High-Efficient and Few-shot Adaptive Encrypted Traffic Classification with Deep Tree

Qiang Wang
Institute of Information Engineering
Chinese Academy of Sciences
owangqiang@qq.com

Wenhao Li
Institute of Information Engineering
Chinese Academy of Sciences
liwenhao@iie.ac.cn

Huaifeng Bao
Institute of Information Engineering
Chinese Academy of Sciences
baohuaifeng@iie.ac.cn

Zixian Tang
Institute of Information Engineering
Chinese Academy of Sciences
tangzixian@iie.ac.cn

Wen Wang
Institute of Information Engineering
Chinese Academy of Sciences
wangwen@iie.ac.cn

Feng Liu
Institute of Information Engineering
Chinese Academy of Sciences
liufeng@iie.ac.cn

Lingyun Ying
Qi-AnXin Technology Research Institute
yinglingyun@qianxin.com

MILCOM 2022 - 2022 IEEE Military Communications Conference (MILCOM) | 978-1-6654-8534-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/MILCOM55135.2022.10017968

***Abstract*—Although network traffic classification has been investigated for decades, the core challenges, including the complex and capricious conditions of network traffic, and the practical application of models, remain unsolved. Meanwhile, the extensive usage of encryption protocols makes encrypted traffic classification become a new challenge. The rapid iteration of network traffic brings the scale drift of encrypted traffic classification. While bulky deep-learning-based methods can barely satisfy the lightweight demand in real-world scenarios.**

To solve this, we propose a efficient encrypted traffic classification method using Deep-Tree with multi-grained scanning and cascade tree to perform high-speed learning and multi classification task. It has the classification accuracy and representation ability of depth model with lightweight computing expenses. The self-adaption and expandable ability of the model make it suit different traffic scenarios without specific model adaptation. The experimental results show that the proposed method achieves superior performance compared with state-of-the-art methods. Particularly, our method can dynamically adapt traffic classification tasks at different scales.

***Index Terms*—Encrypted Traffic Classification, Model Adaption, Lightweight Computation, High-Efficient Implementation**

I. Introduction

Network traffic classification is crucial in network management and security. In the task of ensuring the safe operation of the network, the execution efficiency and identification accuracy are the research focus of the current traffic classification. Traffic classification has been extensively studied, and several approaches have been proposed [1]–[4]. Traffic classification is widely deployed in the industry to fulfill the requirements of network management and security infrastructure [5]. However, their wide applicability and high accuracy are core demands and competence.

With the popularity of secure transport protocols, such as HTTPS, encrypted network traffic has become mainstream [6]. Consequently, the original sequence features of the network traffic are distorted by encryption. The network traffic payloads are no longer visible to traffic analysis systems as well, which makes payload-based feature engineering considerably limited, posing a significant challenge for traffic management. Classic signature-based approaches are not applicable to encrypted traffic classification. The original sequence features of the network traffic are distorted by encryption.

Traditional methods can be used for traffic classification [7]. However, they require heavy manual feature engineering. It relies on a deep understanding of malicious traffic in specific scenarios to build a feature set that can be used for effective classification using machine-learning methods. Machine-learning-based network classification techniques focus on modelling the statistical features of data flow. It extracts feature vectors through side-channel signals, such as inter-packet-sequence and intra-payload-content, from the packet files stored on the disk [8]. Consequently, the combined statistical features remarked in relatively long time segments were used to represent the network flow. However, the statistical method can be easily bypassed by a hacker through a simple black-box test [9]. Moreover, both statistical and machine-learning-based methods mainly focus on model optimisation based on datasets [5]. Little attention has been paid to the execution efficiency of a model or its rapid adaptation application in the real world.

With the rise of deep-learning models that perform image classification tasks with high accuracy, deep-learning-based methods have been applied to traffic classifications [10] and have achieved superior accuracy compared with machine-learning-based methods. However, the traffic data must be pre-processed to the data format that can be fed to deep-learning models (e.g., transform to a grayscale image), which introduces performance losses and makes it difficult to implement in high-throughput real-world scenarios. The complexity

of the previous deep-learning network structure makes it a challenge for deploying in real-world scenarios. Furthermore, the insufficient interpretability of deep-learning models limits their applications in traffic classification which may require explanations of the features and classification process.

Generally, we summarise the following challenges in real-world network traffic classification tasks.

