Hindawi
Security and Communication Networks
Volume 2023, Article ID 3629831, 12 pages
https://doi.org/10.1155/2023/3629831

Research Article

A Few-Shot Malicious Encrypted Traffic Detection Approach Based on Model-Agnostic Meta-Learning

Zhiqiang Wang,[1,2] **Man Li**,[1] **Haiwen Ou**,[1] **Shufang Pang**,[1] **and Ziyan Yue**[1]

[1]*Department of Cyberspace Security, Beijing Electronic Science and Technology Institute, Beijing 102627, China*
[2]*State Information Center, Beijing 100045, China*

Correspondence should be addressed to Zhiqiang Wang; wangzq@besti.edu.cn

Received 29 October 2022; Revised 20 January 2023; Accepted 27 January 2023; Published 13 April 2023

Academic Editor: Naghmeh Moradpoor

Existing malicious encrypted traffic detection approaches need to be trained with many samples to achieve effective detection of a specified class of encrypted traffic data. With the rapid development of encryption technology, various new types of encrypted traffic are emerging and difficult to label. Therefore, it is an urgent problem to train a deep learning model using only a small number of samples to detect new classes of malicious encrypted traffic. This paper proposes a few-shot malicious encrypted traffic detection (FMETD) approach based on model-agnostic meta-learning (MAML), integrating feature selection and classification into an end-to-end framework. The FMETD approach first converts the raw traffic data into two-dimensional grayscale images. Then, FMETD trains a deep learning model (2D-CNN) using the MAML, which is to learn an optimal set of model initialization parameters for the model from a set of classification tasks consisting of grayscale images. The model with this set of parameters can detect new classes of maliciously encrypted traffic data efficiently with a few samples by a few iterations steps. The experimental results show that the FMETD approach has 99.8% accuracy for two-class classification encrypted traffic and 98.5% average accuracy for multi-classification. When the number of grayscale images of each class in the support set and validation set is reduced to 20, the accuracy of our approach to multi-class classification is 97.9% for new classes of traffic.

1. Introduction

The Internet is flooded with massive amounts of encrypted traffic. As of November 2021, the proportion of encrypted traffic in all Google products and services has exceeded 95% [1]. While encryption technology protects communication security and user privacy, it also brings many security risks. More and more criminals use encryption technology to bypass firewalls and hide their true intentions to conduct malicious attacks on the network, which seriously threatens users' privacy and the security of cyberspace. The most effective defense against cyberattacks is detecting and intercepting malicious encrypted traffic before it intrudes the secure network environment. Studying efficient and accurate anomaly detection approaches for encrypted traffic is imperative. Malicious cyberattacks based on encrypted traffic are challenging to defend accurately and effectively without corresponding decryption algorithms. The detection approaches based on decryption techniques are time-consuming, and the decryption of encrypted traffic may violate the user's private information. Therefore, a maliciously encrypted traffic detection method that can quickly detect malicious encrypted traffic without decrypting encrypted traffic is crucial for maintaining cyberspace security and user privacy security.

Existing approaches for detecting malicious encrypted traffic are divided into four categories mainly. The traditional port-based approaches are no longer reliable in increasingly complex network environments. The approaches based on deep packet inspection technology [2, 3] are not suitable for encrypted traffic. The approaches based on traditional machine learning learn specified features from a large amount of training data, which can achieve high-precision classification. Still, traditional machine learning approaches require determining statistical features manually

which would be selected, and the classification accuracy depends heavily on expert experience and dataset distribution. Also, deep learning-based approaches [4] train neural networks to extract features of encrypted traffic automatically and classify them, which has good detection performance while overcomes the shortcomings of traditional approaches and is a research hotspot in the direction of malicious encrypted traffic detection in recent years.

