LaFFT: Length-Aware FFT Based Fingerprinting for Encrypted Network Traffic Classification

Chang Liu[1,2], Zigang Cao[1,2], Zhen Li[1,2], Gang Xiong[1,2]
1.Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
2.School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
caozigang@iie.ac.cn

***Abstract*—Encrypted traffic classficiation has become an emergent and challenging task for network monitoring and management. Traditional classification methods for encrypted traffic rely on complex statistical characteristic construction and in-depth packet resolution, which produce huge loads. In this paper, we develop Length-aware FFT (LaFFT) fingerprinting to identify different encrypted application traffic with packet length sequences. We apply FFT to packet length sequences to generate the frequency domain vectors as LaFFT features. We verify the distinguishability of LaFFT fingerprinting by data analysis. Furthermore, the linear inseparability and the front superiority of LaFFT fingerprinting are demonstrated by comprehensive experiments. In the real-world dataset, the LaFFT fingerprinting with random forest classifier can achieve 96.8% TPR, 0.32% FPR and 0.959 FTF, which significantly outperform the state-of-the-art methods.**

I. Introduction

As encrypted protocols are widely used in network services to enhance communication security and user privacy, encrypted traffic in real network world shows an increasing trend. Meanwhile, along with the emergence of malicious network traffic generated by trojan horses, viruses and network attacks, network management and monitoring attract a lot of attentions. Thereinto, the key and urgent task is encrypted traffic analysis, and the first step is the encrypted traffic classification [1]. However, to classify encrypted traffic is still a formidable task because these traffic flows can not directly be parsed in depth after encryption. Moreover, for answering the need of online traffic monitoring program, the classification methods should have the ability to handle millions of traffic flows from different users at the same time, with low loads and high classification accuracy.

Traditional rule-based classification methods [2], [3] cannot handle encrypted traffic classification task, because of the limited available plain text information and the complexity of decrypting encrypted packets [4]. In recent years, machine learning are applied to settling these problems with human-designed features and the advanced classification algorithms. However, obtaining these features need to parse raw encrypted traffic packets in depth at first. For examples, [5] uses message type sequences as features, [6] parses SSL/TLS handshake packets, and [7] establishes hidden Markov models on packet-level. These methods consume much memory and increase the loads which are not fit for real-time network environment.

In this paper, we focus on the packet length sequence, which is an easiest attribute to get and no need to parse packets in depth. However, packet length sequences cannot be directly used as classification features due to two challenges:

- Variable packet numbers. The packet number of a flow fluctuate in a large range. However, most of classifiers require the input feature vectors with equal length.
- Packet relevance. One packet in a flow is related with context packets, and the order of packets is important to distinguish a flow. However, each dimension of the input feature vectors is non-ordered in classifiers.

To handle these challenges, we purpose the LaFFT fingerprinting as features to classify encrypted traffic flows. The LaFFT fingerprinting is the frequency domain features from Fast Fourier Transform (FFT) of length packet sequence. With FFT, the packet-related length sequences in time domain is transformed into independent frequency domain vectors with equal length, which can be directly fed into classifiers. And the whole information of variable length sequence can be kept by the frequency domain features. Moreover, FFT is quick, which is more suitable for the real-time network environment. By data analysis and comprehensive experiments, we verify that LaFFT fingerprinting is expressive for encrypted traffic classification, especially combined with nonlinear classifier, e.g. Random Forest (RF), and obtains results with 96.8% TPR, 0.32% FPR and 0.959 FTF.

Our contributions can be briefly summarized as follows:

- We purpose LaFFT fingerprinting, which are not only equidimensional independent features, but also keep the whole information of the original packet length sequence.
- We discovered the *distinguishability*, *front superiority* and *linear inseparability* of LaFFT fingerprinting for encrypted traffic classification by data analysis and comprehensive experiments.
- LaFFT fingerprinting can produce an excellent performance on the real-world dataset, outperforming the state-of-the-art methods.

The rest of this paper is organized as follows. We refine related work in Section II and give the dataset description in Section III. Section IV describes the proposed methodology in detail. Comprehensive analysis of experiments is shown in Section V. Finally, we conclude this paper in Section VI.

