Optimizing Feature Selection for Efficient Encrypted Traffic Classification: A Systematic Approach

Meng Shen, Yiting Liu, Liehuang Zhu, Ke Xu, Xiaojiang Du, and Nadra Guizani

Abstract

Traffic classification is a technology for classifying and identifying sensitive information from cluttered traffic. With the increasing use of encryption and other evasion technologies, traditional content-based network traffic classification becomes impossible, and traffic classification is increasingly related to security and privacy. Many studies have been conducted to investigate traffic classification in various scenarios. A major challenge to existing schemes is extending traffic classification technology to a broader space. In other words, most traffic classification work is not universal and can only show great performance on specific datasets. In this article, we present a systematic approach to optimizing feature selection for encrypted traffic classification. We summarize the optional encrypted traffic features and analyze the approaches of feature selection in detail for different datasets. The experimental result demonstrates that our scheme is more accurate and universal than other state-of-the-art approaches. More precisely, our mechanism provides a guideline for future research in the field of traffic classification.

Introduction

Encryption is an important means to protect privacy, which can protect our network traffic data from being peeped on, as well as prevent eavesdroppers from stealing our passwords or the usage habits of our applications. At present, more than half of the world's traffic has been encrypted by encryption protocols, such as SSL/TLS. Although this is good news for privacy-conscious users, network administrators face severe challenges. In the face of a large influx of traffic, without decryption technology, administrators will not be able to view the information contained in the traffic. This means that encryption is a double-edged sword. While protecting privacy, it also provides opportunities for eavesdroppers. Encrypted traffic can hide malicious software like other information, resulting in a series of worms (as well as Trojans and viruses). As a result, how to classify large-scale encrypted traffic and detect abnormal information in time becomes an urgent problem that needs to be solved.

These years, to address the classification problem of encrypted network traffic, many researchers have put forward a series of technologies and countermeasures. For instance, Feghhi and Leith [1] proposed an encrypted web traffic classification attack adopting only timing information. Shen *et al.* [2] used only packet length information to classify encrypted webpage traffic based on the bidirectional interaction between clients and servers. However, a common problem is that these schemes can only perform well on specific datasets. If the datasets are replaced or not processed according to the methods required by researchers, most schemes may lose effectiveness. Consequently, it is imperative to propose a universal encrypted traffic classification mechanism that can be effective for any traffic dataset.

To achieve this goal, we address two main challenges. The first challenge is to collect features of encrypted traffic that may be efficient in traffic classification. Those features based on packet contents are not suitable for encrypted traffic analysis, and are beyond our consideration. The second challenge is the selection and optimization of features. When we get a large number of potentially effective features of encrypted traffic, how to filter these features and optimize the combination of features according to specific scenarios is the most critical step to achieve universal encrypted traffic classification.

In this article, we present a systematic approach to optimizing feature selection for efficient encrypted traffic classification. In the next section, we give a macro explanation of where encryption traffic classification work is carried out, and elaborate the state-of-the-art encrypted traffic classification research. Then we point out the feature set of encrypted traffic. Following that, we describe how to evaluate and combine features for different datasets. Next, we show our experiment performances. Finally, we propose future research directions and conclude our article.

Overview of Encrypted Traffic Classification

Traditional traffic classification techniques are usually based on the analysis of packet contents, such as port-based approaches and payload-based approaches. However, with the enhancement of people's security consciousness, the usage of encryption protocols is growing rapidly, and the packet payloads are encrypted, thus leading to the traditional techniques being unusable. There-

This work is partially supported National Natural Science Foundation of China under Grants 61972039 and 61872041, Beijing Natural Science Foundation under Grant 4192050, China National Funds for Distinguished Young Scientists under Grant 61825204, Zhejiang Lab Open Fund No. 2020AA3AB04, the Key Lab of Information Network Security (Ministry of Public Security), Beijing Outstanding Young Scientist Program with No. BJJW-ZYJH01201910003011, Science and Technology Planning Project of Guangdong Province under Grants LZC0023 and LZC0024, PCL Future Greater-Bay Area Network Facilities for Large-scale Experiments and Applications (LZC0019), and Technology Innovation Program of Beijing Institute of Technology (3052019023).

Digital Object Identifier:
10.1109/MNET.011.1900366

Meng Shen is with Beijing Institute of Technology & Key Lab of Information Network Security & Peng Cheng Laboratory, Yiting Liu and Liehuang Zhu (corresponding author) are with Beijing Institute of Technology; Ke Xu is with Tsinghua University & BNRist & Peng Cheng Laboratory; Xiaojiang Du is with Temple University; Nadra Guizani is with Gonzaga University.