However, deep learning models with good performance need to be trained on a large number of accurately labeled datasets to detect malicious encrypted traffic effectively. In the context of the explosive growth of encrypted traffic, new classes of traffic are constantly emerging. The performance of deep learning-based approaches in detecting this newly emerging encrypted traffic, which is few and hard to label, will decrease dramatically. Improving the generalization ability of classification models and making them perform efficiently in the new classes of traffic with only a few samples has become an urgent problem to be solved in malicious encrypted traffic detection.

We propose a meta-learning-based encryption traffic detection approach in response to the above problems. Our approach can effectively make up for the defect that the performance of existing methods deteriorates when there are few labeled samples, and at the same time, this method can effectively identify new classes of malicious encrypted traffic. We aim to achieve few-shot malicious encrypted traffic detection, so we choose the model-agnostic meta-learning algorithm to train the deep learning model. The model-agnostic meta-learning (MAML) algorithm is an optimization-based meta-learning algorithm, and it can be directly applied to any learning problem and model that is trained with a gradient descent procedure. MAML is a method that can quickly adapt to new tasks when only a small number of labeled samples are available, so we combine MAML with convolutional neural network to achieve fast detection of emerging encrypted malicious traffic.

Representation learning is a method of trying to improve the defects of traditional machine learning, which automatically learns features from raw data. CNN was used as a representation learning technique in our experiments. CNN is mainly used to identify two-dimensional graphics with displacement, scaling, and other forms of distortion invariance. So, we train the 2D-CNN using MAML on a set of classification tasks consisting of grayscale images. We convert raw traffic to grayscale images and use 2D-CNN to learn traffic features for classification. It can fully extract the spatial features of the raw traffic and directly learn the deep abstract features of the traffic data. Visual processing solves the difficulty of extracting original traffic features and transforms the difficulty of extracting original traffic features into image feature extraction, making the training results more accurate. Our approach splits the raw traffic data into sessions with a uniform length of 784 bytes in the data preprocessing phase and converts each session into a grayscale image of 28 bytes*28 bytes. A session is defined as bidirectional flows, in which all packets have the same 5-tuple. The 5-tuple is source IP, source port, destination IP, destination port, and transport-level protocol. The few-shot malicious encrypted traffic detection (FMETD) approach uses the model-agnostic meta-learning (MAML) algorithm to train a deep learning model on various classification tasks so that this model can learn a good initialization parameter for the deep learning model. This model consists of a meta-training phase and a meta-testing phase. In the meta-training phase, our approach will learn to adapt to the new class by training in few-shot class-adaptive malicious traffic detection tasks. The meta-testing phase will use the pretrained model to adapt the new class through a few iterative steps. The model needs only a small amount of labeled data to quickly adapt to new classes of malicious encrypted traffic after a few gradient descent steps. Finally, comparative experiments on public datasets demonstrate the superiority of this approach.

The main contributions of this paper are as follows:

(1) A maliciously encrypted traffic detection approach based on model-agnostic meta-learning (MAML), termed FMETD, is proposed, which provides a new idea for solving the encrypted traffic classification problem.

(2) We apply the end-to-end approach to few-shot malicious encrypted traffic detection for the first time. In the FMETD approach, we converted the traffic data to grayscale images in the data preprocessing process as the input of a two-dimensional convolutional neural network model. Then, we use MAML to train CNN with only a few training samples to learn an optimal set of initialization parameters for the model. The end-to-end approach can detect and classify new classes of malicious encrypted traffic fastly, improving the accuracy of few-shot malicious encrypted traffic detection significantly.

(3) We designed and conducted comparison experiments to verify the effectiveness of FMETD on two public datasets. When the number of grayscale images in each class of data reduced to 20, the detection accuracy of this approach for new class traffic is 97.9%, which was significantly higher than that of other existing approaches. The experiment results prove that the FMETD approach has obvious advantages in detecting new class malicious encrypted traffic with a few labeled samples.