II. Related Work

Encrypted traffic classification brings huge challenges to network management and monitoring. Recently, researches on encrypted traffic classification are mainly based on human-designed feature [1], [6], [8]. S.M. Kim et al. used the publication information field in the certificate packet during the certificate exchanging process as the signatures [9]. A hybrid method with signature-based and flow statistical analysis pattern was proposed to classify SSL/TLS encrypted application traffic [10]. Shbair et al. proposed a robust HTTPS services classification technique [11], and some specialized features are constructed by specific header fields to feed into a multi-level classification framework based on machine learning algorithms. These methods all depend on complex feature construction, which is hard to be balanced with fast mass flow identification.

To model the sequence dependency, message type sequence is considered. M. Korczynski and A. Duda provided a new thought, using first-order Markov chain as application fingerprinting [5]. It takes full advantages of the message type field in server-side SSL/TLS packet to establish the corresponding Markov chain as fingerprinting. Shen Meng et al. successively put forward [12] and [13] to improve the classification accuracy by means of higher Markov chain model and other features extracted from SSL flows. Although these methods can improve the results to a certain degree, they may have the misclassification because of several reasons. One is that the overlaps of message type sequences from different applications is high. The other is that the limited order Markov model only considers several states instead of all the states in flows.

The packet length sequences are also taken into account by researchers. [6] discretize packet lengths into equally sized bins. However, the discretization will fuzz the packet relevance and lose several original information for classification. [7] considers the message type sequences and length sequences, and uses the hidden Markov models and weighed ensemble learning model, which can perform well due to more features considered, but also increase system computation loads of both training and testing.

To address the above limitations, we propose the LaFFT fingerprinting based on only server-side packets of flows, which transforms packet length sequences in time domain into frequency domain features.

III. Data Collection

In this section, we introduce the approaches to make up datasets and acquire labels.

Raw Data. We collect all the traces of traffic flows on specific routers in a campus network for 7*24 hours long traces starting from July 20, 2017, and then we filter the non-SSL/TLS encrypted traffic. Finally, we get 1.6 million SSL/TLS flows (18.6 million packets) during the one-week long traffic traces. Considering the diversity and complexity of different clients, we only focus on the server-side traffic traces to identify applications, referring to [5] and [12].

TABLE I
10 Applications Traffic Data List

ID	Applications	Strings in Domain Names	Flows	Packets
1	Alicdn	*.alicdn	16,560	124,206
2	Alipay	*.alipay	20,299	137,542
3	Baidu	*.baidu, *.bdstatic	373,177	2,500,996
4	Github	*.github.com, *.github.io	7,488	84,618
5	iCloud	*.icloud	22,993	150,278
6	JD	*.jd.*	48,146	177,041
7	Mozilla	*.mozilla.*, *.cdn.mozilla.*	4,265	29,596
8	OneNote	*.onenote.*	6,486	52,840
9	Sogou	*.sogou.com	4,498	24,251
10	Taobao	*.taobao.com	17,267	127,501

* IDs are the corresponding codes of applications, which are used in later results of other contrast experiments.

Ground Truth Label. All the traffic flows captured are not labeled with the corresponding applications, which is an obstacle for training models and evaluating algorithms. Therefore, we establish the ground truth with two steps. The first step is to extract the value of Server Name Indication (SNI) Extension in Client Hello packet of each flow. Normally, SNI value corresponds to the domain name of one application. However, fake SNI values [14] and informal implementations of the SSL/TLS protocol [15] exist universally, which weaken the reliability of SNI values. We use the second step to enhance the dependability of SNI value referring to [12], and select an open web service, Whois [16] which can reversely query their domain names according to IP addresses. If the substring of the domain name is consistent with the substring of SNI value, we can confirm which application the flow belongs to. This approach can indeed build a ground truth dataset to a certain degree, but some constraints exist, such as parsing failure of Whois, no SNI value and insufficient substrings.

