Hindawi
Security and Communication Networks
Volume 2020, Article ID 8824659, 13 pages
https://doi.org/10.1155/2020/8824659

Research Article

Preprocessing Method for Encrypted Traffic Based on Semisupervised Clustering

Rongfeng Zheng,[1] Jiayong Liu,[2] Weina Niu,[3] Liang Liu,[2] Kai Li,[2] and Shan Liao[2]

[1]*College of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China*
[2]*College of Cybersecurity, Sichuan University, Chengdu 610065, China*
[3]*School of Computer Science and Engineering, Institute for Cyber Security, University of Electronic Science and Technology of China (UESTC), Chengdu 611731, China*

Correspondence should be addressed to Jiayong Liu; ljy@scu.edu.cn

Received 25 March 2020; Revised 25 May 2020; Accepted 8 July 2020; Published 27 July 2020

Academic Editor: Sajjad Shaukat

The explosive growth in network traffic in recent times has resulted in increased processing pressure on network intrusion detection systems. In addition, there is a lack of reliable methods for preprocessing network traffic generated by benign applications that do not steal users' data from their devices. To alleviate these problems, this study analyzed the differences between benign and malicious traffic produced by benign applications and malware, respectively. To fully express these differences, this study proposed a new set of statistical features for training a clustering model. Furthermore, to mine the communication channels generated by benign applications in batches, a semisupervised clustering method was adopted. Using a small number of labeled samples, our method aggregated historical network traffic into two types of clusters. The cluster that did not contain labeled malicious samples was regarded as a benign traffic cluster. The experimental results were compared using four types of clustering algorithms. The density-based spatial clustering of applications with noise (DBSCAN) clustering algorithm was selected to mine benign communication channels. We also compared our method with two other methods, and the results demonstrated that the benign channels mined through our method were more reliable. Finally, using our method, 1,811 benign transport layer security (TLS) channels were mined from 18,357 TLS communication channels. The number of flows carried by these benign channels comprised 65.37% of the entire network flows, and no malicious flow was included in our results, which proves the effectiveness of our method.

1. Introduction

Most of the communications making up internet traffic are generated by benign applications. If these communications are directly imported into the network intrusion detection system (NIDS) without any preprocessing, they invariably impose huge computational pressure on the NIDS. Therefore, the exclusion of benign network traffic in advance is a widely adopted strategy in the industry.

Before encryption technology was popularized on the internet, antimalware manufacturers could recognize network traffic using the deep packet inspection (DPI) method. However, with the popularization of the transport layer security (TLS) protocol [1], the malware also gradually adopted this protocol to complete command and control (C&C) communication. This led to a gradual failure in identifying network traffic using the DPI method. In response to this situation, some security vendors use the server name field or the domain name field in the certificate to preprocess the TLS traffic generated by benign applications. However, server names and certificates are easily forged by malware, making this preprocessing strategy unreliable.

Currently, the most efficient preprocessing method uses IP whitelisting technology. However, there is a lack of reliable IP whitelist sources on the internet. Threat intelligence communities such as AlienVault [2], IBM X-Force Exchange [3], and Recorded Future [4] usually provide available IP blacklist resources that are used by malware, but seldom

present whitelist resources. Furthermore, the IP whitelist usually needs to be updated occasionally to ensure its validity. Therefore, it is necessary to provide a fast and reliable method for collecting IP whitelist.

The reverse domain name lookup method can be used to obtain benign IP addresses based on the collection of benign domain names. However, under the primary domain name, there are usually many subdomains that bind to different IP addresses. As a result, it is difficult to enumerate all of the subdomains. Nevertheless, because the domain name whitelists are usually on the order of millions, using the reverse domain lookup method to collect the IP whitelist is significantly inefficient. In addition, the reliability of the IP whitelist is based on the reliability of the domain name whitelist, which makes it difficult to guarantee that all IP addresses in the whitelist are benign.

In recent years, classification and clustering techniques based on machine learning have been widely used in the identification of encrypted communication traffic. Additionally, there have been numerous studies on the application of coarse classification models [5–7] and clustering models [8–11] for preprocessing network traffic. Classification models usually require a large number of labeled samples for training, which results in the improved ability of the classification model to identify the trained samples. In other words, in the classification results of the coarse classification model for identifying malicious traffic, the nonmalicious class cannot be regarded as the benign class, because it may contain untrained malicious network traffic. For the clustering model, it is difficult to ensure the purity of the samples in the clusters based only on the single flow-based features [12]. In particular, if the benign cluster contains malicious samples, it will produce many false negatives in the NIDS.

Our study demonstrates that there are many differences between benign and malicious samples in TLS communication channels, such as the amount of inbound and outbound traffic, the connected devices, and the communication frequency. Based on these differences, the features of TLS communication channels can be extracted and a clustering model for benign applications can be established. Our network traffic preprocessing method can be realized by excluding the benign traffic contained in a cluster. Therefore, the contributions of this study are as follows:

(1) This study proposes a new network traffic preprocessing method based on a semisupervised model. To distinguish it from the traditional single flow-based features, this study presents a new set of statistical features for building a clustering model based on the TLS communication channels.

(2) This study proposes a new feature selection method. In the proposed method, the spectral clustering feature selection algorithm is used to select the top-200 features based on unlabeled samples. Further, by redesigning the evaluation algorithm, the wrapper method based on a semiunsupervised model is used to further select the best performing feature subset.

(3) Experimental results show that the preprocessing method proposed in this study can identify 1,811 benign TLS communication channels from 18,357 TLS communication channels. These channels carry 65.37% of the entire TLS flows. Furthermore, through contrast experiments, the proposed preprocessing method was verified to perform better than two other machine learning-based methods.

2. Related Work

The geometric growth in network traffic in recent times has resulted in increased processing pressure in the detection of malicious communication. Although the traditional preprocessing method based on DPI technology can accurately identify unencrypted communication traffic, it cannot cope with the currently increasing encrypted communication traffic. Machine learning technology has been used extensively in the identification of encrypted traffic and is mainly divided into classification and clustering algorithms. By applying a supervised learning algorithm, the coarse classification model [5–7] is used to preprocess network traffic, whereas the fine classification model [13–15] is used to identify the type of network traffic accurately. The clustering model [8–11] based on unsupervised learning algorithms is mainly used to identify unknown network applications and can also be used as a preprocessing method.

2.1. Supervised Learning Model. The coarse classification model is usually used for preprocessing before identifying the type of network traffic. Zhao [5] proposed a three-layer classifier to detect known and unknown network traffic. The first layer consists of a coarse classification model (a binary classifier), which is mainly used to quickly identify the unknown network traffic and thus reduce the processing pressure of the fine classification model. A similar process can be seen whereby the coarse classification model of the first layer is mainly used to exclude the benign traffic, the second layer is used to classify the different types of malicious traffic, and the third layer is used to identify different malware families [6]. However, in this process, the preprocessing method of the first layer is not described in detail, and its impacts on the detection results are rarely evaluated. To deal with the problem of accurately identifying abnormal network traffic, a two-stage deep learning detection model [7] has been proposed. This method introduced a detailed design scheme of the first stage's binary model and used the probability score value calculated from the binary model as the second stage's input. Experiments show that this method has an accuracy rate of 99.996% for the KDD99 dataset and 89.134% for the UNSW-NB15 dataset, which are higher than those of other current methods. However, because such methods do not exclude normal samples in the first stage, they reduce the efficiency of the entire detection model. Additionally, all the methods that are based on the supervised model process network flow individually and cannot process network flows in batches, which leaves room for the further improvement of preprocessing efficiency.

