$, the family of all elementary sets with respect to a_i in S is denoted by $U/a_i = \{E_{a_i}(\alpha_1), E_{a_i}(\alpha_2), \ldots, E_{a_i}(\alpha_{|V_{a_i}|})\}$, where $E_{a_i}(\alpha_k)$ refers to a specific elementary set (with respect to a_i) whose value of a_i is α_k ($k = 1, 2, \ldots, |V_{a_i}|$). In essence, U/a_i is indeed an n-partitioning ($n = |V_{a_i}|$) on U towards attribute a_i and $E_{a_i}(\alpha_k)$ is the kth partition.

The *roughness* of a_i with respect to another attribute $a_j \in A$ on kth partition is defined as

$$R_{a_j}\left(E_{a_i}(\alpha_k)\right) = 1 - \frac{\underline{\Delta}_{\{a_j\}}\left(E_{a_i}(\alpha_k)\right)}{\overline{\nabla}_{\{a_j\}}\left(E_{a_i}(\alpha_k)\right)}. \tag{9}$$

The average roughness of a_i with respect to a_j is defined as the mean of roughnesses of a_i with respect to a_j on all partitions towards a_i. That is,

$$\overline{R}_{a_j}(a_i \mid X) = \frac{\sum_{k=1}^{|V_{a_i}|} R_{a_j}\left(E_{a_i}(\alpha_k)\right)}{\left|V_{a_i}\right|}. \tag{10}$$

Note that, the average roughness $\overline{R}_{a_j}(a_i \mid X)$ ranges from 0 to 1. When $\overline{R}_{a_j|X}(a_i) = 0$, it implies a crisp partitioning on attribute a_i. When $\overline{R}_{a_j}(a_i) = 1$, it implies a maximum rough partitioning on attribute a_i. The smaller is $\overline{R}_{a_j}(a_i \mid X)$, the crisper is the partitioning on attribute a_i [17, 21].

Furthermore, the *attribute roughness* of a_i, in symbols $\overline{R}(a_i \mid X)$, is the mean of average roughness of a_i with respect to other attributes in Q:

$$\overline{R}(a_i \mid X) = \frac{\sum_{\forall a_j \in Q, a_j \neq a_i} \overline{R}_{a_j}(a_i \mid X)}{|Q| - 1}. \tag{11}$$

Obviously, $\overline{R}(a_i \mid X)$ measures the effect of the partitioning using attribute a_i toward all other attributes. It ranges from 0 to 1. The smaller is the $\overline{R}(a_i \mid X)$, the crisper is the partition [21].

4.2. Splitting a Cluster. Recall that it cannot always achieve the crisp partitioning in real world clustering cases, so we have to consider the partitioning scheme that leads to minimum uncertainty. Following this key, the goal of our algorithm is to minimize the roughness of data set.

Given a specific subset of $C \subseteq U$, we firstly search for the attribute that should be selected to conduct the splitting of C. Recall that the attribute roughness $\overline{R}(a_i \mid C)$ measures the effect of the partitioning using attribute a_i. So, the attribute whose attribute roughness is minimal should be selected for splitting. Hence, the objective attribute is

$$a^* = \arg\min_{a_i \in Q}\left(\overline{R}(a_i \mid C)\right). \tag{12}$$

In what follows, we illustrate how to split C into two partitions of C_1 and C_2 with the help of a^*, where $C_1 \cap C_2 = \phi$ and $C_1 \cup C_2 = C$.