After packet recombination and flow reduction techniques, over 510 thousand traffic flows (about 3.4 million packets) referring to 10 applications are remained as our dataset shown in Table I. Because of real-time capture, the numbers of different applications vary greatly.

IV. Methodology

We introduce LaFFT fingerprinting, demonstrate its discrimination and describe the selected classification algorithm.

A. LaFFT Fingerprinting

Discrete Fourier Transform (DFT) can transform the signals in time domain into the frequency domain. The transformation equation is:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk},\ k = 0, 1, \cdots, N-1 \tag{1}$$

where $X(k)$ is the frequency value at position k, $x(n)$ is the time series at the n-th timestamp, and $W_N = e^{-j\frac{2\pi}{N}}$. Fast Fourier Transformation (FFT) is an efficient algorithm for calculation of DFT, and takes full advantage of the symmetry

and parity [17] of the twiddle factor W_N with algorithm complexity decreasing from $O(N^2)$ to $O(NlogN)$.

To obtain the FFT-based fingerprinting, the first N packets of one flow are considered. We pad a flow with zero if there are no enough packets in the flow to perform FFT. Thus, flows with different lengths in time domain are transformed into feature vectors with the same length in frequency domain. We term these features as Length-aware Fast Fourier Transformation (LaFFT) fingerprinting. The frequency LaFFT features are complex numbers, and we extract the modulus and angle of each dimension to consist the real-value features. The two kinds of features can provide explainabilities, because the modulus is the squire root of energy at a specific frequency and the angle is the initial phase. Due to the symmetry of LaFFT features, only the first half dimensions of both modulus and phase angle feature vectors are selected and concatenated as application LaFFT fingerprinting.

Compared with the original length sequences, the frequency LaFFT fingerprinting enjoy several advantages:

- Each dimension of the LaFFT fingerprinting are non-ordered which is more suitable for classification.
- The vectors (including both modulus and phase angle vectors) generated by FFT are more representative because of taking many packets in one flow into consideration, which increases the discriminations of different applications' flows.
- The computation complexity of FFT is much low, especially with good DSP hardware support [18], which keeps real-time classification with low loads in real-world network environment.
- Each dimension has a real meaning, which can be reasonable to explain and analysis the classification results.

B. The Distinguishability of LaFFT Fingerprinting

We analysis the dataset described in Section III. Each specific flow will produce a specific LaFFT fingerprinting. To learn the statistical characteristics of LaFFT fingerprinting, we take 256-point-FFT on each flow. For each application, we obtain the mean and median value on each dimension from all the LaFFT fingerprinting of this application. The mean and median curves of modulus and phase are show in Figure 1 respectively. No applications have similar modulus and phase curves in both the mean and median curves, which indicate the LaFFT fingerprinting is suitable for application classification.

The mean and median curves show different statistical characteristics of flows of a specific application, and can demonstrate the distribution of this application from different perspectives. For example, the overlap of the median value curves of Alicdn and Mozilla is high from Figure 1(b) and 1(d). However, their mean curves with shown in Figure 1(a) and 1(c) demonstrate the difference of their distributions, which is more clear to show the distinction of LaFFT fingerprinting.

Comparing the curves of applications in Figure 1, the LaFFT features presents great differentiations among applications, i.e. different dimensions of many applications are obviously different from the statistical perspective. Meanwhile, a majority of applications are unique according to the whole curves. Moreover, combing both the modulus and phase parts of LaFFT fingerprints bring discriminability for the application classification. The modulus and phase at a specific frequency position show the amplitude and initial phase of the "signal" (i.e., packet length sequence) in time domain at the selected frequency. Several "signals" have same amplitude (i.e. modulus in frequency domain), but their initial phase can result in the distinction. For example, the JD and Baidu have similar modulus curves, but their phase curves are significantly different. They can be classified properly, the experimental results in Section V also support that both the modulus and phase parts of LaFFT is important in classification.