2.2. Unsupervised Learning Model. Unsupervised learning algorithms are also widely adopted in the preprocessing of network traffic. Zhang [9] used an unsupervised model to preprocess the network traffic to find zero-day network traffic clusters. They then used the zero-day traffic and labeled samples to train a binary classifier that was used to identify zero-day traffic more effectively. Experiments show that their method can more accurately recognize known network applications and also identify zero-day network applications. Similar methods can also be found in the research of Zhao [10]. They used a clustering model to achieve two preprocessing goals, namely, screening out unknown network traffic and expanding more labeled samples based on a few known samples. Experiments show that their method can improve classification accuracy after the preprocessing step. The research of Sacramento [11] is based on the assumption that most network traffic is benign, and only a small part of it is malicious traffic. During their preprocessing step, the largest cluster was considered to be a benign cluster, while the smaller clusters needed to be further analyzed. Through experiments, a variety of network attack behaviors were detected. However, their impact on the detection effect was not evaluated after carrying out the preprocessing step. Liya [8] used a hierarchical clustering model to preprocess a set of samples. First, they divided the set of samples into multiple clusters, after which they selected several representative samples from each cluster. Then, they used the Bayesian algorithm to classify these flows. The classification results of the selected flows represented the classification results of the entire cluster. Thus, network traffic could be quickly processed. Although the clustering model can be used to preprocess network traffic in batches, most of the current research is based on the single flow-based feature, which cannot guarantee the purity of the clusters. The sample purity in some clusters can only reach 35% [12]. Therefore, the current preprocessing method based on the clustering model can be further improved.

2.3. Single Flow-Based and Multiple Flow-Based Feature. Presently, in the field of network traffic classification, most studies focus on single flow-based features. A single flow is composed of packets with the same five-tuple information. Gezer [16] mainly extracted the single flow-based features from multiple dimensions (including the duration of the flow, the maximum, minimum, and average packet length), the interarrival time of the flow, and the number of inbound and outbound packets. Korczyński and Duda [17] used the sequence of packet length to build a Markov model. Yang [14] proposed the packet length and interpackets arrival time's distribution features in a flow. These features can be regarded as single flow-based features and are commonly used in most network traffic classification experiments.

Multiple flow-based features usually represent those that are extracted from multiple flows produced in a sliding time window. In a study on the identification of proxy application traffic based on the characteristic of flow bursts in a short time window [18], these features were designed to include the number of flow bursts, the maximum flow burst lengths, and the sum of all flow burst lengths. To detect network intrusion behaviors, Patil [19] not only applied single flow-based features but also added some multiple flow-based features, such as the number of flows with the same source IP address, and the number of flows with the same destination IP address. The application of these multiple flow-based features in different traffic classification scenarios allows for the improved performance of traffic classification.

Numerous studies [9, 20, 21] have utilized the concept of a bag of flow (BoF). A BoF is a set of flows with the same destination IP address, destination port, and transport protocol, which represents the network traffic generated by the same server application on the same port over time. As long as the type of a certain flow can be determined, the BoF type can also be determined. These studies regard a BoF as the total of flows generated on an application's communication channel. As mentioned earlier, most of the current traffic classification studies are based on single flow-based features. To the best of our knowledge, no feature design is based purely on applications' communication channels. This study sought to find the difference between the benign application and the malware on the TLS communication channel and propose a preprocessing method to exclude the benign TLS traffic. Applying this method to the NIDS can significantly reduce the processing pressure on the detection system.

3. Benign Traffic Characteristics

Before introducing feature design, it is necessary to analyze the behavioral differences between benign applications and malware on the TLS communication channel. We considered flows with the same destination IP address or server IP address and destination port, namely, port 443, as flows on the same communication channel. There are many differences between the communication behaviors of benign applications and those of malware. Firstly, in the transmission direction, benign applications usually initiate a request to the server and obtain resources, such as text, picture, audio, and video data from the server. The transmission payload is concentrated from the server to the client, which is the inbound direction. For example, by analyzing its historical traffic records, github.com has a total of 19 TLS communication channels (19 independent destination IP addresses), which carry a total of 4,080 network flows. Further analysis shows that there are a total of 3,933 flows and that the inbound payload is greater than the outbound payload, which accounts for 96.40% of the total flows. However, malware that focuses on stealing information from the users' host usually produces more outbound traffic. By collecting large amounts of traffic samples as described in Section 4, we compared the inbound and outbound traffic differences between benign and malicious application samples on their communication channels. Figure 1 shows the distribution of the outbound and inbound payload sizes using the same number of samples.

As shown in Figure 1, the abscissa represents the inbound payload sizes, and the ordinate represents the outbound payload sizes. Red dots represent malicious samples

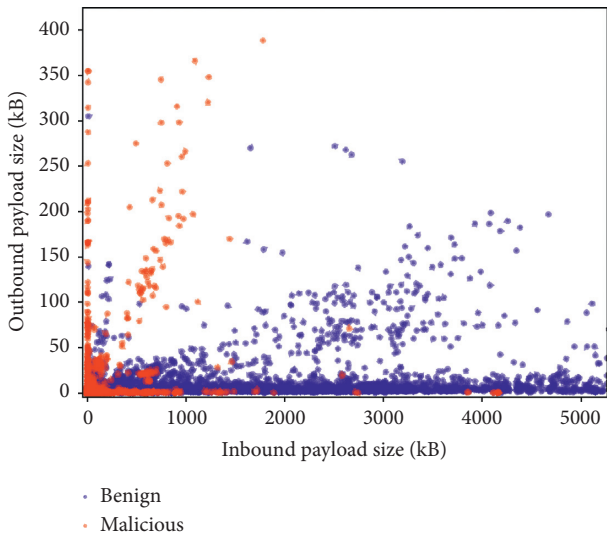

Figure 1: Distribution of inbound and outbound payload sizes.

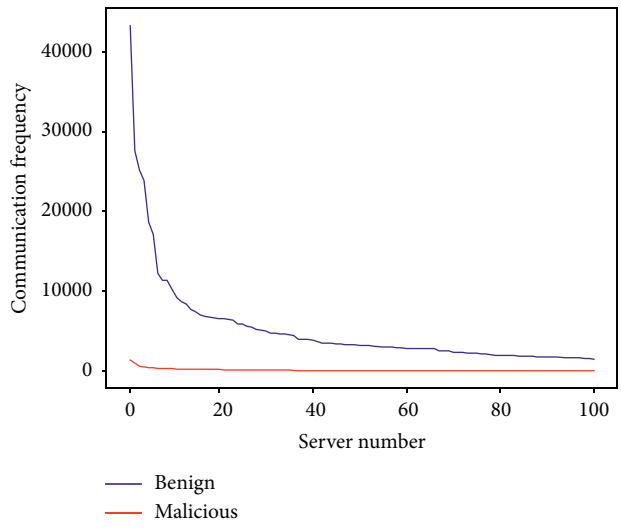

Figure 2: Top 100 servers in terms of communication frequency.

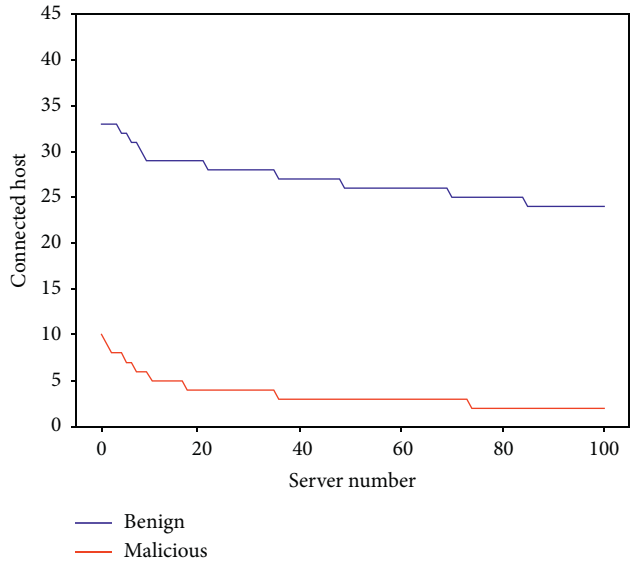

Figure 3: Top 100 servers in terms of connected devices.

and blue dots represent benign samples. It can be seen that malicious samples are more distributed near the vertical axis, whereas benign samples are mostly distributed near the horizontal axis. The proportion of the malicious flows is insignificant when the inbound payload size is higher than 1,000 kB.