1) The traffic possess complex and dynamically changing properties in nature. Machine-learning-based methods require a deep understanding of traffic behavior to build an efficient feature.
2) Deep-learning-based methods is known to lack interpretability. And the packets often need pre-processed when using deep-learning-based methods,which makes it difficult to meet the demand for high-throughput network.
3) Continuous iterative training is necessary in the realistic of malware rapidly changing. But current data driven classification algorithms require too much overhead in time consumption and computational complexity.
4) Network traffic is highly unbalanced, particularly in malicious traffic, which considerably lacks ground-truth labels. This condition makes it urgent for the model to adapt to few-shot classification requirement.

To address the challenges above, we introduce a novel and efficient Deep-Tree based pipeline structure, which uses a multi-grained cascade Deep-Tree to perform high-speed representation learning, for network traffic classification. The pipeline structure adapts to large-scale encrypted traffic classification by swallowing the raw bytes from network traffic, which enables the proposed model to be established directly behind the traffic-acquisition interface without middleware. The representation capability of the proposed model combined with the deep-learning model enables it to achieve high accuracy in a limited training-sample quantity. The complexity and depth of the model can be adapted according to the scale of the datasets owing to the distinct representation structure design. Therefore, it is applicable to a diversified traffic-classification scenario. The main contributions of this study are as follows.

- We propose using the Deep-Tree model for encrypted traffic classification that can efficiently achieve representation learning, reduce the feature engineering overhead that requires considerable professional knowledge. The training time economizaition makes retraining rapidly so as to adapt to complex and frequently changeable network traffic.
- The model complexity can be extended adaptively according to the dataset, which ensures classification accuracy and reduces the calculation cost. This is generally sufficient to accommodate a variety of classification scenarios. Compared to the neural-network-based deep model, the proposed model is more efficient and lightweight in traffic classification.
- In four distinct encrypted network traffic datasets scenarios, the proposed method outperforms state-of-the-art models in classification accuracy without significant performance degradation in few-shot scenarios.

II. Preliminaries

A. Encrypted Traffic Classification

Currently, traffic analysis faces various challenges [5]. With the continuous growth in traffic scenarios and different classification and recognition task environments, an efficient, accurate, easy training, and cross-scenario traffic-classification model is urgently required.

The traffic feature engineering of machine learning is manpower-intensive [6]. The prompt extraction of features from data such that they can be accessed efficiently during online network traffic analysis determines whether the model can be used in the production environment. Owing to the requirement of traffic analysis throughput, the pre-processing degree of features is generally light, and more pre-processing is left to the model, which increases the delay in model reasoning [10]. This is unfavourable for the real-world performance of traffic analysis. The addition of relevant features to the model is a reliable method for boosting the analysis rate. However, identifying new features that actually improve performance can be a slow and tedious process of trial and error.

Deep learning [11] has a powerful representation capability, which weakens the importance of feature engineering and enables end-to-end learning. However, the complexity of the deep model makes it a black box. Consequently, the credibility of the model is invisible. In network security, particularly in traffic-detection scenarios, uninterrupted alarms cannot be handled properly. The related papers and the comparison works are as follows. FS-Net [12] is an end-to-end classification model, which adopts a multi-layer encoder decoder structure to mine the potential sequential features of the stream. FC-net [13] is a deep neural network (DNN) , which is mainly composed of feature extraction network and comparison network.

B. Deep-Tree

Deep-Neural-based network structure are being widely used presently. With the superiority of automatic feature extraction and flexible layer stacking, the end-to-end Deep Neural Networks can promise better feature representation [14].

However, Deep-Neural-Network has high requirements for big training data, hyperparameter optimisation, and computing power. Compared with the classical deep model, Deep-Tree has lower dependence on hyperparameters and lightweight calculate resource cost. The pre training process is more automated and more efficient, which suits the scenario with fast iterations in traffic classification. Therefore, Deep-Tree can replace high-overhead neural-network-based deep learning in traffic classification task.

Deep-Tree is a novel decision-tree ensemble with a cascade structure, that enables representation learning by forests. Deep-Tree is an approach that unifies classification trees with the representation learning functionality known from deep-neural-networks by training them in an end-to-end manner.