First of all, we define the summation of roughness of a_i on partition $E_{a_i}(\alpha)$ as

$$\gamma_{a_i}(\alpha) = \sum_{a_j \in Q, a_j \neq a_i} R_{a_j}\left(E_{a_i}(\alpha)\right). \tag{13}$$

Secondly, we rank $\gamma_{a^*}(\alpha_k)$ in ascending order. Suppose that the rank result is given as

$$\gamma_{a^*}(\alpha_1) \leq \gamma_{a^*}(\alpha_2) \leq \cdots \leq \gamma_{a^*}\left(\alpha_{|V_{a^*}|}\right). \tag{14}$$

Given $1 \leq r \leq s \leq |V_{a^*}|$, we define the following symbols:

$$\begin{aligned}V^*_{r:s} &= \bigcup_{k=r}^{s} \{\alpha_k\},\\ C_{r:s} &= \left\{x \mid \forall x \in C, f(x, a^*) \in V^*_{r:s}\right\}.\end{aligned} \tag{15}$$

Using these notations, the splitting of C can be denoted by

$$F_k(C) = \left\{C_{1:k}, C_{k:|V_{a^*}|}\right\}. \quad (16)$$

The attribute roughness of a^* in $C_{1:k}$ and $C_{k:|V_{a^*}|}$ is $\overline{R}_{C_{1:k}}(a_i \mid C_{1:k})$ and $\overline{R}_{C_{k:|V_{a^*}|}}(a_i \mid C_{k:|V_{a^*}|})$, respectively. Hence, the problem of determining the optimal splitting of C becomes the problem of finding a splitting point k to minimize the total roughness of the two subclusters. In other words, the optimal splitting point should be determined by finding out the optimal value of k^* as

$$k^* = \arg\min_k \left(\overline{R}_{C_{1:k}}(a_i \mid C_{1:k}) + \overline{R}_{C_{k:|V_{a^*}|}}\left(a_i \mid C_{k:|V_{a^*}|}\right)\right). \quad (17)$$

4.3. Selecting Subclusters for Further Splitting. In the previous work, the Min-Min-Roughness [17] algorithm selects the subclusters having more objects for further splitting at subsequent iterations. However, if the uncertainty of a cluster having more objects is much smaller than that of another one which has less objects, it is rational to split the latter cluster for it contains much more intraitem dissonance than that of the former one. Therefore, we should choose the cluster for further clustering according to the degree of uncertainty within the cluster instead of the number of objects within the cluster.

The *cluster roughness* of a cluster C is defined as the summation of all attribute roughness in cluster C. That is,

$$R(C) = \sum_i \overline{R}(a_i \mid C). \quad (18)$$

In this paper, we choose the clusters with the largest cluster roughness for further splitting. We also constrain that the size of selected cluster must be larger than a predefined threshold (minimum cluster size) so as to avoid the overclustering.

We apply our method firstly on the universe U to split it into two subclusters. Then, we choose subclusters whose cluster roughness is the largest to repeat the procedures of splitting to obtain further partitions. We apply our algorithm recursively until the predefined termination condition is satisfied or no cluster is selected to be further splitting. The predefined termination condition is that either the passes of iteration or the number of clusters reaches their corresponding upper bound.

5. Determining Number of Clusters Based on MDL Principle

5.1. The Minimum Description Length Principle. The Minimum Description Length (MDL) principle which is proposed by Rissanen [25, 26] has been successfully applied to select the optimal model from a set of given stochastic models. The MDL principle asserts that the best model inferred from a given set of data is the one which minimizes the total description lengths of both the model and the encoding for the data with the help of the model.

More specifically, when a set of models $\{\theta^{(i)} \mid i = 1, 2, \ldots, I\}$ is given, the description length of an observation $X = \{x_1, x_2, \ldots, x_N\}$ using the ith model is given as

$$L_i(X) = -\log P\left(X \mid \theta^{(i)}\right) + \frac{\alpha_i}{2}\log N + \log I, \quad (19)$$

where α_i is the number of free parameters in the ith model.

We note that $P(X \mid \theta^{(i)})$ denotes the likelihood of data X with respect to model $\theta^{(i)}$. This term can also be viewed as the description length of the encoding of data X with the help of model $\theta^{(i)}$. The second term in (19) is related to the complexity of model $\theta^{(i)}$ and the size of observation data X. The third term is the code length representing the number of models for selection.