Additionally, as a result of the users' online habits, some benign applications are frequently used to obtain resources from the server. This means that the communication frequency to benign application servers is higher than that of malware C&C servers. Because concealment is put first by malware, the frequent connection to the C&C server should be avoided. Figure 2 shows the top 100 servers in terms of communication frequency for both benign and malicious samples.

In addition to some niche applications, other applications have a specific user base, which results in more hosts accessing certain servers. On the contrary, malware generally chooses high-value targets for infection, so that there are relatively fewer devices that connect to the C&C servers in the local area network. Figure 3 shows the top 100 servers in terms of connected devices.

Based on the above analysis of the differences between benign and malicious traffic and their characteristics, we can categorize the network traffic into two types of clusters. The first type is characterized by a larger inbound payload, a higher communication frequency, and more connected devices. This type of cluster can be regarded as the network traffic generated by benign applications. However, theoretically, the other type of cluster contains not only malicious traffic but also traffic generated by some benign niche software.

4. Feature Representation

Based on the analysis of the characteristics of benign traffic mentioned in the previous section, this section mainly introduces the statistical features from five aspects, namely, inbound payload size, outbound payload size, inbound and outbound payload ratio, communication frequency, and connected devices. However, in terms of statistical features, the statistical feature values among benign applications vary significantly. For example, the inbound payload size of some benign applications, such as youtube.com, can reach the gigabyte (GB) scale, while the inbound payload size of other benign applications, such as baidu.com are only at the kilobyte (kB) scale. At the same time, other statistical features show the same problem. Therefore, quantization is needed before extracting features.

The quantitative scheme adopted in this study is divided into three steps. The first step is to set granularity. Based on the historical network traffic records, different statistical intervals can be divided according to different granularities, and these statistical intervals can represent the candidate features.

The second step is to calculate and normalize the feature values on each interval. For a server, there may be many instances of communication behavior, and the statistical value, such as payload size, is different each time. Therefore, we need to map these values into different statistical intervals and use the number of mapping times to calculate the feature value. Figure 4 shows a mapping process for calculating the feature value.

The third step is to compare the differences between benign and malicious samples in each interval and select the training features. To select more appropriate features, we used 5 kB, 10 kB, 50 kB, and 500 kB as the granularities for dividing intervals and compared the proportion of the inbound payload size between the benign and malicious samples in each interval using the same sample size.

As shown in Figure 5, benign and malicious samples show great differences in these intervals. It can be seen that the inbound payload size of malware is mainly concentrated in the 0–5 kB interval, accounting for more than 80% of the total traffic. When the interval is greater than 5 kB, the proportion of the malware is always smaller than that of benign applications. When the interval is greater than 500 kB, the proportion of malicious flow accounts for only 1.30%, which is almost negligible. Other features like outbound payload size, inbound/outbound payload ratio (in- and out-payload ratio), communication frequency, and connected devices show the same trend as shown in Figure 6.

Therefore, for these statistical features, it is feasible to divide the statistics according to different granularities. We designed our feature set, as shown in Table 1, which contains a total of 500 features.

The feature set in Table 1 mainly describes the network behavior in a TLS communication channel. This feature set is completely different from the traditional single flow-based feature set described in other studies [14, 16, 17]. The feature design used in this study can express the accessing behavior of a certain server from a higher level and bring together TLS communication channels with similar network behaviors.

4.1. Spectral Feature Selection Algorithm. This study adopted the spectral feature selection algorithm (SPEC) [22] to select relevant features for an unlabeled sample set. SPEC is a feature selection algorithm based on spectral graph theory [23]. The theory of SPEC is not complex and its performance is superior to other algorithms such as Laplacian Score. It can be used for both supervised and unsupervised feature selection. SPEC calculates the relevance of a feature by evaluating the feature consistency of the spectral matrix derived from the similarity matrix S. The similarity between the two samples x_i and x_j is evaluated using a radial basis function (RBF):

$$S_{ij} = e^{-\left(\left\|x_i - x_j\right\|^2 / 2\sigma^2\right)}. \tag{1}$$

By calculating S_{ij}, the similarity matrix $\mathbb{S}$ can be constructed to represent the relationships among samples. Given $\mathbb{S}$, the undirected graph G and adjacency matrix W can be constructed, where $W(i, j) = w_{ij}$ and the weight w_{ij} is determined by $\mathbb{S}$. The defining vector $\mathbf{d} = \{d_1, d_2, d_3, \ldots, d_n\}$, where $d_i = \sum_{k=1}^{n} w_{ik}$. The degree matrix D is defined as follows: $D(i, j) = d_i$, if $i = j$, and zero otherwise. Given the adjacency with W and the degree matrix D, the Laplacian matrix L and the normalized Laplacian matrix $\mathscr{L}$ are defined as follows:

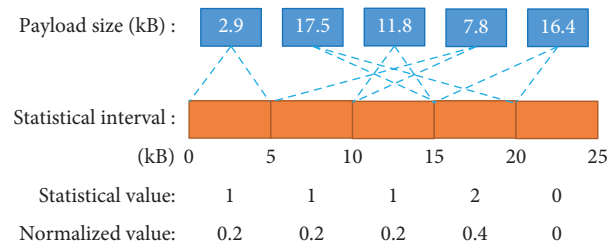

Figure 4: Mapping payload sizes to statistical intervals.

$$\begin{aligned} L &= D - W, \\ \mathscr{L} &= D^{-(1/2)} L D^{-(1/2)}. \end{aligned} \tag{2}$$

The weight of each feature vector f_i can be obtained using three ranking functions, namely, $\varphi_1, \varphi_2, \varphi_3$. Considering that the φ_2 function performs better on the test set used in Zheng and Huan's research study [22], this study selects the φ_2 function as the ranking function:

$$\varphi_2(F_i) = \frac{f_i'^T \mathscr{L} f_i'}{1 - f_i'^T \xi_0}, \tag{3}$$

in equation (3), where $f_i' = (D^{1/2} f_i)$, F_i represents the i-th feature. Given the normalized Laplacian matrix L, its spectral decomposition (λ_i, ξ_i) can be calculated, where λ_i is the eigenvalue and ξ_i is the eigenvector. According to spectral graph theory [23], we have the following: $\lambda_0 = 0$ and $\xi_0 = D^{1/2} e$. Accordingly, using the ranking function φ_2, the weight of a feature can be readily calculated. The entire calculation process is shown in Algorithm 1 which can be used to obtain the top k relevant features. There are three steps in the feature selection process: (1) building similarity set $\mathbb{S}$ and constructing its graph representation according to equations (1) and (3) (lines 1–3); (2) calculating $\varphi_2(F_i)$ according to equation (3) (lines 4–6); (3) ranking features in ascending order for $\varphi_2(F_i)$ (lines 7-8). In fact, a smaller $\varphi_2(F_i)$ represents the improved separability among samples. Hence, the smaller the value of $\varphi_2(F_i)$, the more important the feature f_i is.

To evaluate the selected feature set, we prepared 6,978 samples and used Algorithm 1 to rank the importance of the features. Table 2 shows an example of the top 20 features. It can be seen that the features of the downstream payload size have the highest proportion, meaning that these types of features are the most important.