0890-8044/20/$25.00 © 2020 IEEE

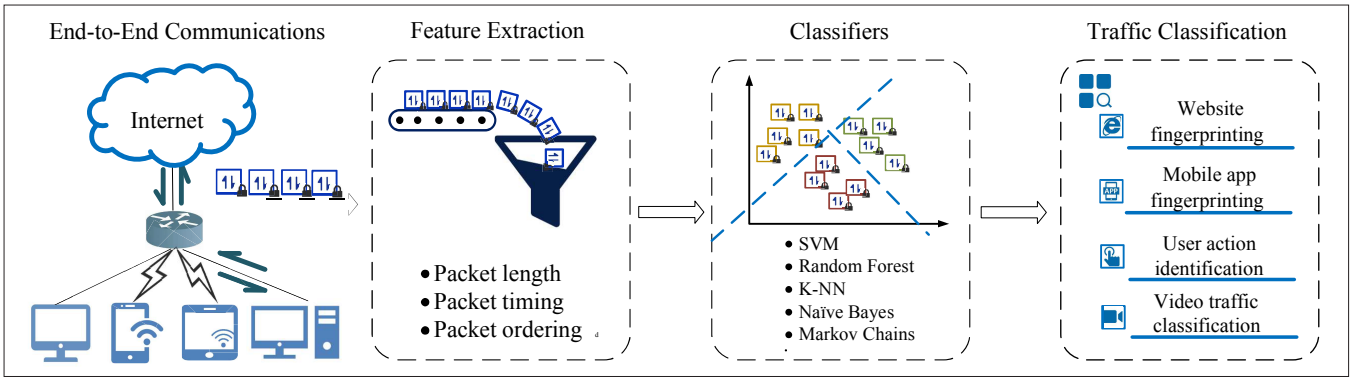

FIGURE 1. Overall framework of encrypted traffic classification.

Design goal	Features	Classifiers	Traffic type	Ref.
Website fingerprinting	Packet timing information	*k*-NN and Naïve Bayes	VPN	[1]
	Cumulative packet length	SVM	Tor	[6]
	Distance between packet sequences	*k*-NN	Tor	[7]
Mobile app fingerprinting	Statistical features of packet length	Random Forest	SSL/TLS	[8]
	Application attribute bigrams	Second-order Markov Chains	SSL/TLS	[9]
User action identification	Complete flow time series	Random Forest	SSL/TLS	[10]
Video traffic classification	Total number of bits in a peak	SVM and Nearest Neighbor	SSL/TLS	[11]
	Statistical features of downlink packet length and time	*k*-NN	SSL/TLS	[12]

TABLE 1. State of the art of encrypted traffic classification.

fore, how to classify encrypted traffic effectively has become a research hotspot in the field of network security [3–5]. A lot of research has been conducted in various scenarios. For instance, some researchers identify website fingerprints with traffic from Tor or other encrypted networks, others classify encrypted traffic generated by mobile applications, and some identify encrypted video traffic such as video titles and video types. Figure 1 shows the overall framework for encrypted traffic classification. Several common types of encrypted traffic classification studies are as follows.

Website fingerprinting. Feghhi and Leith [1] proposed a website traffic classification approach using only packet timing information on the link. They applied dynamic time warping (DTW) to classify traffic traces with time sequences. Their approach does not need to achieve the start or end of web fetches. Panchenko *et al.* [6] adopted cumulative packet length sequences as website fingerprints and extracted a fixed number of discriminative features from traffic traces with different lengths. Wang *et al.* [7] calculated the distance between packet sequences to realize website fingerprinting.

Mobile application fingerprinting. Taylor *et al.* [8] proposed an Android app fingerprinting scheme applying packet length statistics features, which is called AppScanner. It is capable of automatic fingerprinting and real-time identification. Shen *et al.* [9] utilized second-order Markov chains to classify encrypted traffic generated by different mobile apps. Their program is based on the features of application attribute bigrams containing certificate packet length and the first application data size in SSL/TLS traffic. Mauro *et al.* [10] achieved the identification of user actions by analyzing an Android encrypted network. They adopted the DTW algorithm to calculate the sequence of data packets and extract features. After that, random forest was used to be their classifier. Finally, the classification of user actions is reached by clustering.

Video traffic classification. Ran *et al.* [11] proposed a scheme to classify video stream title by analyzing encrypted traffic. Support vector machine (SVM) and Nearest Neighbor were used as their classifiers. They can also identify the video titles not existing in the training set as unknown. Dong *et al.* [12] proposed a fine-grained classification method for video traffic based on the *k*-Nearest Neighbor algorithm. All these articles are summarized in Table 1.

Although many researchers have proposed a variety of encrypted traffic classification schemes, these schemes are not universal. In other words, all these schemes only have good effects on the specific dataset proposed by the researchers. When replacing the dataset, many approaches lose their effectiveness. Motivated by this situation, we show the candidate features that can be leveraged to discriminate encrypted traffic and propose a systematic approach to optimize feature selection for efficient encrypted traffic classification.