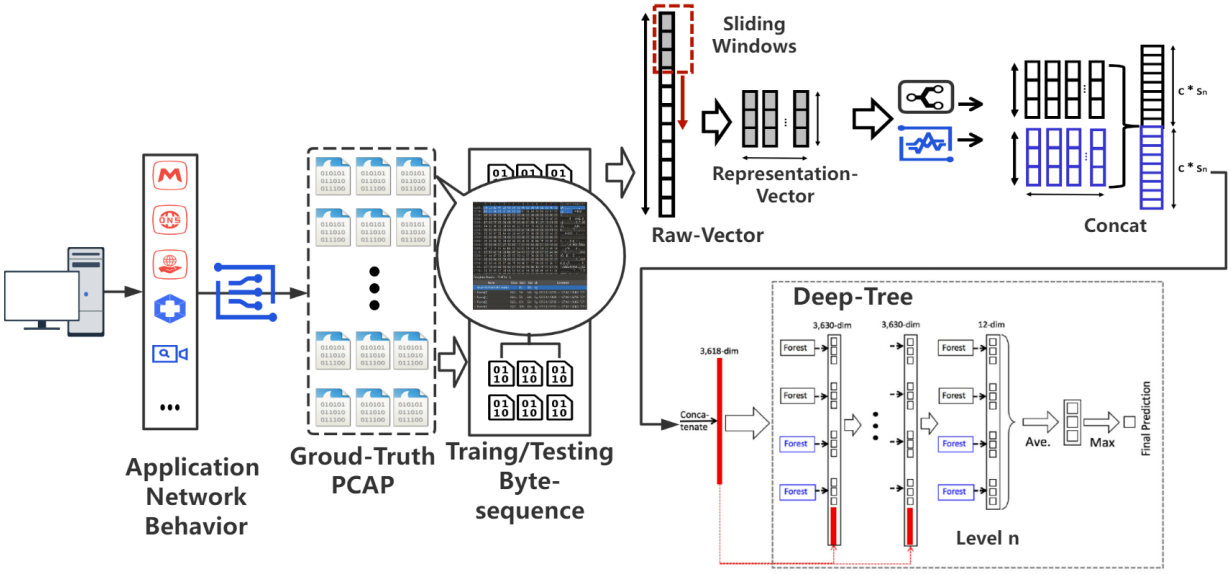

Fig. 1: The framework of Deep-Tree-based Encrypted Traffic Classification.

Inspired by layer-by-layer processing, Deep-Tree adopts a cascade structure. Each level of cascade forest contains several ensemble learning classifiers. The stack of Deep-Tree aims to improve the classification ability of input data through cascade forest and reflect the difference in input data through multi-granularity scanning.

Traffic classification scenarios often encounter extremely unbalanced training datasets. Conventional machine-learning-based methods, such as boosting integration algorithms, only deal with the given features. The complexity of the model was limited. They cannot transform the features inside the model to learn high-dimensional features. Conversely, the depth iteration makes Deep-Tree well demonstration application effect. The design of Deep-Tree contains multi-granularity scanning, which expands the feature dimension and provides better adaptability to few-shot and unbalanced realistic conditions.

III. System Model

In this section, we introduce the proposed Deep-Tree encrypted traffic classification model on a multi-grained cascade forest. This tree-based integration method integrates traditional forest models in terms of breadth and depth. The proposed method has efficient training and accurate classification. It is more adaptive compared to the classical deep model in a changeable large-scale traffic-classification scenario.

Figure 1 shows the main framework of the proposed Deep-Tree encrypted traffic classification. The proposed model consists of Traffic2Byte-sequence, traffic-mapping model and Deep-Tree classifier. In the subsequent sections, we introduce each part of the proposed model.

A. Traffic2Byte-sequence

Network traffic is reflected from the observer's perspective, which is a running byte-stream sequence. Therefore, it is necessary to design a classifier with a pipeline structure. The framework first divides traffic into metadata according to the network five-tuple. In the network traffic analysis process, the most commonly used sample granularity is the PCAP packet. For encrypted network traffic, the segmentation logic divides it into the interactive head of the encrypted connection and payload transmission after handshake. Generally, the initiation and protocol structure of encrypted connections head provide considerable message for the analysis of encrypted traffic. So that the proposed model is designed to takes the original byte of the first N bytes of the network stream as the input, cuts it out, and provides it to the designed multi-granularity Deep-Tree classifier.