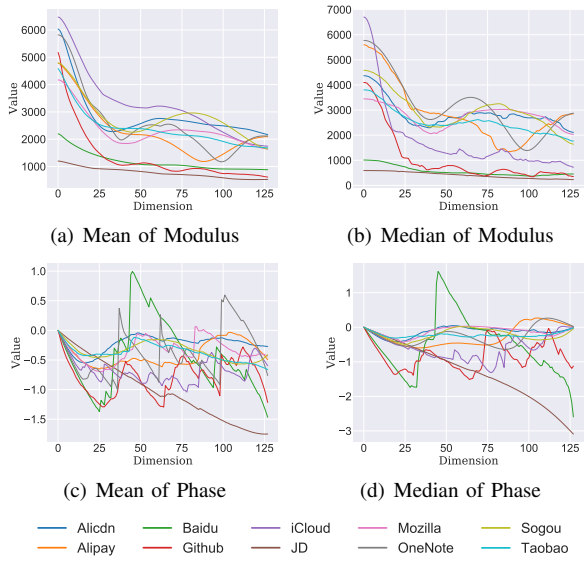


Fig. 1. The mean and median modulus curves of LaFFT fingerprinting are shown in (a) and (b), and the mean and median phase curves of LaFFT fingerprinting are presented in (c) and (d).

C. Classification Algorithms

The LaFFT features are fed into a classifier to obtain the final prediction label. In this paper, we take RF as the classifier. RF contains multiple decision trees, and the final label is decided by the votes of all the decision trees. When establishing each decision tree, a subset of the samples and a random subset of features are used, in order to reduce the dependence of trees. RF enjoys several advantages: 1) Extensive applicability. RF is suitable for both non-linear and linear classification problems, especially when the decision planes are complex. The LaFFT fingerprinting for applications are a non-linear problem, which we will show in Section V.C. 2) Robustness. Based on the bagging ensemble method, RF is under low risk of overfitting. In another words, the overfitting risk is lower with more decision trees. 3) Feature Selection. RF can provide the importance of each feature during the training process. Intuitively, a feature is more important if

it serves as the split point in more different trees. And this can help us to select efficient features and reduce both the time and space complexity. 4) Fast Training. RF can find non-linear decision planes quickly because fitting a decision tree is relatively simpler than other non-linear methods, e.g. kernel SVM. Moreover, each decision tree is independent and can be trained in parallel, which can accelerate the training process.

V. Experiments

We introduce our experiments in detail and analyze LaFFT fingerprinting in depth.

A. Experiment Setting

1) Cross Validation: In consideration of our dataset, we use 5-fold cross validation to evaluate the reliability and stability of different methods. The average of classification results are regarded as the final performance.

2) Assessment Criteria: We use True Positive Rate (TPR), False Positive Ratio (FPR) and FTF as assessment criteria.

TPR means the ratio of predicting positive samples as positive and all positive samples, while FPR means the ratio of predicting negative samples as positive and all negative samples. Because of different flow numbers, we define TPR_{AVE} as the ratio of the classified rightly flow number and the total flow number in Eq. (2), and FPR_{AVE} as the ratio of the classified wrongly flow number and the total flow number in Eq. (3).

$$TPR_{AVE} = \frac{1}{AFlN} \sum_{i=0}^{n} TPR_i * FlN_i \tag{2}$$

$$FPR_{AVE} = \frac{1}{AFlN} \sum_{i=0}^{n} FPR_i * FlN_i \tag{3}$$

where n means the number of applications, e.g. 10 applications in our dataset. TPR_i and FPR_i means two indicators of application i. FlN_i represents the flow number of application i and $AFlN$ represents the total traffic flow number. Therefore, TPR_{AVE} and FPR_{AVE} give two overall classification measures of all the traffic rather than considering the specific application separately.

Generally, results with higher TPR and lower FPR are better. For reasonable comparison with the existing methods, we refer to the criteria of [13], FTF, which considers both TPR and FPR to more visibly reveal the effectiveness of methods in Eq. (4).

$$FTF = \sum_{i=0}^{n} w_i \frac{TPR_i}{1 + FPR_i} \tag{4}$$

where w_i means the weight of each application i. Obviously, higher TPR_i and lower FPR_i makes the classification result more excellent, in accordance with our expectations. Moreover, FTF takes the different weights of applications into consideration, which represents different applications can affect the final classification accuracy with various degrees. In this paper, we let w_i be the ratio of the flow number of one application i and all the flow number because one application may be more important when there are more its traffic flows.