Note that while some existing traffic analysis schemes [8] also considered evaluating feature contribution, there are several obvious differences between the proposed method in this article and the existing methods. First, we give a systematic method to summarize all the categories

When replacing the dataset, many approaches lose their effectiveness. Motivated by this situation, we show the candidate features that can be leveraged to discriminate encrypted traffic and propose a systematic approach to optimize feature selection for efficient encrypted traffic classification.

and sub-classes of candidate features that can be extracted from encrypted traffic. While existing studies focus merely on concrete and limited numbers of features. Second, by considering the association relationships of features and their influence on the results, we make flexible feature selection according to the needs of different scenarios (objective and constraint). This framework does not exist in the existing approaches since existing papers usually have already set a fixed scenario in advance. Overall, our scheme is more universal and can effectively analyze any traffic datasets, thus providing a guide for future traffic classification work.

Feature Set of Encrypted Traffic

As illustrated in Table 1, a rich number of features of encrypted traffic can be used to train efficient classifiers. However, existing studies focus merely on concrete features that are proven to be discriminative for the datasets they use, leaving the whole space of candidate features unexploited. Thus, we summarize the categories and sub-classes of candidate features that can be extracted from encrypted traffic.

For the sake of analysis, it is necessary to sort out the messy encrypted traffic into multiple flows based on the five-tuple: source/destination IP addresses, source/destination port numbers, and protocol (TCP/UDP). For each flow, three types of packet series are considered: uplink packets only, downlink packets only and complete packets (i.e., both uplink and downlink packets). Furthermore, we define an uplink packet burst as a group of uplink packets in which there are no two adjacent downlink packets.

Features Based on Packet Length

As packet length is an affiliated feature of network packets, packet length information becomes a kind of commonly used feature and has demonstrated its effectiveness in encrypted traffic analysis.

Packet length sequence. For each flow, the packet length sequence of the first *n* packets can be used as an important feature. For instance, the first *n* packets may vary greatly in terms of packet length for individual websites due to the differences in their contents and protocol parameters such as those in the SSL/TLS handshake process.

Unique packet length. The existence of unique packet length in a flow is a significant feature, which is also mentioned in the literature [7]. It represents that in a traffic dataset, some packet lengths only appear in one type of traffic, but not in others. We are able to utilize the unique packet length to distinguish different types of traffic. However, it loses effect if there is packet padding, as on Tor.

Packet length box. The length of the packets in a flow is usually scattered over the packet length interval. In order to get statistical characteristics of the packet length in a flow, we can aggregate the packet length into a fixed number of boxes. Assume that each box represents a certain range of packet length; it is straightforward to obtain the number of packet lengths that fall into each range.

Sequences of cumulative length. When considering the direction of flows, we can first set the length of uplink packets as negative and the length of downlink packets as positive. Then we accumulate the packet length forward to obtain a sequence of the cumulative lengths of the first *n* packets. When considering the flows as directionless, the length of both uplink and downlink packets can be set positive. The sequence of the cumulative lengths of the first *n* packets in a flow is captured as a discriminative feature [6].

Statistical features. For each flow, we are able to calculate statistical values of packet lengths, such as minimum, maximum, mean, median absolute deviation, standard deviation, variance, skew, kurtosis, percentiles (from 10 to 90 percent), length summation, and number of packets [8]. We totally consider these 57 statistical values for the three packet series (uplink packets, downlink packets, and complete packets).

Features Based on Packet Ordering

In certain cases, the packet lengths are fixed or similar among different encrypted traffic flows, thereby making those features based on packet length information less effective. For instance, Tor uses cell padding techniques to send data in fixed-size (512-byte) cells. To deal with these situations, packet ordering information can be taken as important features.

Packet counts. Several types of packet counts can be considered. We can count the number of uplink and downlink packets for every *n* packets. Furthermore, the total number of packets before each uplink packet is also a useful feature. We could extract the feature that indicates the number of downlink packets between every two uplink packets.

Burst counts. Burst is a significant feature as demonstrated in the literature [13]. Here, we focus on uplink packet burst as downlink packets are vulnerable to network delay. We regard a burst as a whole and count the number of bursts as well as the maximum and mean of burst length in each flow.

Features Based on Packet Timing

Several researchers also classify encrypted traffic utilizing timestamps of packets. The features extracted from packet timing information are as follows.

Inter-packet delays. For each packet series, inter-packet delay is defined as the difference sequence of timestamps of two adjacent packets.

Combination of packet counts and timing. For each packet series, we calculate the number of packets in a certain time interval. This indicates the time period in which data packet transmission is concentrated.

Theoretically speaking, we treat all these features equally. However, timing characteristics are usually not useful, because the distribution of timings in different websites is similar. In addition, timestamp information of packets is greatly affected by network fluctuations. As a result, in most cases, we prioritize other features.