5.2. Determine the Number of Clusters Based on MDL. Given $S = (U, Q, V, \rho)$, let $F^{(k)} = \{C_1^{(k)}, C_2^{(k)}, \ldots, C_{n_k}^{(k)}\}$ be the classification of U in the kth iteration and let α_{a_i} be a variable standing for the value of a_i. For each class $C_j^{(k)}$, we calculate the entropy of a_i with respect to $C_j^{(k)} \in F^k$ as follows:

$$H_{C_j^{(k)}}\left(\alpha_{a_i}\right) = -\sum_{\alpha_k \in V_{a_i}} P_{C_j^{(k)}}(\alpha_k)\log P_{C_j^{(k)}}(\alpha_k), \quad (20)$$

where $P_{C_j^{(k)}}(\alpha_k) = P(\rho(x \in C_j^{(k)}, a_i) = \alpha_k)$.

For the sake of simplicity, we also denote the description length of universe U as follows:

$$H_{C_j^{(k)}}(U) = \sum_{\forall a_i \in Q} H_{C_j^{(k)}}\left(\alpha_{a_i}\right). \quad (21)$$

The total description length of the information system S in kth iteration is given by

$$L_k = \underbrace{\sum_j H_{C_j^{(k)}}(U)}_{L_{(k,1)}} + \underbrace{\frac{k}{2}\log|C|}_{L_{(k,2)}} + \underbrace{\log I}_{L_{(k,3)}}, \quad (22)$$

where the first term, $L_{(k,1)}$, in (22) is the description length of encoding data set U using the kth clustering scenario of $F^{(k)}$, the second term, $L_{(k,2)}$, is the length describing the complexity of $F^{(k)}$, and the last term, $L_{(k,3)}$, is the code length of representing the number of models for selection.

Recall that, in each iteration, we select an optimal attribute for each candidate nodes to perform splitting procedure so as to minimize the roughness in the information system S, so the uncertain information in S decreases. Therefore, as the recursive procedure of clustering goes further, the term of $L_{(k,1)}$ in (22) will decrease constantly. On the other hand, as the number of clusters in F increases, the classification F will become more and more complex and computational cost will increase. So, the second term of $L_{(k,2)}$ in (22) will increase. In particular, $L_{(k,3)}$ is a constant. In summary, the total description length of S will firstly decrease until it reaches the minimum value point and then increase.

Table 1: Data sets for evaluation.

Data set	Collecting date	Total flows
Data set I	Aug 08, 2012	385
Data set II	Aug 10, 2012	1334
Data set III	Aug 11, 2012	683

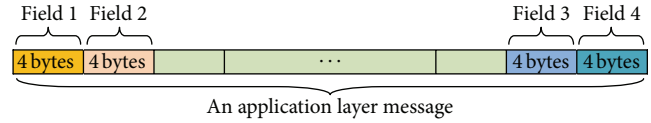

Figure 1: Four-byte fields in application layer messages.

Thus, we can search for a k_0 to minimize the value of L_k in (22):

$$k_0 = \arg\min_k L_k. \tag{23}$$

As a result, the number of clusters in k_0th iteration is optimal.

6. Computation Complexity

In this section, we discuss the complexity of our algorithm. Suppose that there are totally n objects and m attributes are considered. The worst-case condition is that each attribute has distinct values for each object. That is, for each attribute a_i, the cardinality of V_{a_i} is exactly equal to n ($|V_{a_i}| = n$) and the cardinality of the family of elementary sets with respect to this attribute is also equal to n ($|U/\{a_i\}| = n$). Therefore, we need at most n comparisons for each object to judge whether it belongs to one elementary set. In the worst situation, we have to perform the n-comparison judgement for totally n elementary. So, the complexity of calculating the average roughness of a_i to another attribute a_j is n^2. Since there are totally m attributes, the complexity of computing the attribute roughness of a specific attribute is $m \times n^2$. In the procedure of finding the partitioning attribute, we have to do m passes of attribute roughness calculations, so the complexity is $m^2 \times n^2$.