Each byte is used as a single-dimensional feature, which effectively avoids feature engineering based on professional knowledge. The network traffic scenarios is volatile. For example, the enterprise baseline iterates rapidly with business changes, and the network-traffic classification model for boundary protection must be updated over time. This method of taking raw byte as input, which is referred to as 'Eating a whole EXE' [15], becomes feasible to establish a simple and fast iterative traffic classifier. It is only applied by the industry with practical value, and the iterative ability of the model is connected to existing traffic classification infrastructure.

The output of Traffic2byte-sequence is the raw stream after the head N traffic is cut. In the subsection, we introduce the model design for the trainer and classifier.

B. Encrypted Network Traffic Classification with Deep-Tree

Following the design of Deep-Neural-Networks [14], the original features are processed layer-by-layer. The plain structure of the cascade forest is shown at the top right of Figure 1. The raw streams are inputted into the classifier Deep-Tree model, which is a new designed tree-based deep-learning

method. It uses a series-ensemble-learning classifier to achieve representation learning.

The cascade forest structure uses different integrated operators (Random Forest is selected as an example) in each layer. In this example, two completely-Random Forests and two Random Forests were used. Each completely-Random Forest contained 1000 completely-random trees. Each tree was generated by randomly selecting a feature and dividing it at each node of the tree. The trees were expanded until each leaf node contained only instances of the same class, or no more than 10 instances. Similarly, each RF contained 1000 trees. $\sqrt{d}$ dimensions of features were randomly selected as candidates, where d is the number of input features. Then the feature with the most Gini score was selected as the segmentation.

Given an input sample, each forest calculated the percentage of training samples of different classes at the leaf nodes where the relevant instances fell, and averaged all trees in the forest to generate an estimate of the distribution of classes. The estimated class distribution formed a class vector. The class vectors generated by each forest were generated by k-fold cross validation and averaged to generate the final enhancement vector to reduce overfitting. Each forest generated a probability vector with length c (target-classification number). If each layer of Deep-Tree was composed of N forests, the output of each layer was $N * c$ dimensional vectors connected together, that is, $c * N$ dimensional vectors (called enhanced vector-augmented feature). These vectors were then connected to the original eigenvector input in the next layer of the cascade. Thus, we made a feature change and retained the original features for subsequent processing. This was the output of the first layer. The input of each layer of the subsequent cascade forest exhibited the same splicing.

Before extending a new layer, the performance of the current model was estimated using the verification dataset. If there is no significant performance gain, the training process terminates, and the number of levels in the cascade structure is automatically determined. Unlike most deep neural networks with fixed model complexity, Deep-Tree can appropriately determine its model complexity (early stop) by terminating training. This makes the Deep-Tree suitable for different scale-training data rather than being limited to large-scale requirement. This simplified the training process. A model that can be easily trained is particularly required for traffic classification across datasets and applications.

C. Multi-grained Scanning

We used a multi-granularity scanning process to enhance the cascade forest and set a sliding window to scan the original feature. We assumed that there were 400 original features, and a window size(s) of 100 features was used. For the sequence data, a 100 dimensional feature vector was generated by sliding a feature window; Three hundred and one eigenvectors were generated. The final transformed feature vector includes more features using multiple sliding windows.

The model input is the raw traffic flow, which replaces the traditional map conversion pre-processing operation and accelerates the training process. The scanning capability takes part in protocol relevance as a dimensional feature to strengthen the understanding of attribute traffic in a dataset. The traffic was successfully represented and learned using the proposed model. The input of the original cascade forest is the transformation feature of the final output of multi-granularity scanning, and the scanning features on each scale are input into the first-level forest of the cascade forest . The last layer adds and averages the C-dimensional vectors of all random forest outputs to calculate the maximum one-dimensional as the final classification decision.

IV. Experiment and Result

In this section, we evaluate the proposed efficient method using four public datasets to verify the rationality, effectiveness and adaptability cross-scenario of the model. First, we introduce the datasets used in this study. Second, we describe the experimental setting and the state-of-the-art, which are considered as the baseline. Subsequently, we discuss the experimental results for the training time overhead and few-shot set performance in detail. Finally, we test the efficiency of the model. The scale of the dataset and the amount of network traffic contained are listed in Table I.

A. Dataset Organisation

Four open datasets were used in the experiment: MalDroid2017, USTC2016, ISCX-EX, and DataCon2021. The datasets were composed of diverse scenarios with malicious, benign application, and tunnel traffic.