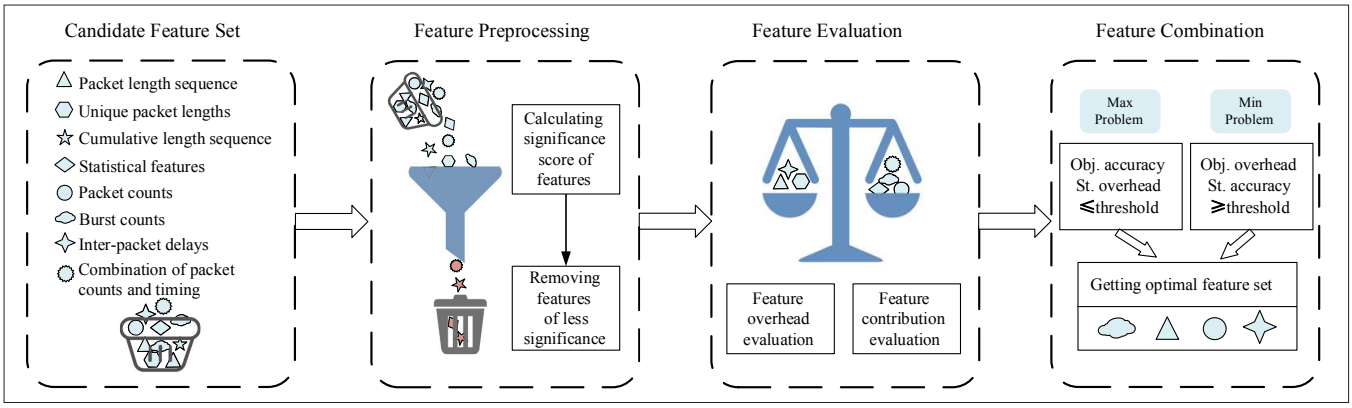

FIGURE 2. Process of feature selection.

A Systematic Approach for Optimizing Feature Selection

This section describes a systematic approach for optimizing feature selection. As shown in Fig. 2, the feature selection framework consists of three components: feature preprocessing, feature evaluation, and feature combination.

The purpose of feature preprocessing is to preliminarily remove traffic characteristics without discrimination. In these three steps, feature evaluation is the most significant one. The evaluation of a feature is usually considered from two aspects: feature contribution and feature overhead. For different issues, the focus of feature evaluation may be different. More details are presented later.

After data preprocessing and feature evaluation, we need to select and combine meaningful features and input these features into classifiers (e.g., machine learning models) so as to achieve traffic classification (e.g., website fingerprinting). The main purpose of feature selection is to reduce the number and dimension of features, thus making the model more generalized and reducing over-fitting.

Feature Preprocessing

In order to find out the meaningful features for traffic classification more efficiently, we need to pre-process the traffic features with significance testing and remove those features of less significance. It can be achieved by calculating the variance of features.

Assume that the feature values are only 0 and 1, and the value is 1 in 95 percent of all input traffic. Then it can be considered that the contribution of this feature is small. If 100 percent is 1, the feature is meaningless. Generally speaking, if the feature values of network traffic are discrete variables, we can calculate the variance of features directly and remove those features with low variance. However, if the features are continuous variables, they need to be discretized before they can be used.

In the actual process of traffic classification, there are generally no more than 95 percent of characteristics that are the same values. Accordingly, although this method is simple, it cannot be classified very accurately. As a result, we take it as the preprocessing of feature selection. We first remove those features with small change in values, and then select the appropriate features by the feature evaluation approaches mentioned below so as to better carry out traffic classification.

Feature Evaluation

Through feature preprocessing, we initially remove some traffic features with low discrimination. Next, we need to further evaluate features from two aspects: feature contribution and feature overhead.

Feature Contribution Evaluation: Feature contribution is an important index to measure whether a feature plays a role in traffic classification. The main approaches for obtaining feature contribution are as follows.

Chi-square test. The classical chi-square test is to test the correlation between qualitative independent variables and qualitative dependent variables. In traffic classification, it is mainly used for evaluating feature contribution for binary classification problems; for example, distinguishing whether the encrypted traffic is generated by an anonymous communication system (e.g. Tor) or not. The main idea is that calculating chi-square values of the features of each dimension and ranking them. The chi-square values represent the feature importance.

Term frequency-inverse document frequency (TF-IDF). TF-IDF is a common weighting technique for information retrieval and data mining, where TF means term frequency and IDF means inverse document frequency. It is mainly adopted to assess the importance of a word to a document set or one of the documents in a corpus. We are able to utilize this method to evaluate feature importance in traffic classification. First of all, we expect to code the TCP packets. Packet encoding could be combined with the direction, length, and flag bit of the packet. For instance, *U_54_SY N* represents an uplink SYN packet with a length of 54. Then we are able to calculate the value of TF and IDF of encoded packets. After that, we obtain the packets with TF-IDF frequency. When it comes to feature selection, we are able to remove those packets whose TF-IDF frequency is less than a certain threshold, and utilize the remaining packets for classification. Certainly, appropriate transformations are expected to be made for different problems.