3) Contrast Methods: Here, we list all the comparison methods.

- *FoSM*, namely First-order State Markov chain fingerprint, which uses state sequences to build first-order Markov chain to predict [5].
- *SoSM*, which is analogous to FoSM, but using second-order Markov chain as application fingerprint [12].
- *SOCRT*, which considers the probability distribution of certificate packet lengths and the probability of SoSM [12] to classify applications.
- *SOB*, which adds the distribution probability of both certificate packet lengths and the first communication packet lengths to SoSM to predict applications [13].
- *LaFFT(LR)*, which combines LaFFT features of packet length sequence and Logistic Regression (LR) Classifier.
- *LaFFT(LR)+SoSM*, which considers both the probabilities of LaFFT(LR) and SoSM to classify applications.
- *LaFFT(RF)*, is our method, namely RF Classifier is applied to LaFFT features of packet length sequence.
- *LaFFT(RF)+SoSM*, which unites LaFFT(RF) and SoSM probabilities to predict classification.

B. Experiment Results

We contrast LaFFT with the state-of-the-art methods and the variant methods of LaFFT whose results are all shown in Table II. Obviously, LaFFT(RF) can get 0.959 FTF on classification of 10 applications, far better than other methods.

(1) FoSM (0.745 FTF) and SoSM (0.776 FTF) only focus on the message type of each packet in the flow, which exists unilateralism. In other words, message types of the SSL/TLS protocol are limited which leads to the limitation of built-up sequences and feature discrimination. Although higher order Markov chain can improve the final classification results in some degree, the transformation matrix becomes larger and more sparse. The experiment results also demonstrate both of them can't classify Taobao, Alipay and Github in accuracy. However, LaFFT considers the whole packet length sequences, and the diversity of packet lengths is obviously more than that of message types, which improve the classification results of LaFFT. **(2)** Both SOCRT (0.787 FTF) and SOB (0.791 FTF) import more features (i.e. certificate packet length and the first communication packet length) into message type sequences as features, and the final classification results are improved, but the promotion effect is not significant in our dataset. Since certificate packet lengths and the first communication packet lengths of different applications could be clustered together which reduces the differences. Therefore, misclassification is inevitable and they even reduce the accuracy of some applications with SoSM, such as Alicdn, Github and Mozilla. In contrast, LaFFT fully remains their all raw length information. Based on the diversity of packet lengths, LaFFT also takes many packets into consideration to generate the fingerprinting. Therefore, LaFFT (0.959 FTF) obviously has huge superiority.

For LaFFT fingerprinting, we adopt the representative LR (linear classifier) and RF (non-linear classifier) to build