On the other hand, in the worst case, the complexity for $\gamma_{a_i}(\alpha)$ in (13) is $m \times n$, the complexity of sorting $\gamma_{a_i}(\alpha)$ is n, and the complexity for finding optimal splitting point in (17) is $m \times n^2$.

In a summary, the total complexity of our algorithm is $O(m^2 \times n^2 + m \times n + n + m \times n^2)$. For a large data set, the value of n is very large so the value of m could be considered as a constant comparatively. Thus, the complexity of our algorithm is $O(n^2)$.

7. Evaluation

The proposed algorithm is implemented in a system called RSCluster. In the first phase, the RSCluster is applied to clustering application layer traffic. Three data sets (i.e. data sets I, II and III, as shown in Table 1) are collected from School of Information and Science Technology in Sun Yat-Sen University on August 8, 10, and 11, in 2012. Table 2 shows the detail information of our data sets.

The overall accuracy is used to evaluate the overall effectiveness of the proposed algorithm based on rough sets theory. The dominating application in a cluster is used to label the cluster. Thus, the overall accuracy of clustering is defined as the ratio from the number of flows labeled correctly in all clusters to the total number of flows in the data set.

Suppose that the number of flows labeled correctly in a cluster of c_i is referred to as the True Positives (TP), denoted by T_i. Thus, the overall accuracy is given as $\theta = \sum_i T_i/N$, where N is the total number of sessions in data set.

Other metrics are listed as follows.

False negative (FN) is the number of sessions which are incorrectly classified as not belonging to a cluster.

False positive (FP) is the number of sessions that are incorrectly classified as belonging to a cluster.

True negative (TN) is the number of sessions that are correctly classified as not belonging to a cluster.

There are a total of 26 features used in our experiments, including the transport layer protocol type (TCP or UDP), transport layer port number, and the 4-byte fields occurring in the messages.

In the domain of traffic classification, Moore and Papagiannaki [27] and Ye et al. [28] show that effective application signatures with high accuracy can be generated using the first N bytes of each flow. Haffner et al. [29] present three main motivations for limiting the data size to first N bytes. (1) It is helpful for identifying traffic as early as possible. (2) Most application layer headers at the beginning of a data exchange are easy to be identified. (3) It allows the proposed algorithm to process less amount of data.

Therefore, it is enough to capture sufficient information about the class characteristic of message by considering the first 4-byte field (field 1), the second 4-byte field (field 2), the last 4-byte field (field 3), and the last but second 4-byte field (field 4) in each message as shown in Figure 1.

Previous researches [27–30] also indicate that concrete signature usually exists in the first few packets of a connection. So, for each session, it is sufficient to consider the first 6 messages as shown in Figure 2. If the corresponding feature does not exist, a special string "NULL" would be used in that position.

Since RSCluster has no prior information about data set, the exact number of clusters is totally unknown to our system. In order to determine an appropriate number of clusters, the MDL criteria are applied to choose a clustering model whose description length is the minimum. The candidate models are the mediate results in each pass of the clustering process. Figure 3 illustrates the description length for the three data sets. For example, in data set I, the minimum value of total description length is taken in 21st iteration.

Figure 4 shows the overall accuracy as the iterations increase. As we see, the overall accuracy of RSCluster increases at first until it reaches an upper bound. The reason for this upper bound is that there is no longer cluster selected by RSCluster for further splitting so the number of cluster

Table 2: Traffic class breakdown for data sets.