We selected the top 200-feature subsets as our candidate feature subsets by using Algorithm 1 to quickly exclude ineffective features due to the sparsity of the designed feature set. To further select the relevant features, we used the wrapper method to evaluate whether the feature subset could meet the requirements of clustering. This part can be seen in the next section.

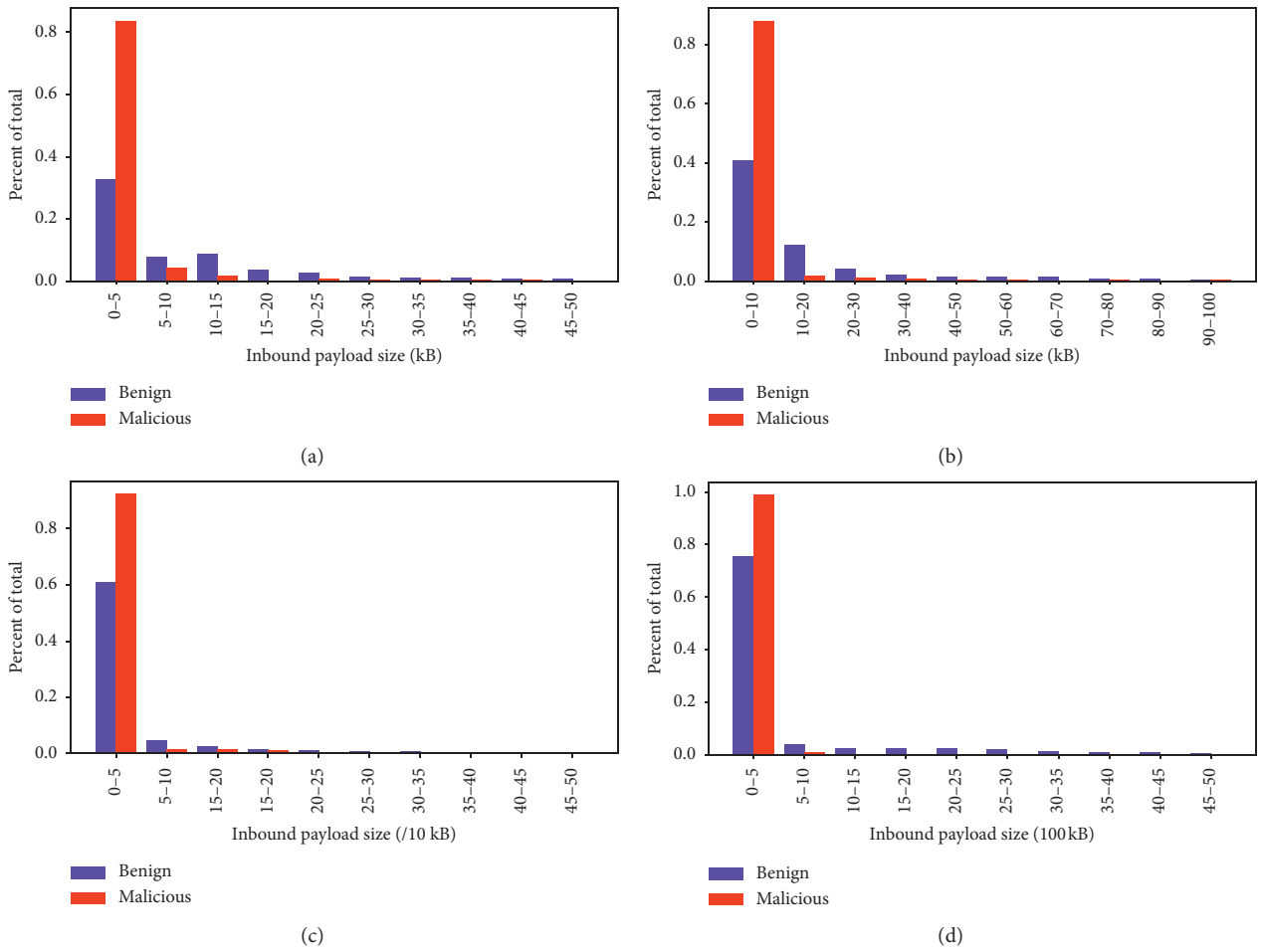

Figure 5: Proportions in different intervals.

5. Model Training

The goal of preprocessing is to mine the TLS communication channels used by benign applications as much as possible. However, the feature subset selected by the SPEC algorithm did not necessarily meet the requirements of the clustering model; i.e., the cluster of benign TLS channels should contain as few malicious TLS channels as possible. Figure 7 shows an example of the description of different clustering effects that originated from different feature subsets. Labeled samples are represented by + and – where + indicates benign TLS samples and – indicates malicious TLS samples. Unlabeled samples are represented by △. F_i and F_i' are used to denote different feature subsets. In the clustering process, we labeled the attribute of a cluster by calculating its proportion of positive and negative samples. If the proportion of positive samples in the cluster is higher, the cluster is considered to be benign; otherwise, it is marked as malicious.

We did not require distinguishing the two types of samples. We only needed to ensure that the benign cluster contained as few malicious samples as possible. In Figure 7, although the clustering results based on F_i' were better, we preferred the clustering result based on F_i to ensure the purity of the benign cluster. Hence, we needed to redesign the evaluation algorithm used in the process of training the model. In this process, two goals can be achieved simultaneously: (1) selecting the best performance feature subset; (2) selecting a more appropriate clustering algorithm.

Semisupervised clustering was adopted to evaluate the clustering effect, and the wrapper method was used to select a subset of the best performing features based on the ranked top 200-feature sets obtained in the previous section. Two rules were used to evaluate the performance of each subset of features: (1) whether the subset of features reduced the proportion of malicious channels in the benign cluster; (2) whether the subset of features could improve the recognition rate of benign channels when the proportion of malicious channels in the benign cluster did not change. We evaluated the performance of the feature subsets by calculating two indicators of the labeled samples: the false positive rate (FPR) and the true positive rate (TPR). The evaluation algorithm is outlined in Algorithm 2.

Algorithm 2 is divided into three main steps: (1) selecting a clustering algorithm and calculating the confusion matrix of the X_{labeled} (labeled samples) to obtain the initial FPR and TPR (lines 1-2); (2) constructing feature subsets using backward selection and judging whether a feature should be excluded by comparing FPR and TPR with the last result (lines 3–13); (3) obtaining the final feature subset and clustering result of the labeled and unlabeled samples (lines