The rest of this paper is organized as follows. In Section 2, we review related work of the malicious encrypted traffic detection. Section 3 introduces the data preprocessing operation, which converts raw traffic data into grayscale images. In Section 4, the algorithmic details of the FMETD

approach are described. In Section 5, the experimental environment and dataset are introduced. Section 6 introduces the traditional and few-shot experiment based on the FMETD approach and gives the results and analysis of the experiment. Finally, this paper's conclusion and future work are given in Section 7.

2. Related Work

This section mainly summarizes the existing malicious encrypted traffic detection and classification approaches and research progress related to meta-learning.

2.1. Malicious Encrypted Traffic Detection. In the early research, researchers mostly used rule-based approaches to detect malicious encrypted traffic. Such approaches need to choose the distinguishing factors of various traffic data automatically from gigantic data as classification features. The process is time-consuming and labor-intensive. Due to the complexity of the network environment, the classification efficiency and effect of these approaches are unsatisfactory. Existing malicious encrypted traffic detection approaches are mostly based on traditional machine learning and deep learning. The machine learning-based approaches use encrypted traffic statistical features to build models and summarize traffic features to distinguish traffic categories. Traditional machine learning algorithms include C4.5 decision tree (DT) [5], Naive Bayes (NB) [6], K-means [7], support vector machine (SVM) [8], and random forest (RF) [9]. Approaches based on traditional machine learning [10–12] make up for rule-based approaches' deficiencies and have greatly improved classification accuracy. However, these approaches still need to determine statistical features manually which would be selected, and the classification effect depends heavily on expert experience and the distribution of the dataset. Approaches based on deep learning [13–15] train neural networks on massive amounts of data to automatically learn the classification features of encrypted traffic, which can achieve end-to-end encrypted traffic classification. In other words, these approaches can get the classification results through raw traffic.

Wang et al. [16] proposed an end-to-end encrypted traffic classification approach using one-dimensional convolutional neural networks, which integrated feature extraction, feature selection, and classifier into a unified end-to-end framework, intending to automatically learn the nonlinear relationship between the original input and the expected output. In the same year, Wang et al. [17] proposed an approach to classify encrypted network traffic using two-dimensional convolutional neural networks, which converted the first 784 bytes of traffic data into the two-dimensional format IDX files to implement encrypted traffic classification. They also found that sessions with all layers were the best types of traffic representation. Aceto et al. [18] proposed a scheme for classifying encrypted network traffic on mobile using deep learning methods and provided a comparative analysis of the classification effectiveness of various types of deep learning models. Lotfollahi et al. [19] proposed a deep learning-based approach which integrates both feature extraction and classification phases into one system. The system can identify encrypted traffic and also distinguishes between VPN and non-VPN network traffic. Aceto et al. [20] also proposed a novel multi-modal DL framework for encrypted traffic classification, which overcomes performance limitations of existing (myopic) single-modality DL-based TC proposals and supports the challenging mobile scenario. Zeng et al. [21] sliced encrypted traffic data into 30 bytes × 30 bytes two-dimensional format IDX files and used three network structures of CNN, LSTM, and SAE to identify and classify encrypted traffic. Liu et al. [22] applied recurrent neural networks to the problem of encrypted network traffic classification and proposed an end-to-end classification model using stacked autoencoders to deeply mine the potential time features of the traffic data, effectively enhancing the feature validity. Lim et al. [23] extended the dataset provided by UPC's Broadband Communication Research Group to classify network traffic using convolutional neural network (CNN) and residual network (Resnet) as deep learning models. Hu et al. [24] proposed the OpenCBD model based on convolutional neural networks and transformer encoder to identify unknown encrypted traffic and classify known encrypted traffic.