Application	Data set I		Data set II		Data set III	
	Flows	Percent	Flows	Percent	Flows	Percent
FTP	82	21.%	1	0.1%	1	0.1%
SMTP	21	5.5%	720	54.0%	350	51.2%
DNS	22	5.7%	156	11.7%	102	14.9%
POP3	26	6.8%	0	0.0%	0	0.0%
HTTPS	38	9.9%	334	25.0%	155	22.7%
XunLei	11	2.9%	1	0.1%	1	0.1%
MSSQL	0	0.0%	1	0.1%	0	0.0%
eMule	44	11.%	34	2.5%	14	2.0%
BiTorrent	7	1.8%	71	5.3%	50	7.3%
Kugoo	134	34.8%	4	0.3%	3	0.4%
QQ	0	0.0%	3	0.2%	2	0.3%
PPTV	0	0.0%	1	0.1%	0	0.0%
BitSpirit	0	0.0%	8	0.6%	5	0.7%
Total	385		1334		685	

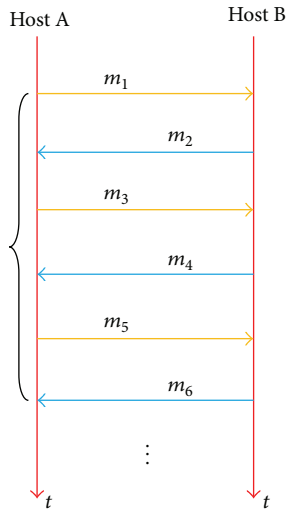

Figure 2: Messages exchanged between two hosts. "m_1," "m_2,"... stand for messages exchanged between host A and host B. Only the first 6 messages are considered in the experiments.

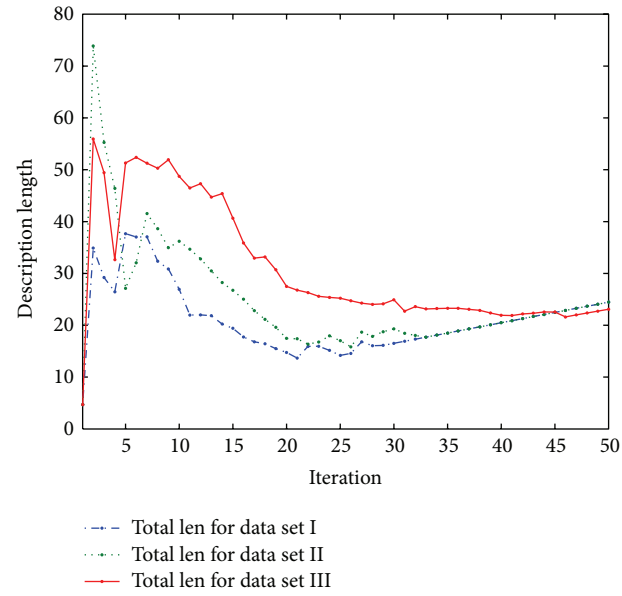

Figure 3: Description lengths.

no longer changes. Recall that, RSCluster chooses the cluster whose cluster roughness is largest as the candidates for further splitting. Thus, if all cluster roughness is 0 or the cluster size of each candidate is smaller than the minimum cluster size, no cluster will be chosen to split any more.

We also implement EM and K-means algorithm to cluster the same data sets. We repeat the two clustering algorithms for 200 iterations. In the kth iteration, we set the expected number of clusters to k. As shown in Figure 4, the proposed RST-based clustering algorithm excellently outperforms EM and K-means algorithms.

7.1. Message Grouping for Improving Protocol Reverse Engineering. In the second phase, the RSCluster is used to cluster application layer messages to improve the protocol reverse engineering accuracy. The traffic is captured from School of Information and Science Technology in Sun Yat-Sen University during April, 2011. The traffic has been classified into 4 classes of protocols (i.e., HTTP, POP, SMTP, and FTP) using the well-known network traffic analysis tool named Wireshark. In this experiment, the RSCluster is used to cluster messages for each protocol to improve the results of reverse engineering. The features used in this experiment include message direction (i.e., from the initiator to responder or from the responder to initiator) and the 4-byte fields in the messages as shown in Figure 1. The parameters and procedures are the same as Section 7.1. The grouped messages are taken as input into AutoReEngine [4] to extract protocol keywords. The protocol keywords extracted by AutoReEngine are shown in Table 3. The keywords in italic (e.g., *POST, Origin:, Cache-Control:, ITYPE:,* and *OTYPE:*) are those keywords with low occurrence probability.