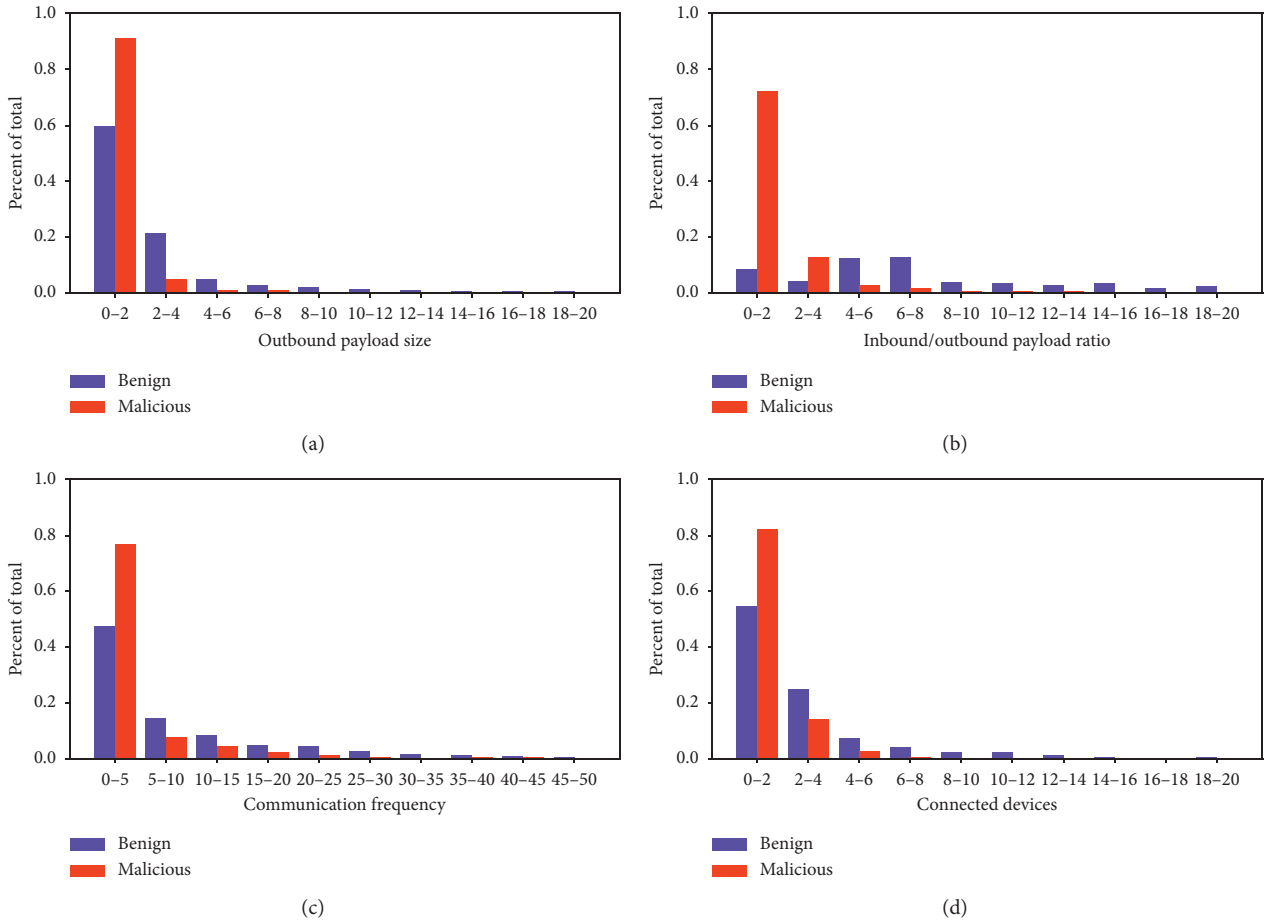

Figure 6: Comparison of other feature expressions.

Table 1: Designed feature set.

Feature name	Description	Number
Inbound payload size	Interval granularities with 2, 3, 5, 10, 20, 50, 100, 200, 500, 5000 kB	100
Outbound payload size	Interval granularities with 2, 3, 5, 10, 20, 50,100, 200, 500, 5000 kB	100
In-and-out payload ratio	Interval granularities with 2, 3, 5, 10, 20, 50, 100, 200, 500, 5000	100
Communication frequency	Interval granularities with 2, 3, 5, 10, 20, 50, 100, 200, 500, 5000	100
Connected devices	Interval granularities with 2, 3, 5, 10, ,20, 50, 100, 200, 500, 5000	100

Input: X, $\gamma(\cdot)$, k, F_i
Output: $\mathrm{SF}_{\mathrm{SPEC}}$–the ranked feature list
(1) construct $\mathbb{S}$, the similarity set from X (and Y);
(2) construct graph G from $\mathbb{S}$;
(3) build W, D and L from G ;
(4) **for** each feature vector f_i **do**
(5) $f_i' \leftarrow (D^{1/2} f_i / \|D^{1/2} f_i\|)$; $\mathrm{SF}_{\mathrm{SPEC}}(i) \leftarrow \varphi_2(F_i)$
(6) **end**
(7) ranking $\mathrm{SF}_{\mathrm{SPEC}}$ in ascending order for $\varphi_2(F_i)$
(8) return $\mathrm{SF}_{\mathrm{SPEC}}$.

Algorithm 1: SPEC.

14-15). Additionally, by comparing the clustering effects of different clustering algorithms, the best performing clustering algorithm can also be obtained.

6. Experiment and Evaluation

The experiment was composed of three steps: (1) data collection and pretreatment; (2) algorithm selection, whereby the best performing algorithm could be selected from the four clustering algorithms; (3) method comparison. The method proposed in this study was compared with other existing methods and their effectiveness evaluated.

6.1. Data Collection. The test data of TLS traffic used in this method were collected from the gateway of our laboratory; we mirrored all network traffic including TLS flows to our experimental platform. In this study, we only used TLS traffic to mine TLS benign communication channels.

The network flow was collected using a tool developed by us [24]. The basic information of the flow included the

Table 2: Top 20 features.

Feature description	$\varphi_2(F_i)$
Proportion of payload ratio in [200, 400) to the total	0.5381
Proportion of payload ratio in [8, 10) to the total	0.5421
Proportion of in-payload size in [0, 2) kB to the total	0.5433
Proportion of payload ratio in [30, 40) to the total	0.5436
Proportion of out-payload size in [20, 40) kB to the total	0.5451
Proportion of in-payload size in [40000, 45000) kB to the total	0.5473
Proportion of in-payload size in [25000, 30000) kB to the total	0.5531
Proportion of payload ratio in [150, 200) to the total	0.5553
Proportion of in-payload size in [450, 500) kB to the total	0.5558
Proportion of in-payload size in [1500, 2000) kB to the total	0.5572
Proportion of payload ratio in [500, 1000) to the total	0.5575
If the communication frequency in [20, 40)	0.5642
Proportion of payload ratio in [6, 8) to the total	0.5656
Proportion of in-payload size in [45000, 50000) kB to the total	0.5663
Proportion of out-payload size in [16, 18) kB to the total	0.5664
If the communication frequency in [200, 400)	0.5670
Proportion of out-payload size in [90, 100) kB to the total	0.5704
If the communication frequency in [70, 80)	0.5726
Proportion of payload ratio in [18, 20) to the total	0.5735
Proportion of in-payload size in [50, 60) kB to the total	0.5738

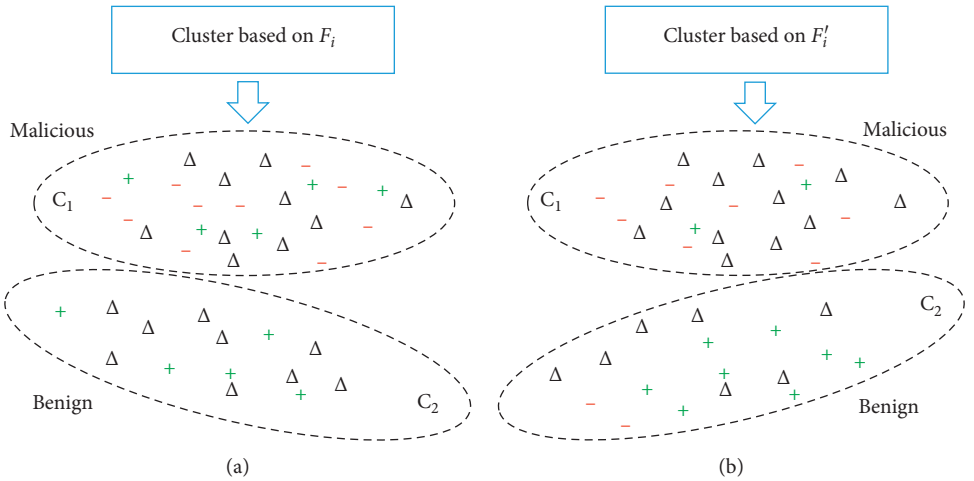

Figure 7: Comparison of clustering results using different feature subsets.