MalDroid2017 [16]: MalDroid2017 contained encrypted traffic from 42 families of Android malware. Traffic was captured from 426 successfully installed Android malware collected from the Google Play market and published in 2015, 2016, and 2017.

USTC2016 [17]: USTC2016 was composed of encrypted traffic generated by ten Windows malware and ten benign applications. Approximately 2 GB of network traffic was captured in the USTC2016.

ISCX-EX [18]: ISCX-EX was composed of ten classes of applications from the original ISCX2016 (traffic from non-VPN part) and additional eight commonly used applications, including Aiqiyi, Tim, Weibo, Zhihu, Youku, Baidupan, Sougou and Huya. ISCX-EX contained 21.5 GB of network traffic.

DataCon2021 [19]: DataCon2021 contained approximately 5 GB encrypted traffic data from 100 webpages. The traffic of DataCon2021 was the access request traffic of 100 common web main pages obtained using the same tunnel proxy software.

TABLE I: Description of encrypted network traffic datasets.

Dataset	Description (classes)	Scale	traffic
MalDroid [16]	42 Android malware	32 GB	73,507 traffic
USTC [17]	10 malware, 10 benign	2 GB	9,965 traffic
ISCX-EX [18]	10 popular applications	22 GB	162,515 traffic
DataCon [19]	100 webpages	5 GB	45,721 traffic

TABLE II: The baseline comparison in full datasets scenarios on different scale.

Models	MalDroid2017		USTC2016		ISCX-EX		DataCon2021	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
APPScanner [20]	72.11	73.77	57.42	56.23	89.50	86.82	73.60	73.92
FlowPrint [21]	36.35	36.82	57.73	53.28	84.13	83.51	61.17	61.23
FS-Net [12]	56.41	55.34	94.53	87.51	73.44	69.71	59.23	60.15
Datanet [22]	87.21	88.30	91.21	89.59	84.40	83.45	62.01	64.51
ACNN [23]	78.54	77.66	82.19	83.08	76.57	78.23	48.59	49.57
FC-Net [13]	43.67	43.54	92.73	91.45	81.40	82.15	58.14	59.51
LSTM [24]	91.09	90.42	93.61	93.41	88.19	88.17	75.58	75.42
ACGAN [25]	86.11	87.51	88.51	89.19	81.55	83.75	52.96	53.44
MaMPF [26]	53.88	56.71	93.19	94.01	80.07	81.32	51.19	52.34
XGBoost [27]	24.87	25.50	97.92	97.97	87.78	84.76	48.83	49.22
RF [28]	26.39	26.91	97.23	96.75	83.97	83.81	46.99	46.94
Ours	**96.66**	**96.81**	**99.64**	**99.65**	**92.22**	**96.36**	**82.54**	**82.89**

TABLE III: Few-shot scenarios encrypted traffic classification .

Models	MalDroid2017		USTC2016		ISCX-EX		DataCon2021	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
APPScanner [20]	39.42	30.35	32.56	34.53	32.32	32.14	41.11	24.01
FlowPrint [21]	18.29	16.57	88.01	87.47	39.68	36.58	41.38	41.04
FS-Net [12]	16.41	15.53	89.18	90.19	71.09	61.28	48.68	39.71
Datanet [22]	24.53	24.87	56.75	57.89	77.35	79.70	25.84	26.79
ACNN [23]	29.74	30.06	53.69	55.97	64.11	64.57	26.18	27.18
FC-Net [13]	35.09	34.99	71.48	73.74	79.84	78.88	36.10	38.07
LSTM [24]	42.51	43.20	72.38	74.62	84.25	83.45	37.95	38.18
ACGAN [25]	22.89	22.91	49.94	49.44	62.18	62.57	27.99	27.95
MaMPF [26]	32.80	34.52	90.03	89.10	82.15	83.00	39.02	39.11
XGBoost [27]	26.48	25.74	95.89	95.65	89.71	89.98	48.40	49.51
RF [28]	21.15	20.38	95.50	96.35	88.82	86.87	46.38	46.72
Ours	**79.21**	**81.57**	**99.43**	**99.43**	**90.48**	**90.53**	**67.10**	**68.33**

B. Experimental Design

1) Evaluation Metrics: We compared the Deep-Tree method with state-of-the-art methods with four standard evaluation metrics that are commonly used in classification tasks, including accuracy (Acc) and F1-score (calculated from precision and recall).