Table 3: Protocol keywords extracted by AutoReEngine.

Protocol	Results without message grouping	Results with message grouping
HTTP	GET /, Referer: http://, Date:, HTTP/1.1, Keep-Alive, Accept:, Content-Length:, Last-Modified:, Accept-Encoding: gzip, Server:, Content-Type:, Connection:, Accept-Language: zh-cn, Host:, 200 OK, User-Agent: Mozilla/	GET /, Referer: http://, Date:, HTTP/1.1, Keep-Alive, Accept:, Content-Length:, Last-Modified:, Accept-Encoding: gzip, Server:, Content-Type:, Connection:, Accept-Language: zh-cn, Host:, 200 OK, User-Agent: Mozilla/, *Origin:, Cache-Control:, POST, ITYPE:, OTYPE:*
POP	USER, PASS, STAT, DATA, +OK Welcome to coremail Mail Pop3 Server, +OK core mail, message(s) [, +OK, byte(s)]	USER, PASS, STAT, DATA, +OK Welcome to coremail Mail Pop3 Server, +OK core mail, message(s) [, +OK, byte(s)] *LIST, UIDL, CAPA*
SMTP	EHLO, 250-PIPELINING, 220, 250-SIZE, 250-AUTH=LOGIN, 250 8BITMIME, 334, DATA, 250-AUTH LOGIN PLAIN,	EHLO, 250-PIPELINING, 220, 250-SIZE, 250-AUTH=LOGIN, 250 8BITMIME, 334, DATA, 250-AUTH LOGIN PLAIN, *QUIT, RCPT TO, MAIL FROM*
FTP	USER, PASS, 331, 30, 220, ready. . ., User, logged in.	USER, PASS, 331, 30, 220, ready. . ., User, logged in. *QUIT, PSAV, SIZE, CMD, 150, 530*

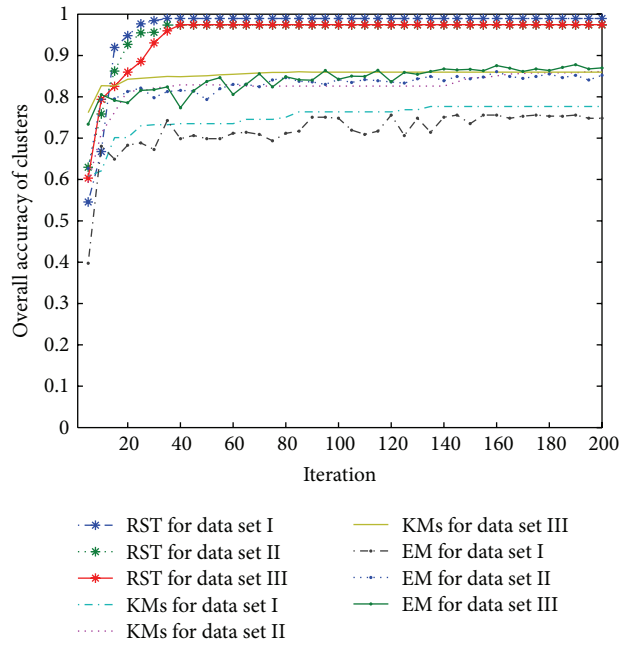

Figure 4: The overall accuracy of clustering. "RST" stands for the proposed RST-based algorithm, "KMs" stands for K-means algorithm and "EM" stands for the Expectation-Maximization algorithm.