TABLE II
RESULT COMPARISON BETWEEN TRADITIONAL MARKOV AND FSLPM

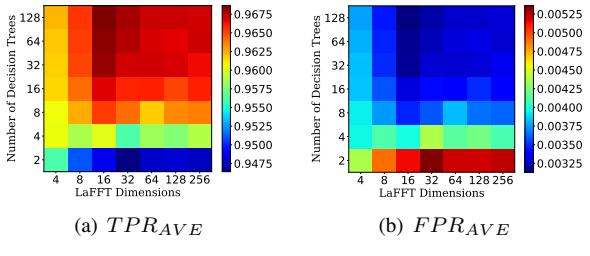
ID	FoSM		SoSM		SOCRT		SOB		LaFFT(LR)		LaFFT(LR)+SoSM		LaFFT(RF)		LaFFT(RF)+SoSM	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
1	0.747	0.0200	0.783	0.0261	0.546	0.0284	0.673	0.0319	0.120	0.0054	0.718	0.0141	0.901	**0.0042**	**0.919**	0.0086
2	0.582	0.0174	0.673	0.0375	0.718	0.0280	0.722	0.0261	0.134	0.0030	0.678	0.0240	0.900	**0.0035**	**0.911**	0.0104
3	0.761	**0.0007**	0.796	0.0023	0.823	0.0042	0.824	0.0029	0.980	0.2424	0.857	0.0210	**0.983**	0.0122	0.962	0.0032
4	0.558	0.0169	0.444	0.0089	0.343	0.0102	0.462	0.0099	0.281	0.0028	0.507	0.0057	**0.851**	**0.0009**	0.840	0.0030
5	0.956	0.0189	0.961	0.0199	0.958	0.0075	0.857	0.0051	0.246	0.0067	0.964	0.0158	0.960	**0.0014**	**0.984**	0.0027
6	0.898	0.0705	0.880	0.0394	0.882	0.0432	0.879	0.0415	0.019	0.0097	0.894	0.0600	**0.962**	**0.0034**	0.958	0.0044
7	0.800	0.0078	0.830	0.0070	0.523	0.0080	0.521	0.0062	0.000	**0.0000**	0.681	0.0043	**0.951**	0.0006	0.944	0.0005
8	0.986	0.0115	0.973	0.0068	0.666	0.0063	0.860	0.0122	0.000	0.0016	0.970	0.0048	**0.994**	**0.0002**	0.991	0.0005
9	0.806	0.0676	0.765	0.0431	0.659	0.0347	0.652	0.0334	0.000	**0.0000**	0.475	0.0043	**0.930**	0.0006	0.927	0.0016
10	0.082	0.0158	0.187	0.0260	0.433	0.0340	0.359	0.0329	0.008	**0.0006**	0.285	0.0197	0.878	0.0046	**0.898**	0.0102
AVE	0.753	0.0247	0.783	0.0217	0.796	0.0204	0.798	0.0202	0.728	0.0272	0.826	0.0174	**0.968**	**0.0032**	0.955	0.0045
FTF	0.745		0.776		0.787		0.791		0.591		0.807		**0.959**		0.951	

* The parameter of LaFFT dimension is 16 for all the LaFFT methods.

the classification models with FFT features. **(3)** Apparently, LaFFT(LR)+SoSM (0.807 FTF) is better than LaFFT(LR) (0.591 FTF), but worse than LaFFT(RF)+SoSM (0.951 FTF), and the best model is LaFFT(RF) (0.959 FTF), a little better than LAFFT(RF)+SoSM. The results demonstrate the LaFFT fingerprinting is nonlinear separable, and more detailed analysis will be presented in section V.C. **(4)** SoSM improves the TPR_{AVE} of LaFFT(LR) 72.8% to 82.6% and reduces the FPR_{AVE} from 2.72% to 1.74%. However, it is not much helpful for the classification of LaFFT(RF), and even reduces the accuracy on the contrary. The main reason may be the overlap of message type sequences of different applications could be high which finally disturbs the results of LaFFT(RF).

C. LaFFT Model Analysis

We analyze the parameter sensitivity of LaFFT fingerprinting, and show the front superiority and linear inseparability of LaFFT fingerprinting. We conduct the following experiments on the LaFFT(RF) model.



(a) TPR_{AVE} (b) FPR_{AVE}

(c) FTF (d) $Time/s$

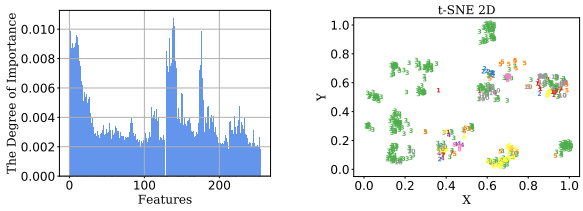
Fig. 2. LaFFT dimension and number of decision trees analysis of LaFFT

1) Parameter Sensitivity Analysis: For optimal performance of encrypted traffic classification, we need to adjust two main LaFFT parameters, i.e. LaFFT dimension and the number of decision trees. LaFFT dimension directly affects how many packets are considered to generate LaFFT features. As it increases, although more packets in long flows can be integrated into LaFFT features, there are more 0 as substitution to produce final features for short flows. In addition, on the basis of principles and characteristics of RF, more decision trees can reduce the overfitting but increase the training time. Therefore, reasonable LaFFT dimension and the number of decision trees need to be further considered.