Input: $F_{\text{top-200}}$, X_{labeled}, $X_{\text{unlabeled}}$
Output: Best feature subset (BFS), Clustering result (CR)
(1) select a cluster algorithm
(2) calculate initial confusion matrix for X_{labeled}
obtain FPR_{ini}, TPR_{ini}
(3) **for** backward selection $F_{\text{top-200}}$ and exclude f_i **do**
(4) calculate confusion matrix for X_{labeled}
(5) **if** $\text{FPR} < \text{FPR}_{\text{last}}$:
(6) exclude (f_i)
(7) $\text{FPR}_{\text{last}} = \text{FPR}$, $\text{TPR}_{\text{last}} = TPR$
(8) **if** $\text{FPR} = \text{FPR}_{\text{last}}$ and $\text{TPR} \geq TPR_{\text{last}}$:
(9) exclude (f_i)
(10) $\text{FPR}_{\text{last}} = \text{FPR}$, $\text{TPR}_{\text{last}} = \text{TPR}$
(11) **if** $\text{FPR} > \text{FPR}_{\text{last}}$:
(12) retain (f_i)
(13) **end**
(14) obtain feature subset as BFS, CR for $X_{\text{unlabeled}}$ and X_{labeled}
(15) return BFS, CR.

Algorithm 2: SMFS.

source IP address, destination IP address, inbound and outbound payload size, number of inbound and outbound packets, and the inbound/outbound payload ratio. This information on network flow can also be acquired using tools such as NetFlow [25], which was developed by Cisco or Moloch [26].

To verify this method, we collected all the network traffic based on the TLS protocol generated from July 1, 2019, to July 15, 2019. We collected a total of 1,655,498 TLS flows containing 18,357 TLS communication channels (18,357 unique destination IP addresses). In the experiment, we mined the benign TLS channels from the abovementioned 18,357 communication channels. In addition, this study also used the malicious traffic samples provided by Stratosphere Lab [27]. We downloaded a total of over 300 GB in traffic samples and extracted 14,544 TLS flows generated by malware, which makes up a total of 970 TLS communication channels. These malware samples were mainly used to verify the reliability of benign TLS channels mined using this method.

Before the experiments, it was necessary to carry out some pretreatment work on the samples to improve the efficiency of our method and reduce noise samples. Some channels with low accessing behaviors and low payload sizes were excluded in advance. The pretreatment rules were as follows: (1) The total number of access behaviors must not exceed 20 times; (2) The inbound payload size must not exceed 40 kB. Samples that satisfied these two rules were filtered out in advance. After pretreatment, we had 6,700 test samples and 278 labeled malicious samples remaining.

6.2. Algorithm Selection. The preprocessing method proposed in this study adopted a semisupervised clustering method to mine the benign TLS channels. The requirements for clustering were clear. We only needed to train two clusters. The first cluster was the cluster that conformed to the benign communication characteristics described in Section 2. This cluster was mainly composed of benign TLS channels. The other cluster was composed of malicious TLS channels and also contained some benign TLS channels generated by niche applications. Additionally, it should be noted that the method used in this study focused on clustering part of the benign TLS channels and not all the benign TLS channels. The reason is that some benign applications also have similar communication behaviors as malware.

To evaluate the effectiveness of the clustering algorithm, we selected 297 representative benign samples and 278 malicious samples and mixed these labeled samples with unlabeled samples for training the model. We used labeled samples to determine the cluster in which the benign samples were located and to verify whether the method could effectively mine benign TLS channels.

Because different sample sets were adapted to different types of clustering algorithms, we needed to determine the clustering algorithm that was more suitable for training the samples. Four clustering algorithms were selected for comparative experiments, namely, K-means [28], based on the central point of samples; DBSCAN [29], based on the density of samples; Gaussian Mixture Models (GMM) [30], based on the distribution of samples; and Unweighted Pair Group Method with Arithmetic Mean (UPGMA) [31], which is a hierarchical clustering algorithm. We used Algorithm 2 to test these four clustering algorithms and compared the clustering results among them. Table 3 shows the clustering results of the four clustering algorithms.

It can be seen that the UPGMA and GMM clustering algorithms cannot effectively distinguish the labeled samples, indicating that the characteristics of the sample set are not suitable for these two clustering algorithms. The K-means clustering algorithm can recognize all the labeled malicious samples. However, a majority of the labeled benign samples are also identified as malicious. Only 95 of the 297 benign samples are correctly distinguished. The DBSCAN algorithm has the best clustering effect, whereby 204 benign samples are correctly recognized, and the recognition rate of labeled malicious samples reaches 100%. This means that all the labeled malicious samples are in the same cluster, which guarantees the purity of the benign samples contained in other clusters. Additionally, the testing set contained 6,700 TLS channels, with 4,889 in the malicious cluster, and 1,811 in the benign cluster. In other words, we mined 1,811 benign TLS channels that can be excluded through preprocessing. In the DBSCAN clustering algorithm, the selected best feature subset is shown in Table 4, with a total of 25 features.

6.3. Method Comparison and Evaluation. To further verify the effectiveness of our method, we compared it with a classification method proposed by Anderson et al. [13] and a clustering method proposed by Su [8], respectively. These methods are used to recognize benign traffic and mine benign TLS channels.

In the process of reproducing the method proposed by Anderson [13], we first selected 11,500 malicious samples and 11,500 benign samples. We then completed the feature extraction according to the feature set given in the method. The logistic regression algorithm (the random forest algorithm works best in actual tests) was used to train the sample set and evaluate the model using 10-fold cross-validation. The average accuracy of the classification model was 94.44%, and the recall rate was 89%. The test results achieved the effect described in the literature [13].

In the method proposed by Su [8], they built a hierarchical clustering process. In their method, the 7-dimensional structural features were first used for coarse-grained traffic clustering, after which the 7-dimensional temporal features were used for fine-grained traffic clustering. Finally, using Naïve Bayes classifiers, a small number of samples were selected from each cluster, and the classification results of these samples represented the classification results of the entire cluster. Therefore, their methods can also be used to identify benign network traffic.

In the first comparative experiment, we also used 297 labeled benign samples and 278 malicious samples as our experimental dataset. Our method used a feature set derived from TLS communication channels, whereas Anderson et al.

Table 3: Comparison of the clustering results of the four clustering algorithms.

Clustering algorithm	Testing samples		Labeled benign samples		Labeled malicious samples		Features in BFS
	Benign	Malicious	Benign	Malicious	Benign	Malicious	
K-means	733	5967	95	202	0	278	82
DBSCAN	1811	4889	204	93	0	278	25
GMM	4115	2385	229	68	129	149	93
UPGMA	2	6698	0	297	0	278	39

Table 4: Subset of features in DBSCAN.