8. Conclusion

The RST is a powerful mathematical tool for dealing with categorical data and uncertain information. We propose to apply a RST-based approach to cluster application layer network traffic and group protocol messages according to message types. The key of our approach is to consider multidimension categorical attributes based on rough sets theory and diminish the dissonance hidden in the data set. With the concepts introduced from the field of rough sets theory, the dissonance hidden in the data set can be quantified by the notion of roughness. The proposed approach aims to minimize the total roughness in the data set by selecting the clusters with the largest cluster roughness for further splitting in each iteration of clustering. The proposed approach is also unsupervised and the optimal number of clusters is determined by the Minimum Description Length principle. The experimental results show that our method can cluster the application layer payload with a high accuracy and group the protocol messages effectively to improve the accuracy of protocol keyword extraction. Some protocol keywords with low occurrence probability can be found with the help of message grouping by our method. In the future work, we will apply the hierarchical data structure and semantic information in the traffic to further improve the accuracy of traffic clustering.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by Guangdong Natural Science Foundation (Grant no. S2013040014339), Guangdong Natural Science Foundation (Grant no. 2014A030313637), Guangdong Provincial Department of Education Innovation Project (2014KTSCX149), the State Key Program of NSFC-Guangdong Joint Funds (U0735002), and the National Natural Science Foundation of China (Grant no. 60970146).

References

[1] W. Cui, J. Kannan, and H. J. Wang, "Discoverer: automatic protocol reverse engineering from network traces," in *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, pp. 1–14, USENIX Association, Berkeley, Calif, USA, 2007.

[2] Z. Lin, X. Zhang, and D. Xu, "Reverse engineering input syntactic structure from program execution and its applications," *IEEE Transactions on Software Engineering*, vol. 36, no. 5, pp. 688–703, 2010.

[3] J. Caballero and D. Song, "Automatic protocol reverse-engineering: message format extraction and field semantics inference," *Computer Networks*, vol. 57, no. 2, pp. 451–474, 2013.

[4] J.-Z. Luo and S.-Z. Yu, "Position-based automatic reverse engineering of network protocols," *Journal of Network and Computer Applications*, vol. 36, no. 3, pp. 1070–1077, 2013.

[5] H. C. Kim, Y. H. Choi, and D. H. Lee, "Efficient file fuzz testing using automated analysis of binary file format," *Journal of Systems Architecture*, vol. 57, no. 3, pp. 259–268, 2011.

[6] C. Y. Cho, D. Babić, C. E. R. Shin, and D. Song, "Inference and analysis of formal models of botnet command and control protocols," in *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS '10)*, pp. 426–439, October 2010.

[7] C. Leita, M. Dacier, and F. Massicotte, "Automatic handling of protocol dependencies and reaction to 0-day attacks with scriptgen based honeypots," in *Recent Advances in Intrusion Detection*, D. Zamboni and C. Kruegel, Eds., vol. 4219 of *Lecture Notes in Computer Science*, pp. 185–205, Springer, Berlin, Germany, 2006.

[8] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," in *Proceedings of the SIGCOMM Workshop on Mining Network Data (MineNet '06)*, pp. 281–286, ACM, New York, NY, USA, 2006.

[9] J. Erman, A. Mahanti, and M. Arlitt, "Internet traffic identification using machine learning techniques," in *Proceedings of the 49th IEEE Global Telecommunications Conference (GLOBECOM '06)*, pp. 1–6, IEEE, San Francisco, Calif, USA, December 2006.

[10] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson, "Semi-supervised network traffic classification," in *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '07)*, pp. 369–370, ACM, New York, NY, USA, June 2007.

[11] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, "Traffic classification on the fly," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 2, pp. 23–26, 2006.