Model-based ranking. A mainstream feature contribution evaluation method is based on a machine learning model. Some machine learning methods have their own scoring mechanism, or

Features	Time complexity	Space complexity
Packet length sequence	$O(n)$	$O(n)$
Unique packet lengths	$O(n)$	$O(1)$
Packet length box	$O(n)$	$O(1)$
Cumulative length sequence	$O(n)$	$O(n)$
Statistical features	$O(n^2)$	$O(n)$
Packet counts	$O(n^2)$	$O(n)$
Burst counts	$O(n^2)$	$O(n)$

TABLE 2. Feature overhead evaluation.

can easily be applied to feature evaluation tasks, including regression model, SVM, decision tree, random forest, and so on. Taking random forest as an example, we can measure the importance of features by the average impurity decay of all decision trees in the forest without considering whether the data is linear separable. More conveniently, random forests implemented in scikit-learns have already collected information about the feature contribution for us. After fitting RandomForestClassifier, we can get these contents through feature_importances. Besides, other machine learning models we have mentioned are the same. As a result, this method is convenient and useful, and it has been widely used in feature contribution evaluation.

Feature Overhead Evaluation: Feature overhead mainly contains two aspects: the time complexity and space complexity of the feature extraction algorithm.

Time complexity. The time complexity of feature extraction reflects the increased magnitude of feature extraction time with the increase of traffic scale. To a large extent, it can reflect the quality of a feature extraction algorithm. To calculate time complexity, we need to find out the basic statements in the feature extraction algorithm. The most frequently executed statements in the algorithm are the basic statements. Then we expect to calculate the order of execution times of basic statements, which means that all the coefficients of the lowest power and the highest power can be ignored as long as the highest power in the function of the number of executions of the basic statement is correct. This simplifies algorithm analysis and makes us focus on the most important point: growth rate. After that, O is used to represent the time performance of the algorithm, and n represents the scale of the problem. We need to put the order of magnitude n of the number of executions of the basic statement into O.

Space complexity. Similar to the discussion of time complexity, the space complexity of the feature extraction algorithm is defined as the storage space consumed by the algorithm. The storage space occupied by a feature extraction algorithm in computer memory includes the storage space occupied by the algorithm itself, the input and output data, and the temporary storage space occupied by the algorithm in operation. The storage space occupied by data is determined by the traffic scale, and it does not change with the difference of the algorithm. The storage space occupied by the storage algorithm itself is proportional to the length of the algorithm writing. The temporary storage space occupied by the algorithm varies with the algorithm.

Here, we summarize the time and space complexity of each feature mentioned earlier, and exhibit it in Table 2.

Feature Combination

After feature preprocessing and feature evaluation, we get the features with three attributes: contribution degree, time complexity, and space complexity. The last step is to select features and combine them into an optimal feature set for the corresponding dataset. The main aim of feature combination is to select a more representative subset from the original feature set.

In different scenarios of traffic classification applications, the features we need are different. Some traffic classification scenarios require rapid response; for instance, network management needs to quickly identify traffic generated by users using illegal anonymous software and block traffic in time. At this time, it can relax the requirement of traffic identification accuracy (e.g., not less than 70 percent), but it needs to ensure the identification speed. Conversely, for some offline traffic classification work, what we are more concerned about is the accuracy of classification, while we have greater tolerance for the storage and time consumption of classification.

Generally speaking, feature selection and combination are required to consider the following two aspects:

- Minimizing overhead while guaranteeing certain accuracy
- Improving accuracy when overhead does not exceed a certain threshold

For those traffic classification tasks that require low time and space complexity, we first need to clarify the time and space complexity of each feature, and give priority to the features with low complexity. Then we need to evaluate feature contribution to further select valuable features from them. For those traffic classification tasks that require high accuracy, we need to extract traffic features and rank them by feature contribution. After that, we expect to pick up those traffic features with high contribution.

When we sort the features according to the demand, the simplest way to select the features is to add them into the feature set in turn and observe the accuracy of traffic classification until the highest accuracy is achieved. Also, we could first try those features that are ranked in the top 50 percent and calculate the classification accuracy. Then adding or subtracting features in feature set until the highest accuracy is achieved.

Experiment Evaluation

In this section, universal traffic classification experiments are performed on one self-collected dataset and one representative dataset, provided by Panchenko *et al.* [6], to evaluate the performance of our proposed method UTA. The schemes called Appscanner [8] and CUMUL [6] are used for comparison.

Datasets

This section describes the datasets used in our experiments. An independent and representative dataset is of vital importance for significant exper-

iment results. Here, we consider adopting one typical dataset provided by Panchenko *et al.* [6] called Panchenko16 and another dataset collected by ourselves called Website100.

The reason why we choose Panchenko16 as our experiment dataset is that it consists of the most extensive data in all public datasets. Instead of just collecting traffic from index pages of a limited number of websites, Panchenko16 contains more webpages of a site besides the index page and collects a large amount of website traffic, which is able to reflect the representative samples of the Internet. Cui *et al.* [14] also leveraged Panchenko16 to confirm the effectiveness of their approach. Specifically, it contains 1125 retrievable webpages' traffic from 712 different websites, and each webpage has 40 instances. This dataset includes information about the direction and size of each packet.