We display the classification results with LaFFT dimension from $\{4, 8, 16, 32, 64, 128, 256\}$ and the number of decision trees from $\{2, 4, 8, 16, 32, 64, 128\}$ in Figure 2. No matter in TPR_{AVE} (Figure 2(a)), FPR_{AVE} (Figure 2(b)) and FTF (Figure 2(c)), we can see more decision trees can indeed produce better results. This is consistent with our cognition (i.e. more decision trees make validation results more accuracy because of training with a low overfitting). But for LaFFT dimension, TPR_{AVE}, FPR_{AVE} and FTF all climb up first and then decline, which is the same with our thought that the balance between more packets and less 0 can significantly affect the results. From Figure 2(d), we can conclude that the training time increases rapidly with the growth of either of these two parameters.

In this paper, we target at our dataset described in Section III, and set 16 as LaFFT dimension and 128 as the number of decision trees, considering time and accuracy simultaneously. And for other scenarios, we advise reasonable LaFFT dimension and as many decision trees as possible for better performances.

2) The Front Superiority of LaFFT Fingerprinting: We conclude that 16 as LaFFT dimension can get better performance than higher LaFFT dimensions from the above parameter sensitivity analysis. To explain the phenomenon, we analyze the importance degree of each LaFFT feature. Specifically, we set 128 as the number of decision trees and 256 as LaFFT dimension, and then calculate the probability distribution of LaFFT features. Obviously, higher probability represents that the corresponding LaFFT feature is more significant. The analysis result is shown in Figure 3(a) where



(a) Feature Importance (b) Nolinear Characteristic

Fig. 3. LaFFT features analysis

the first several features, i.e. the low-frequency features, make more contributions to the final results, no matter in modulus and phase angle, which is consistent with the conclusion of parameter sensitivity. We speculate the reason: importing more LaFFT features leads to the overfitting of every decision tree, which further increases the overfitting risk of RF once the number of decision trees is determined.

3) The Linear Inseparability of LaFFT Fingerprinting : From Table II, we can know the performance of LaFFT(LR) is far worse than LaFFT(RF). Here we use dimensionality reduction to give a visual interpretation of the linear inseparability of LaFFT fingerprinting. T-Distributed Stochastic Neighbor Embedding (t-SNE) [19] is a famous technique for dimensionality reduction, which is particularly suited for the visualization of high-dimensional datasets. T-SNE can keep the relative distance, i.e. two points with long distance on high-dimensional space map a long distance between these two points on low-dimensional space. Therefore, t-SNE can approximately represent the distribution of data. We randomly extract 1% flows from our dataset, and apply t-SNE to this subdataset. The distributions of 10 applications are displayed in Figure 3(b). From this figure, these 10 applications obviously cannot be distinguished by linear classifiers. For example, the application with the label of 3, namely Baidu, distributes in a lot of areas and cannot be separated with others by straight lines. Therefore, LaFFT fingerprinting are highly linear inseparable and cannot be classified by linear classifiers, which results in that the performance of LaFFT(LR) is far worse than LaFFT(RF).

VI. Conclusion

In this paper, we purpose LaFFT fingerprinting to identify encrypted traffic of different applications. Specifically, we apply FFT to packet length sequences to generate feature vectors in frequency domain as LaFFT fingerprinting that contains the packet relativity information of each flow. We confirm the distinguishability of LaFFT fingerprinting for different applications. Furthermore, the linear inseparability and the front superiority of LaFFT fingerprinting are demonstrated, which advises us to choose the random forest as classifier. With comprehensive comparison experiments, the LaFFT fingerprinting with random forest classifiers can achieve the best performance on a real-world dataset, with 96.8% TPR, 0.32% FPR and 0.959 FTF, outperforming the state-of-the-art methods significantly.

Acknowledgment

This work is supported by The National Key Research and Development Program of China (No.2016QY05X1000 and No.2016YFB0801200) and The National Natural Science Foundation of China (No.61602472). Research is also supported by the CAS/SAFEA International Partnership Program for Creative Research Teams and IIE, CAS international cooperation project. Zigang Cao is the corresponding author.