2.2. Few-Shot Malicious Encrypted Traffic Detection. Many approaches based on deep learning depend heavily on a large amount of correctly labeled data for training to obtain good malicious traffic detection results. So, the performance of the existing approaches will be degraded dramatically by the increasingly complex network environment and the emergence of various classes of new malicious traffic. Nowadays, designing and implementing few-shot learning algorithms for malicious encrypted traffic detection is a hot and challenging research field. Sun et al. [25] represented the structure of network traffic data by constructing KNN graphs. The proposed encrypted traffic classification model consisted of a graph convolutional network module and an autoencoder module to learn the traffic structure feature representation of the traffic data. Rezaei and Liu [26] proposed a semisupervised learning approach. The approach first pretrains a model on a training set containing a large number of samples and then transfers the pretrained model to a new architecture and retrains it with a small amount of data. With only a few labeled traffic data, the pretrained model can quickly solve other new encrypted traffic classification problems.

2.3. Meta-Learning. The success of deep learning relied on multiple gradient descent to optimize weights and update internal parameters. Gradient descent-based optimization algorithms will fail on few-shot learning. The reason is that the few parameter update iterations of few shot cannot make the network learn a feature representation with strong generalization ability, resulting in poor classification effect. Meta-learning [27–30] is a high-level cross-task learning strategy that can find an optimal set of initialization

parameters for deep learning models with only a few iterations on a few labeled training samples. It is an effective way to achieve few-shot learning.

There are three main types of meta-learningmodel-based, metric-based, and optimization-based. Most algorithms with more applications and better experimental accuracy come from optimization-based meta-learning algorithms. In optimization-based meta-learning, Finn et al. [31] proposed a model-agnostic meta-learning (MAML) algorithm, which finds a good set of initialization parameters for the basic learner through a cross-task training strategy. The basic learner with this initialization parameter can quickly adapt to a new task using only a few support samples. MAML can be combined with any models amenable to gradient-based training for classification, regression, and reinforcement learning. Antoniou et al. [32] analyzed the advantages and disadvantages of the MAML framework and proposed an improved algorithm MAML++. They took multiple methods to reduce the inner loop hyperparameter sensitivity, improved the generalization error, and stabilized and sped up the MAML. Ravi and Larochelle [33] proposed an LSTM-based meta-learning model that uses LSTM states to represent neural network parameters and learn neural network parameter update rules by training the LSTM. This meta-learning model finds a successful parameter update mechanism with optimal initialization parameters for new classification tasks with only a small number of samples. Nichol et al. [34] extended Finn et al.'s research by proposing a new meta-learning algorithm Reptile based on FOMAML. Reptile is the same as MAML to find the optimal initialization parameters for the neural network. Still, it differs from MAML in that its gradient update approach is changed from second order to first order, saving computational costs and speeding up learning.

A summary of existing work in the literature dealing with malicious traffic detection is presented in Table 1.

In this paper, we propose a malicious encrypted traffic detection approach with MAML. We use the excellent few-shot learning ability of MAML to detect and classify new classes of maliciously encrypted traffic efficiently with a few labeled training samples.

3. Data Preprocessing

In this section, we describe the four steps of data preprocessing, which converts raw traffic to grayscale images.

The data preprocessing operation converts the raw traffic data in pcap form into two-dimensional grayscale image data. The data preprocessing process is shown in Figure 1, including the steps of traffic data slice, data cleaning, data trimming, and image generation.

(1) *Traffic Data Slice*. This step splits the continuous raw traffic data in packets into discrete traffic units with all protocol layer data by session. It is the best type of traffic representation in deep learning-based malicious traffic detection evaluated by Wang et al. [17].

(2) *Data Cleaning*. We first randomize each session's IP address and MAC address in this step. The IP address and MAC address in data link layer and IP layer are ineffective features to distinguish different kinds of traffic. If the classification model learns these features, it will overfit. Therefore, this paper obfuscates MAC addresses and IP addresses using the random substitution operation. The second work of this step removes some duplicate and empty data generated by traffic data slice operation, which can interfere with the model's training.

(3) *Data Trimming*.Sessions from the different networks have different lengths, and the structure of the classification model is fixed, so the session length needs to be uniform. This step modifies the length of all sessions to 784 bytes uniformly. In order to compare experimental results with related work, we refer to Wang et al.'s [17] data preprocessing step and select 784 bytes as the fixed length of the session. The excess part is intercepted if the length is more significant than 784 bytes. If the length is less than 784 bytes, 0 is added at the end.