Most prior public datasets including Panchenko16 lack part of the traffic information (e.g., port number or arrival time), which makes us unable to extract all the candidate features mentioned earlier. As a result, we collect our own dataset named Website100, which keeps the complete traffic information, so that we can extract all candidate features and further conduct feature selection. It is composed of the traffic from the top 100 websites in China. For each website, we downloaded 200 instances. We divided the traffic into flows based on a five-tuple representation: srcIP, dstIP, srcPort, dstPort, protocol (TCP/UDP), where srcIP represents the client IP address, dstIP represents the server IP address, srcPort represents the client port number, dstPort represents the server port number, and protocol represents the communication protocol established between the client and the server. Furthermore, we remove TCP retransmission packets because retransmissions are mostly caused by network conditions.

In the following experiments, we apply UTA to these two datasets to demonstrate the universality of the proposed method.

Experimental Results

In this section, we first briefly introduce two typical website fingerprinting schemes. Then we put forward our solution, UTA, and take those two state-of-the-art methods for comparison to validate the universality of UTA.

Comparison Scheme: Here, we take two state-of-the-art website fingerprinting approaches into consideration as our comparison schemes:

- *CUMUL* is proposed by Andriy *et al.* [6]. This scheme has the highest website fingerprinting accuracy known to date. They adopted cumulative packet lengths to represent the traffic traces and leverage SVM as their classifier.
- *Appscanner* is another typical traffic classification scheme proposed by Taylor *et al.* [8]. It is a robust method that was designed to identify smartphone apps using encrypted network traffic analysis. This method is also widely used in website fingerprinting. It utilizes 54 statistics features of uplink flow, downlink flow, and complete flow as their feature set and classify traffic based on random forest.

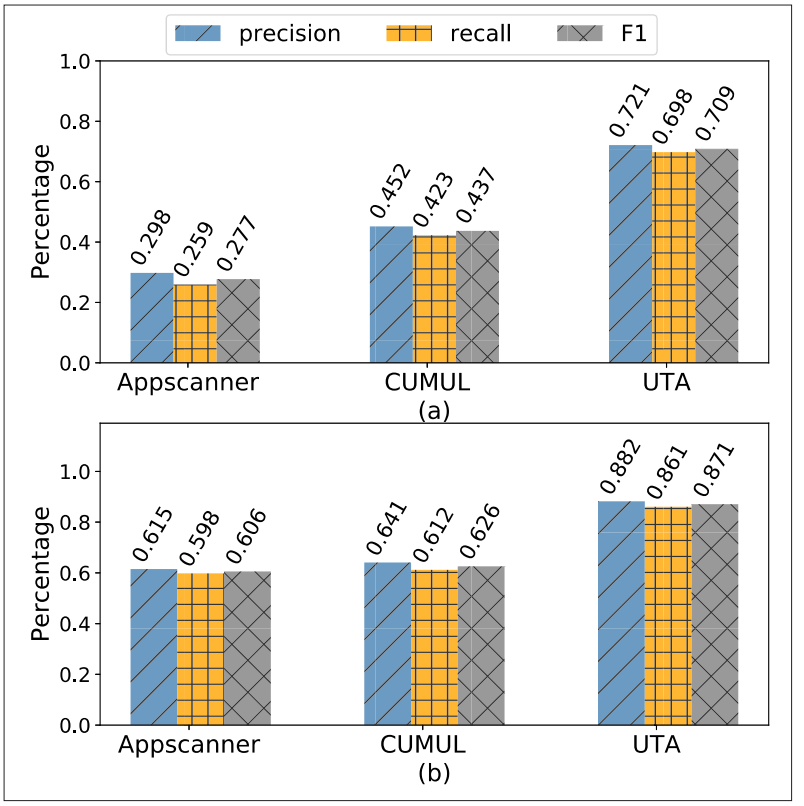

FIGURE 3. Website fingerprinting results with datasets of a) Panchenko16; b) Website100.

Performance: According to the idea of UTA, we start with extracting all possible useful features from each dataset. The dataset is a collection of website traffic that has been grouped into flows. The features we extracted were shown earlier. We expect to pre-process the feature sets to remove meaningless features. We calculate the variance of each feature and filter out the features with low variance. After that, we count the contribution of the remaining features. In our experiments, we utilize the random forest algorithm implemented in scikit-learn to calculate the importance of features. We chose the top 45 percent contribution features. Table 3 shows the features we pick out for the two datasets: Panchenko16 and Website100. After combining these significant features, we apply random forest as our classifier and use 10-fold cross-validation to get website fingerprinting results.

We consider three metrics — precision, recall, and F1 — to measure the classification results. Figure 3 depicts the performance of these three methods (UTA, Appscanner, and CUMUL) on the two datasets. We can see that no matter which metrics, the performance of UTA is significantly higher than that of the other two methods on both datasets. This result strongly proves that our feature combination scheme is effective and that UTA is more universal than other state-of-the-art schemes.