Feature description	Category
Proportion of in-payload size in [10, 15) kB to the total	Numerical
Proportion of in-payload size in [15, 20) kB to the total	Numerical
Proportion of in-payload size in [20, 30) kB to the total	Numerical
Proportion of in-payload size in [0, 500) kB to the total	Numerical
Proportion of in-payload size in [500, 1500) kB to the total	Numerical
Proportion of in-payload size in [1500, 2000) kB to the total	Numerical
Proportion of out-payload size in [5, 10) kB to the total	Numerical
Proportion of out-payload size in [20, 30) kB to the total	Numerical
Proportion of out-payload size in [50, 100) kB to the total	Numerical
Proportion of out-payload size in [300, 400) kB to the total	Numerical
Proportion of payload ratio in [5, 10) to the total	Numerical
Proportion of payload ratio in [10, 20)to the total	Numerical
Proportion of payload ratio in [30, 40) to the total	Numerical
Proportion of payload ratio in [150, 200) to the total	Numerical
Proportion of payload ratio in [200, 400) to the total	Numerical
Proportion of payload ratio in [500, 1000) to the total	Numerical
If the communication frequency in [20, 40)	Boolean
If the communication frequency in [40, 60)	Boolean
If the communication frequency in [70, 80)	Boolean
If the communication frequency in [100, 150)	Boolean
If the communication frequency in [200, 300)	Boolean
If the communication frequency in [400, 500)	Boolean
If the connected hosts in [3, 6)	Boolean
If the connected hosts in [9, 12)	Boolean
If the connected hosts in [15, 20)	Boolean

and Su et al. used feature sets based on a single flow in their proposed methods. Therefore, in the methods of Anderson [13] and Su [8], we stipulate that in a communication channel, as long as any of the flows is determined to be malicious, the entire communication channel is untrustworthy. By reproducing these two methods, we obtained the results of the comparative experiments as shown below.

As shown in Figure 8, the A-Method represents the method proposed by Anderson et al. [13], and the S-Method represents the method proposed by Su et al. [8]. It can be seen that the A-Method has the highest recognition rate for benign samples, reaching 97.97%. However, it cannot identify all malicious samples, which means that the A-Method is not reliable. Both the S-Method and our method can identify all malicious samples, but the S-Method has a lower recognition rate for benign samples. Our method can not only cluster all the malicious samples but also have the highest recognition rate for benign samples.

In the next experiment, we compared the ability of these three methods to mine the benign TLS channels based on the test set. At the same time, we used open-source threat intelligence and manual inspection methods to evaluate the candidate benign TLS channels by checking whether the server IP address contained in the TLS channels was used by malware.

We selected the AlienVault [2] threat intelligence community, which owns a significantly comprehensive threat intelligence library that can be used to determine whether a server IP address is malicious. In AlienVault, some of the benign IP addresses, such as 8.8.8.8 (Google Public DNS), are also considered to be malicious (This is related to AlienVault's strategy for collecting the indicator of compromise (IoC). Further discussion is not required here). Hence, manual inspection is also needed to evaluate whether the malicious results provided by AlienVault are correct. It is worth mentioning that we cannot directly mine the benign TLS channels using AlienVault because it is unable to collect all malicious IP addresses worldwide. Another reason is that some benign IP addresses are also marked as malicious. However, AlienVault can reflect the reliability of the mined benign TLS channels to some extent. The test results are shown in Table 5.

It can be seen that the number of candidate benign channels mined using the A-Method is the largest, reaching

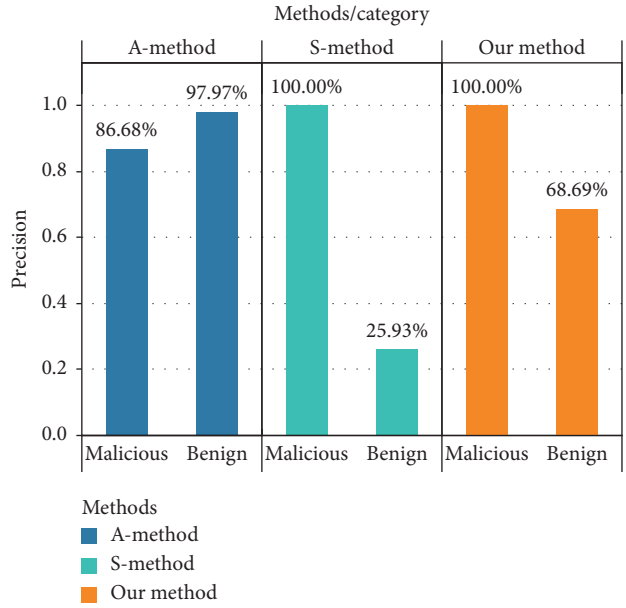

Figure 8: Comparison of different methods used on the labeled dataset.

Table 5: Evaluation of benign channels mined using different methods.

Benign channels	A-method	S-method	Our method
Candidate benign channels	11,286	1,722	1,811
Checking by AlienVault	11,166	1,699	1,779
Manual inspection	11,261	1,716	1,811

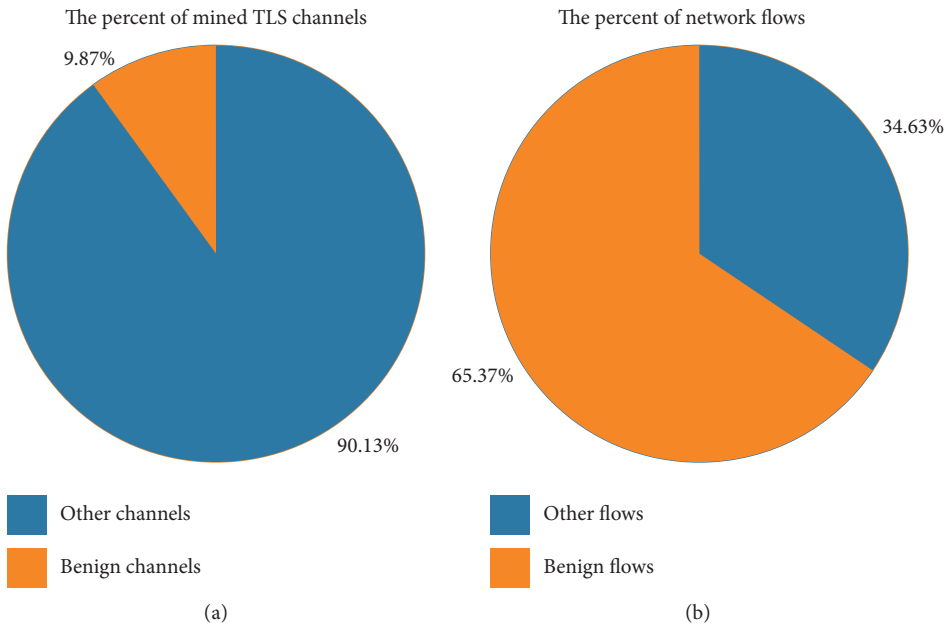

Figure 9: Statistics related to the benign TLS channels.

11,286, and that of benign channels mined using the S-Method is the least. Our method's results are slightly higher than those of the S-Method. However, through the step of checking using AlienVault, the candidate benign channels mined using the A-Method fall the fastest from 11,286 to 11,166. Manual inspection demonstrates that these IP addresses are mostly benign. However, there are still 25 malicious IP addresses in the candidate benign channels mined using the A-Method. We also found six malicious IP addresses mined using the S-Method. The candidate benign

channels mined using our method did not contain any malicious IP addresses. Therefore, the results show that our method is more reliable than the other two methods.

Using our method, we mined 1,811 benign channels from 18,357 channels. These channels account for 9.87% of the total mined IP addresses, as shown in Figure 9. The number of network flows carried by these 1,811 channels accounts for 65.37% of the total number of network flows. In other words, applying our preprocessing method in the NIDS can reduce the processing pressure by at least 65.37%. Other TLS channels are not all malicious because the characteristics of some niche benign applications are inconsistent with our assumptions. Therefore, our method is only suitable for mining benign TLS channels and cannot be used to identify malicious TLS channels.

Finally, it is worth mentioning that the benign channels can be used to form IP whitelist rules. The preprocessing method we proposed can also be used for mining whitelist IP addresses.