References

[1] V. F. Taylor, R. Spolaor, M. Conti, and I. Martinovic, "Robust smartphone app identification via encrypted network traffic analysis," *IEEE TIFS*, vol. 13, no. 1, pp. 63–78, 2018.

[2] S.-H. Yoon, J.-W. Park, J.-S. Park, Y.-S. Oh, and M.-S. Kim, "Internet application traffic classification using fixed ip-port," *Management Enabling the Future Internet for Changing Business and New Computing Services*, pp. 21–30, 2009.

[3] F. Risso, M. Baldi, O. Morandi, A. Baldini, and P. Monclus, "Lightweight, payload-based traffic classification: An experimental evaluation," in *IEEE ICC'08*. IEEE, 2008, pp. 5869–5875.

[4] J. Liu, Y. Fu, J. Ming, Y. Ren, L. Sun, and H. Xiong, "Effective and real-time in-app activity analysis in encrypted internet traffic streams," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 335–344.

[5] M. Korczynski and A. Duda, "Markov chain fingerprinting to classify encrypted traffic," in *INFOCOM'14 IEEE*, 2013, pp. 781–789.

[6] B. Anderson, S. Paul, and D. Mcgrew, "Deciphering malwares use of tls (without decryption)," *Journal of Computer Virology & Hacking Techniques*, pp. 1–17, 2016.

[7] W. Pan, G. Cheng, and Y. Tang, "Wenc: Https encrypted traffic classification using weighted ensemble learning and markov chain," in *Trustcom/bigdatase/icess*, 2017.

[8] P. Velan, "A survey of methods for encrypted traffic classification and analysis," *International Journal of Network Management*, vol. 25, no. 5, pp. 355–374, 2015.

[9] S. M. Kim, Y. H. Goo, M. S. Kim, S. G. Choi, and M. J. Choi, "A method for service identification of ssl/tls encrypted traffic with the relation of session id and server ip," in *Network Operations and Management Symposium*, 2015, pp. 487–490.

[10] G. L. Sun, Y. Xue, Y. Dong, and D. Wang, "An novel hybrid method for effectively classifying encrypted traffic," in *Global Telecommunications Conference*, 2010, pp. 1–5.

[11] W. M. Shbair, T. Cholez, J. Francois, and I. Chrisment, "A multi-level framework to identify https services," in *Network Operations and Management Symposium*, 2016.

[12] M. Shen, M. Wei, L. Zhu, M. Wang, and F. Li, "Certificate-aware encrypted traffic classification using second-order markov chain," in *IWQoS'16 IEEE/ACM*. IEEE, 2016, pp. 1–10.

[13] M. Shen, M. Wei, L. Zhu, and M. Wang, "Classification of encrypted traffic with second-order markov chains and application attribute bigrams," *IEEE Transactions on Information Forensics & Security*, vol. PP, no. 99, pp. 1–1, 2017.

[14] W. M. Shbair, T. Cholez, A. Goichot, and I. Chrisment, "Efficiently bypassing sni-based https filtering," in *Ifip/ieee International Symposium on Integrated Network Management*, 2015, pp. 990–995.

[15] W. M. Shbair, T. Cholez, J. Francois, and I. Chrisment, "Improving sni-based https security monitoring," in *IEEE International Conference on Distributed Computing Systems Workshops*, 2016, pp. 72–77.

[16] P. T. Endo and D. F. H. Sadok, "Whois based geolocation: A strategy to geolocate internet hosts," in *IEEE AINA*, 2010, pp. 408–413.

[17] E. O. Brigham and E. Brigham, *The fast Fourier transform and its applications*. prentice Hall Englewood Cliffs, NJ, 1988, vol. 1.

[18] B. M. Baas, "A low-power, high-performance, 1024-point fft processor," vol. 34, no. 3, pp. 380–387, 1999.

[19] G. E. Hinton, "Visualizing high-dimensional data using t-sne," *Vigiliae Christianae*, vol. 9, no. 2, pp. 2579–2605, 2008.