2) Experimental Setting: The experiments were performed using the following hardware and software platforms: Intel i7-9750 at 2.6GHz, 16GB RAM, NVIDIA GeForce RTX2060, Windows 10, CUDA 10.1, and PyTorch 1.0.1. The deep-learning-based baseline experiment uses a GPU as an accelerated resource; however, the proposed lightweight method requires only a CPU.

3) Baseline Evaluation: We evaluated the following well-designed experiments to validate the rationality and advancement of the proposed model. We evaluated the model using MalDroid2017 to classify large-scale encrypted traffic. Cross-dataset classification was evaluated using USTC2016, ISCX-EX, and DataCon2021. We compared the proposed model with state-of-the-art baselines for different traffic classification scenarios. The baseline methods include machine learning based [27] [28], network-fingerprint based [20] [21], neural-network-based [23] [13] [12] [25], and so on.

C. Encrypted Traffic Classification on full datasets

The assessment based on full datasets are randomly segmented according to 80% training set and 20% test set. TableII illustrates the classification results on diverse encrypted traffic. The proposed model achieves better performance compared with the baseline result because of the superiority of express learning and scalable structure. Our method has scenarios-adaptation ability and scalability of different size of dataset.

D. Encrypted Traffic Classification on few-shot scenarios

The scale of the label traffic in the few-shot sample scenario was limited. Fifty network traffic samples were randomly selected from each class. Experiment results were obtained through multiple rounds of verification.

TABLE IV: Comparison of training duration on expandable scale datasets.

Models	MalDroid2017	USTC2016	ISCX-EX	DataCon2021
ACGAN [25]	54m 34s	20m 5s	15m 42s	1h 25m
Datanet [22]	17m 7s	9m 13s	2m 52s	37m 19s
ACNN [23]	44m 54s	28m 23s	8m 45s	1h 13m
LSTM [24]	19m 11s	11m 43s	5m 38s	39m 9s
FS-Net [12]	12m 53s	7m 9s	3m 58s	24m 43s
MaMPF [26]	15m 32s	8m 15s	4m 34s	28m 2s
FC-Net [13]	49m 41s	31m 35s	27m 9s	1h 26m
Ours	**11m 12s**	**6m 27s**	**2m 4s**	**20m 44s**

The evaluation of the limited datasets was well. In few-shotIII training datasets, the effects of previous classification methods obviously decrease. Conversely, the proposed method's effect decreases slightly. Through representational learning, the proposed method can effectively use the information of the original traffic to learn the core features of encrypted traffic communication.

E. Efficiency Evaluation of the proposed Method

Table 4 IV shows a comparison of the efficiencies. On all datasets, the proposed method was superior to the existing deep-learning-based model in terms of training time and computational performance cost. Particularly, the training time overhead based on a neural network (GPU-accelerated) is three–eight times that of the proposed method (CPU only).

V. Conclusion

In this paper, we proposed an efficient solution for universal encrypted traffic classification. Feature engineering is skillfully combined with adaptive lightweight cascade forests by deploying multi-granularity scanners to the byte stream of raw traffic. We designed full training scenarios and few-shot experiments using 4 datasets from different real-world scenarios (app correlation, intrusion detection, web fingerprinting, and malware detection) to validate the rationality and robustness of the proposed method. The adaptability of our model well matches the training demand in large-scale traffic classification and representation requirement in few-shot traffic scenarios. The proposed method performs effective even when trained with limited data, which offers further deployments in few-shot scenarios.

Acknowledgement

This work was supported by Key Laboratory of Information System Security Technology (CNKLSTISS-6142111190501).

References

[1] Z. Cao, G. Xiong, Y. Zhao, Z. Li, and L. Guo, "A survey on encrypted traffic classification," in *International Conference on Applications and Techniques in Information Security*. Springer, 2014, pp. 73–81.

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

[3] S. Rezaei and X. Liu, "Deep learning for encrypted traffic classification: An overview," *IEEE communications magazine*, vol. 57, no. 5, pp. 76–81, 2019.

[4] O. Salman, I. H. Elhajj, A. Kayssi, and A. Chehab, "A review on machine learning–based approaches for internet traffic classification," *Annals of Telecommunications*, vol. 75, no. 11, pp. 673–710, 2020.

[5] N. Namdev, S. Agrawal, and S. Silkari, "Recent advancement in machine learning based internet traffic classification," *Procedia Computer Science*, vol. 60, pp. 784–791, 2015.