[12] L. Bernaille and R. Teixeira, "Early recognition of encrypted applications," in *Passive and Active Network Measurement*, S. Uhlig, K. Papagiannaki, and O. Bonaventure, Eds., vol. 4427 of *Lecture Notes in Computer Science*, pp. 165–175, Springer, Berlin, Germany, 2007.

[13] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of Intelligent Information Systems*, vol. 17, no. 2-3, pp. 107–145, 2001.

[14] Z. Pawlak, "Rough classification," *International Journal of Man-Machine Studies*, vol. 20, no. 5, pp. 469–483, 1984.

[15] Z. Pawlak, "Rough sets," *International Journal of Computer and Information Sciences*, vol. 11, no. 5, pp. 341–356, 1982.

[16] B. Walczak and D. L. Massart, "Rough sets theory," *Chemometrics and Intelligent Laboratory Systems*, vol. 47, no. 1, pp. 1–16, 1999.

[17] D. Parmar, T. Wu, and J. Blackhurst, "MMR: an algorithm for clustering categorical data using Rough Set Theory," *Data and Knowledge Engineering*, vol. 63, no. 3, pp. 877–891, 2007.

[18] IANA, Internet Assigned Numbers Authority (IANA), 2012, http://www.iana.org/assignments/port-numbers.

[19] A. N. Mahmood, C. Leckie, and P. Udaya, "An efficient clustering scheme to exploit hierarchical data in network traffic analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 6, pp. 752–767, 2008.

[20] P. Lingras and G. Peters, "Applying rough set concepts to clustering," in *Rough Sets: Selected Methods and Applications in Management and Engineering*, Advanced Information and Knowledge Processing, pp. 23–37, Springer, London, UK, 2012.

[21] L. J. Mazlack, A. He, and Y. Zhu, "A rough set approach in choosing partitioning attributes," in *Proceedings of the 13th ISCA International Conference (CAINE '00)*, pp. 1–6, New Orleans, La, USA, March 2000.

[22] Y. Wang, Y. Xiang, and S.-Z. Yu, "Automatic application signature construction from unknown traffic," in *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA '10)*, pp. 1115–1120, April 2010.

[23] Y. Wang, Y. Xiang, and S.-Z. Yu, "An automatic application signature construction system for unknown traffic," *Concurrency Computation Practice and Experience*, vol. 22, no. 13, pp. 1927–1944, 2010.

[24] O. Georgieva, K. Tschumitschew, and F. Klawonn, "Cluster validity measures based on the minimum description length principle," in *Knowledge-Based and Intelligent Information and Engineering Systems*, vol. 6881 of *Lecture Notes in Computer Science*, pp. 82–89, Springer, Berlin, Germany, 2011.

[25] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.

[26] J. Rissanen, "Universal coding, information, prediction, and estimation," *IEEE Transactions on Information Theory*, vol. 30, no. 4, pp. 629–636, 1984.

[27] A. W. Moore and K. Papagiannaki, "Toward the accurate identification of network applications," in *Proceedings of the 6th International Conference on Passive and Active Network Measurement (PAM '05)*, pp. 41–54, 2005.

[28] M. Ye, K. Xu, J. Wu, and H. Po, "Autosig-automatically generating signatures for applications," in *Proceedings of the 9th IEEE International Conference on Computer and Information Technology (CIT '09)*, vol. 2, pp. 104–109, Xiamen, China, October 2009.

[29] P. Haffner, S. Sen, O. Spatscheck, and D. Wang, "ACAS: automated construction of application signatures," in *Proceedings of the ACM SIGCOMM 1st Workshop on Mining Network Data (MineNet '05)*, pp. 197–202, ACM, August 2005.

[30] B.-C. Park, Y. J. Won, M.-S. Kim, and J. W. Hong, "Towards automated application signature generation for traffic identification," in *Proceedings of the IEEE/IFIP Network Operations and Management Symposium: Pervasive Management for Ubiquitous Networks and Services (NOMS '08)*, pp. 160–167, April 2008.