Challenges and Future Directions

We present a feature selection approach for encrypted traffic classification, and the experimental results show that our method is effective for different datasets. However, several issues still need to be explored in order to further improve

Dataset	Feature set	
	Features	# features
Panchenko16 (277)	Statistical features	57
	The first 40 cumulative packet lengths with direction	40
	Number of packets before each uplink packet	180
Website100 (157)	Statistical features	57
	The first 40 cumulative packet lengths with direction	40
	The first 30 cumulative packet lengths without direction	30
	The first 10 of uplink, downlink, and complete packet length sequences, respectively	30

TABLE 3. Significant features selected by UTA for two datasets.

our approach. The challenges and future directions are as follows.

Relationship between features and datasets. In different traffic classification problems, we usually need to extract a large number of possible features and repeatedly attempt to find those features with significant discrimination for specific datasets. This process is time-consuming and labor-intensive with little effect. Therefore, one of the challenges at present is to find the relationship between datasets and features. For a given dataset, we are able to find optimal features more quickly.

Based on our experience, we preliminarily summarize the possible relationship between features and datasets. In terms of the bi-classification problem, for instance, we need to distinguish whether the traffic in the dataset is generated by visiting the website when opening the encryption service application or by visiting the website normally. The encryption service application usually has a unique information exchange protocol, and this protocol is ordinarily reflected in the packet length information in the process of packet transmission. Hence, based on experience, we generally analyze the packet length sequence for such problems.

For the multi-classification problems, the situation is usually complex, and we need to analyze it according to the datasets. Still using website fingerprinting as an example, we need to determine which websites a user is visiting over an encrypted connection. If the traffic in a dataset is mostly composed of short flows (e.g., no more than 100 packets), we could mainly consider the special packet length, the state transition characteristics of the packet length, and the statistical characteristics of the packet length. For those datasets that are mostly composed of long streams, we give priority to consider cumulative packet length sequence and statistical characteristics. These are just preliminary summaries. We need to do further research to find the exact relationship between features and datasets.

Real-time detection. The traffic we analyze now is pure traffic after de-noising in the laboratory, but in the real world, traffic is complex and ever-changing. Furthermore, network administrators need to detect abnormal traffic in real time and block it in time. As a result, we expect to analyze the real world traffic directly and achieve real-time detection in a future experiment, thus applying the universal classification model to a wider space.

To realize real-time detection, the most crucial step is to find traffic characteristics that can support it. Among the features summarized in this article, only the cumulative packet length feature can barely support real-time detection, and other features need to wait for all flow loads to be completed before they can be extracted. Therefore, we need to carry out more research to solve this problem.

Features compression. Although we have extracted meaningful features from a large feature set, these features still have the problem of information overlap or little contribution to some extent. Accordingly, a major challenge is feature compression. We look forward to finding effective algorithms to minimize feature dimension under the condition of guaranteeing accuracy, thus reducing the storage and computation overheads of traffic classification.

Principal component analysis (PCA) is a statistical method for dimensionality reduction. It also has an effect on traffic feature compression [15]. However, this algorithm is sensitive to the scaling of the data, and there is no consensus as to how to best scale the data to obtain optimal results. As a result, using PCA for feature compression often results in loss of classification accuracy. It is of vital importance to explore a compression algorithm that can minimize the loss of classification accuracy or even make it non-destructive, thus making the research on traffic classification more efficient and lightweight.

Conclusion

In this article, we propose a systematic approach to optimize feature selection for efficient encrypted traffic classification. We start by introducing an overview of encrypted traffic classification studies. Following that, we summarize the feature set of encrypted traffic and show how to select these features for different datasets, including feature preprocessing, feature evaluation, and feature combination. Then we exhibit the experiment performance and prove that our feature selection scheme has the quality of universality in encrypted traffic classification. We also discuss the challenges and future directions in the traffic classification field.

References

[1] S. Feghhi and D. J. Leith, "A Web Traffic Analysis Attack Using Only Timing Information," *IEEE Trans. Info. Forensics and Security*, vol. 11, no. 8, 2016, pp. 1747–59.

[2] M. Shen *et al.*, "Webpage Fingerprinting Using Only Packet Length Information," *IEEE ICC 2019*, 2019, pp. 1–6.

[3] X. Du *et al.*, "A Routing-Driven Key Management Scheme for Heterogeneous Sensor Networks," *Proc. IEEE ICC 2007*, Glasgow, Scotland, 2007, pp. 24–28.

[4] L. Xiao *et al.*, "Cloud-Based Malware Detection Game for Mobile Devices with Offloading," *IEEE Trans. Mobile Computing*, vol. 16, no. 10, Oct. 2017, pp. 2742–50.

[5] M. Shen *et al.*, "Secure SVM Training Over Vertically-Partitioned Datasets Using Consortium Blockchain for Vehicular Social Networks," *IEEE Trans. Vehic. Tech.*, 2019, pp. 1–1.