[6] Z. Wang, K. W. Fok, and V. L. Thing, "Machine learning for encrypted malicious traffic detection: Approaches, datasets and comparative study," *Computers & Security*, vol. 113, p. 102542, 2022.

[7] Z. Tang, Q. Wang, W. Li, H. Bao, F. Liu, and W. Wang, "Hslf: Http header sequence based lsh fingerprints for application traffic classification," in *International Conference on Computational Science*. Springer, 2021, pp. 41–54.

[8] Y. Dhote, S. Agrawal, and A. J. Deen, "A survey on feature selection techniques for internet traffic classification," in *2015 International Conference on Computational Intelligence and Communication Networks (CICN)*. IEEE, 2015, pp. 1375–1380.

[9] Y. Hu, J. Tian, and J. Ma, "A novel way to generate adversarial network traffic samples against network traffic classification," *Wireless Communications and Mobile Computing*, vol. 2021, 2021.

[10] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 445–458, 2019.

[11] T. Shapira and Y. Shavitt, "Flowpic: Encrypted internet traffic classification is as easy as image recognition," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops*. IEEE, 2019, pp. 680–687.

[12] C. Liu, L. He, G. Xiong, Z. Cao, and Z. Li, "Fs-net: A flow sequence network for encrypted traffic classification," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1171–1179.

[13] C. Xu, J. Shen, and X. Du, "A method of few-shot network intrusion detection based on meta-learning framework," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3540–3552, 2020.

[14] Z.-H. Zhou and J. Feng, "Deep forest," *arXiv preprint arXiv:1702.08835*, 2017.

[15] E. Raff, J. Barker, J. Sylvester, R. Brandon, B. Catanzaro, and C. K. Nicholas, "Malware detection by eating a whole exe," in *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[16] W. Li, X.-Y. Zhang, H. Bao, and F. Liu, "A glimpse of the whole: Detecting few-shot android malware encrypted network traffic," *Available at SSRN 3995981*, 2021.

[17] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *2017 International conference on information networking (ICOIN)*. IEEE, 2017, pp. 712–717.

[18] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," in *2017 IEEE international conference on intelligence and security informatics (ISI)*. IEEE, 2017, pp. 43–48.

[19] "Datacon2021," https://datacon.qianxin.com/opendata/openpage, 2021, accessed Feb 18, 2022.

[20] V. F. Taylor, R. Spolaor, M. Conti, and I. Martinovic, "Appscanner: Automatic fingerprinting of smartphone apps from encrypted network traffic," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2016, pp. 439–454.

[21] T. van Ede, R. Bortolameotti, A. Continella, J. Ren, D. J. Dubois, M. Lindorfer, D. Choffnes, M. van Steen, and A. Peter, "Flowprint: Semi-supervised mobile-app fingerprinting on encrypted network traffic," in *Network and Distributed System Security Symposium, NDSS 2020*. Internet Society, 2020.

[22] P. Wang, F. Ye, X. Chen, and Y. Qian, "Datanet: Deep learning based encrypted network traffic classification in sdn home gateway," *IEEE Access*, vol. 6, pp. 55 380–55 391, 2018.

[23] Y. Yang, C. Kang, G. Gou, Z. Li, and G. Xiong, "Tls/ssl encrypted traffic classification with autoencoder and convolutional neural network," in *IEEE 16th International Conference on Smart City*. IEEE, 2018, pp. 362–369.

[24] Z. Zou, J. Ge, H. Zheng, Y. Wu, C. Han, and Z. Yao, "Encrypted traffic classification with a convolutional long short-term memory neural network," in *IEEE 16th International Conference on Smart City*. IEEE, 2018, pp. 329–334.

[25] L. Vu, C. T. Bui, and Q. U. Nguyen, "A deep learning based method for handling imbalanced problem in network traffic classification," in *Proceedings of the Eighth International Symposium on Information and Communication Technology*, 2017, pp. 333–339.

[26] C. Liu, Z. Cao, G. Xiong, G. Gou, and Yiu, "Mampf: Encrypted traffic classification based on multi-attribute markov probability fingerprints," in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. IEEE, 2018, pp. 1–10.

[27] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[28] B. Yamansavascilar, M. A. Guvensan, A. G. Yavuz, and M. E. Karsligil, "Application identification via network traffic classification," in *2017 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2017, pp. 843–848.