7. Conclusions

Because of the increasing TLS traffic, importing them into the NIDS indiscriminately will undoubtedly result in substantial processing pressure. Hence, it is a consensus to preprocess network traffic before completing detection. However, current studies seldom evaluate the impact of the results brought about by the preprocessing methods. Moreover, the classification and clustering models based on the single flow-based features are not significantly reliable in the preprocessing of TLS traffic. This study proposed a semiunsupervised model for quickly mining benign TLS channels to cope with such problems. We analyzed the differences between benign applications and malware on the communication behaviors of TLS channels in detail and proposed a set of new features. By adopting a spectral clustering algorithm and a wrapper method, a set of relevant features were selected, and a preprocessing model was established by applying a semiunsupervised algorithm. Through a set of experiments, the DBSCAN algorithm was selected from three other clustering algorithms to build a preprocessing model. Additionally, by comparing our method with two other methods, our experiments demonstrate that it not only performs better in terms of processing efficiency but is also significantly reliable for mined benign TLS channels.

The preprocessing model based on TLS channel features proved that it can mine the benign TLS channels used in this study. Indeed, TLS channel features have the potential to be further mined, and subsequently used to recognize malicious channels based on supervised models. For high camouflaged TLS flows produced by malware, it is difficult to detect them only based on a single flow, but we can observe and evaluate their behaviors, such as data-stealing behavior, from the perspective of their communication channels.

Therefore, in future studies, two main points need to be further explored for recognizing malicious TLS channels. One is to explore new TLS channel features that are solely effective in detecting malware traffic; the other is to ascertain whether the performance of classifiers can be improved by combining TLS channel-based features with traditional single flow-based features.

Data Availability

The malware traffic data used to support the findings of this study have been deposited in the Stratosphere IPS project repository (Available: https://www.stratosphereips.org/datasets-malware).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Defense Innovation Special Zone Program of Science and Technology (Grant no. JG2019055), in part by the Frontier Science and Technology Innovation Projects of National Key Research and Development Program (Grant no. 2019QY1405), and in part by the Key Research and Development Program of Sichuan Province (Grant no. 2020YFG0076).

References

[1] T. Dierks and C. Allen, *The TLS Protocol Version 1.0*, pp. 1–75, Internet Engineering Task Force (IETF), Fremont, CA, USA, 1999.

[2] Alienvault, *Alienvault, Inc.*,Alienvault, San Mateo, CA, USA, 2020, https://otx.alienvault.com.

[3] IBM X-Force Exchange, *IBM Security*, IBM X-Force Exchange, Atlanta, GA, USA, 2020, https://exchange.xforce.ibmcloud.com.

[4] Recorded Future, *Recorded Future, Inc*, Recorded Future, Somerville, MA, USA, 2020, https://support.recordedfuture.com.

[5] S. Zhao, S. Chen, Y. Sun, Z. Cai, and J. Su, "Identifying known and unknown mobile application traffic using a multilevel classifier," *Security and Communication Networks*, vol. 2019, Article ID 9595081, 11 pages, 2019.

[6] Y. C. Chen, Y. J. Li, A. Tseng, and T. Lin, "Deep learning for malicious flow detection," in *Proceedings of the 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, IEEE, Montreal, QC, Canada, pp. 1–7, October 2017.

[7] F. A. Khan, A. Gumaei, A. Derhab, and A. Hussain, "TSDL: a two-stage deep learning model for efficient network intrusion detection," *IEEE Access*, vol. 7, pp. 30373–30385, 2019.

[8] L. Su, Y. Yao, N. Li, J. Liu, Z. Lu, and B. Liu, "Hierarchical clustering based network traffic data reduction for improving suspicious flow detection," in *Proceedings of the 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)*, pp. 744–753, IEEE, New York, NY, USA, August 2018.

[9] J. Zhang, X. Chen, Y. Xiang, W. Zhou, and J. Wu, "Robust network traffic classification," *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1257–1270, 2015.

[10] S. Zhao, Y. Zhang, and P. Chang, "Network traffic classification using tri-training based on statistical flow characteristics," in *Proceedings of the 2017 IEEE Trustcom/BigDataSE/ICESS*, pp. 323–330, IEEE, Sydney, NSW, Australia, August 2017.

[11] L. Sacramento, I. Medeiros, J. Bota, and M. Correia, "Flowhacker: detecting unknown network attacks in big traffic data using network flows," in *Proceedings of the 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)*, pp. 567–572, IEEE, New York, NY, USA, August 2018.

[12] T. Glennan, C. Leckie, and S. M. Erfan, "Improved classification of known and unknown network traffic flows using semi-supervised machine learning," in *Proceedings of the 21st Australasian Conference on Information Security and Privacy*, Springer, Melbourne, VIC, Australia, pp. 493–501, June 2016.

[13] B. Anderson, S. Paul, and D. McGrew, "Deciphering malware's use of TLS (without decryption)," *Journal of Computer Virology and Hacking Techniques*, vol. 14, no. 3, pp. 195–211, 2018.

[14] Y. Yang, C. Kang, G. Gou, Z. Li, and G. Xiong, "TLS/SSL encrypted traffic classification with autoencoder and convolutional neural network," in *Proceedings of the 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pp. 362–369, IEEE, Exeter, United Kingdom, June 2018.

[15] B. A. AlAhmadi and I. Martinovic, "Malclassifier: Malware family classification using network flow sequence behaviour," in *Proceedings of the 2018 APWG Symposium on Electronic Crime Research (eCrime)*, pp. 1–13, IEEE, San Diego, CA, USA, May 2018.

[16] A. Gezer, G. Warner, C. Wilson, and P. Shrestha, "A flow-based approach for trickbot banking trojan detection," *Computers & Security*, vol. 84, pp. 179–192, 2019.

[17] M. Korczyński and A. Duda, "Markov chain fingerprinting to classify encrypted traffic," in *Proceedings of the IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pp. 781–789, IEEE, Toronto, ON, Canada, April 2014.

[18] X. Zeng, X. Chen, G. Shao et al., "Flow context and host behavior based shadowsocks's traffic identification," *IEEE Access*, vol. 7, pp. 41017–41032, 2019.

[19] R. Patil, H. Dudeja, and C. Modi, "Designing an efficient security framework for detecting intrusions in virtual network of cloud computing," *Computers & Security*, vol. 85, pp. 402–422, 2019.

[20] M. Baldi, A. Baldini, N. Cascarano, and F. Risso, "Service-based traffic classification: principles and validation," in *2009 IEEE Sarnoff Symposium*, pp. 1–6, IEEE, Princeton, NJ, USA, April 2009.

[21] J. Ma, K. Levchenko, C. Kreibich, S. Savage, and G. M. Voelker, "Unexpected means of protocol inference," in *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, pp. 313–326, Rio de Janeiro, Brazil, October 2006.

[22] Z. Zhao and H. Liu, "Spectral feature selection for supervised and unsupervised learning," in *Proceedings of the 24th International Conference on Machine Learning*, pp. 1151–1157, Corvallis, OR, USA, June 2007.

[23] F. Chung, "Spectral graph theory," in *Proceedings of the of CBMS Regional Conference Series in Mathematics*, AMS, Providence, RI, USA, December 1997.

[24] scu_igroup: Streamdump, 2020, https://github.com/scu-igroup/StreamDump.

[25] Netflow, 2020, https://netflow.us/.

[26] Moloch, 2020, https://molo.ch/.

[27] S. Lab: Malware Capture Facility Project, 2020, https://www.stratosphereips.org/datasets-malware.

[28] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 14, pp. 281–297, Berkeley, CA, USA, 1967.

[29] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Kdd-96*, vol. 96, no. 34, pp. 226–231, Portland, OR, USA, August 1996.

[30] D. A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," *Speech Communication*, vol. 17, no. 1-2, pp. 91–108, 1995.

[31] R. R. Sokal and C. D. Michener, "A statistical method for evaluating systematic relationships," *The University of Kansas Science Bulletin*, vol. 38, pp. 1409–1438, 1958.