[6] A. Panchenko *et al.*, "Website Fingerprinting at Internet Scale," *Network and Distributed System Security Symp.*, 2016, pp. 21–24.

[7] T. Wang *et al.*, "Effective Attacks and Provable Defenses for Website Fingerprinting," *Usenix Conf. Security Symp.*, 2014, pp. 143–57.

[8] V. F. Taylor *et al.*, "Robust Smartphone App Identification Via Encrypted Network Traffic Analysis," *IEEE Trans. Info. Forensics and Security*, vol. 13, no. 1, 2018, pp. 63–78.

[9] M. Shen *et al.*, "Classification of Encrypted Traffic with Second-Order Markov Chains and Application Attribute Bigrams," *IEEE Trans. Info. Forensics and Security*, vol. 12, no. 8, 2017, pp. 1830–43.
[10] M. Conti *et al.*, "Analyzing Android Encrypted Network Traffic to Identify User Actions," *IEEE Trans. Info. Forensics and Security*, vol. 11, no. 1, 2016, pp. 114–25.
[11] R. Dubin *et al.*, "I Know What You Saw Last Minute Encrypted Http Adaptive Video Streaming Title Classification," *IEEE Trans. Info. Forensics and Security*, vol. 12, no. 12, 2017, pp. 3039–49.
[12] Y. Dong, J. Zhao, and J. Jin, "Novel Feature Selection and Classification of Internet Video Traffic Based on a Hierarchical Scheme," *Computer Networks*, vol. 119, 2017, pp. 102–11.
[13] T. Stöber *et al.*, "Who Do You Sync You Are?: Smartphone Fingerprinting Via Application Behaviour," *Proc. 6th ACM Conf. Security and Privacy in Wireless and Mobile Networks*, 2013, pp. 7–12.
[14] W. Cui *et al.*, "Revisiting Assumptions for Website Fingerprinting Attacks," *Proc. 2019 ACM Asia Conf. Computer and Commun. Security*, ser. Asia CCS '19, 2019, pp. 328–339.
[15] M. Nasr, A. Houmansadr, and A. Mazumdar, "Compressive Traffic Analysis: A New Paradigm for Scalable Traffic Analysis," *Proc. 2017 ACM SIGSAC Conf. Computer and Communications Security*, 2017, pp. 2053–69.

Biographies

Meng Shen [M'14] received his B.Eng. degree from Shandong University, Jinan, China, in 2009 and his Ph.D. degree from Tsinghua University, Beijing, China, in 2014, both in computer science. Currently he serves at Beijing Institute of Technology, as an associate professor. His research interests include privacy protection for cloud and IoT, blockchain applications, and encrypted traffic classification. He received the Best Paper Runner-Up Award at IEEE IPCCC 2014.

Yiting Liu received her B.Eng. degree in computer science from Northwest A&F University, Shanxi, China, in 2017. Currently, she is a graduate student in the Department of Computer Science, Beijing Institute of Technology. Her research interests include anonymity networks and cyber security.

Liehuang Zhu [M'16] is a professor in the Department of Computer Science at Beijing Institute of Technology. He is selected into the Program for New Century Excellent Talents in University from the Ministry of Education, P.R. China. His research interests include the Internet of Things, cloud computing security, and Internet and mobile security.

Ke Xu [M'02, SM'09] received his Ph.D. from the Department of Computer Science and Technology of Tsinghua University, where he serves as a full professor. He serves as Associate Editor of the *IEEE Internet of Things Journal* and has guest edited several special issues in IEEE and Springer journals. His research interests include next generation Internet, P2P systems, the Internet of Things, network virtualization, and network economics. He is a member of ACM.

Xiaojiang Du [S'99, M'03, SM'09, F'19] is a tenured professor in the Department of Computer and Information Sciences at Temple University, Philadelphia, Pennsylvania. He received his B.S. and M.S. degree in electrical engineering from Tsinghua University in 1996 and 1998, respectively. He received his M.S. and Ph.D. degrees in electrical engineering from the University of Maryland College Park in 2002 and 2003, respectively. His research interests are wireless communications, wireless networks, security, and systems. He has authored over 300 journal and conference papers in these areas, as well as a book published by Springer. He has been awarded more than US$5 million in research grants from the U.S. National Science Foundation, Army Research Office, Air Force, NASA, the State of Pennsylvania, and Amazon. He won the best paper award at IEEE GLOBECOM 2014 and the best poster runner-up award at ACM MobiHoc 2014. He serves on the Editorial Boards of three international journals. He is a Life Member of ACM.

Nadra Guizani is a lecturer in computer science at Gonzaga University. She received her Ph.D degree in computer engineering from Purdue University in 2018. Her research interests include machine learning, mobile networking, large data analysis, and prediction techniques. She is an active member in the Eta Kappa Nu Honors Society, Women in Engineering Program, and Computing Research Association for Women.