(4) *Image Generation*.The unified length data are converted into 28 bytes × 28 bytes two-dimensional grayscale images. Images in png format are considered the output of this step. One byte in the raw traffic data represents a one-pixel value in the grayscale image, for example, 0x00 for black and 0xff for white.

4. Methodology

In this section, we introduce the algorithm of FMETD approach in detail based on MAML, which detects encrypted malicious traffic with only a few labeled samples.

In this paper, we use MAML to train CNN to detect and classify malicious encrypted traffic with only a few samples. As shown in Figure 2, this approach consists of a meta-learner and a basic learner. The meta-learner is a model-agnostic meta-learning (MAML) algorithm, and the basic learner is deep learning model CNN. The meta-learner consists of a meta-training phase and a meta-testing phase. Few-shot malicious encrypted traffic detection aims to train a basic learner that can adapt to new malicious traffic with only a few training data and parameter iterations. To achieve this, an optimal set of initialization parameters is learned during the meta-training phase by training the basic learner through a few gradient descent steps on a set of tasks. In the meta-testing phase, new malicious encrypted traffic not present in the training data is detected and classified by the trained basic learner.

4.1. MAML. In this paper, we use MAML to train CNN with only a few training samples to learn an optimal set of initialization parameters for the model. MAML is an optimization-based meta-learning algorithm, and it can be directly applied to any learning problem and model that is trained with a gradient descent procedure. MAML is a high-level cross-task learning strategy that can find an optimal set of initialization parameters for deep learning model so that the

Table 1: Overview of research methods (first group adopted ML, second one employed DL, and third one is few-shot learning).

Paper	Recognition methods	Classifier	Input data	Research conclusion
[5]	Machine learning	C4.5 decision tree	HTTP traffic	TCP flow:98.16% UDP flow:99.65%
[9]	Machine learning	RF	Packet header information and payload	Acc:99.13% Dr:99.26%
[10]	Machine learning	SVC, K-means	Statistics of PS and IAT	Acc ≥ 90%
[11]	Machine learning	Soft/hard combination of traffic classifiers	Statistics of PS	+9.5% rec. with respect to best classifier (49/45 Android/iOS apps)
[12]	Machine learning	WF methods	First 64 TCP PS	88% best acc. (1595 Android apps)
[16]	Deep learning	1D-CNN	First 784 bytes of raw traffic	Two-class acc:99.5% Multi-class acc:99.41%
[17]	Deep learning	2D-CNN	First 784 bytes ALL layer + session two-dimensional image	Four-class acc Multi-class acc:99.17%
[18]	Deep learning	SAE, LSTM 1D-CNN, 2D-CNN Hybrid LSTM + 2D-CNN	ALL/L7 layers 4–6 fields Packet directions	Comprehensive evaluation 86%/83% acc. (49/45 Android/iOS apps)
[19]	Deep learning	1D-CNN, SAE	Tor's traffic Pcap file	Recall = 94%
[20]	Deep learning	Multi-modal DL (1D-CNN, LSTM/GRU)	Heterogeneous input data, session	iOS apps acc = 82.99%
[21]	Deep learning	Deep-full-range	30 bytes × 30 bytes two-dimensional image	Identify and classify encrypted traffic
[22]	Deep learning	FS-Net	Packet length sequences	99.14% TPR, 0.05% FPR, and 0.9906 FTF
[23]	Deep learning	CNN and Resnet	Two-dimensional image	Classify network traffic without the intervention of the network operator
[24]	Few-shot learning	OpenCBD	ALL layer + session two-dimensional image	9-class classification is over 72%
[25]	Few-shot learning	GCN	KNN graphs	Obtain higher classification performance with only very few labeled data
[26]	Few-shot learning	1-D CNN	Raw traffic data	Use only 20 samples per class accuracy:98%

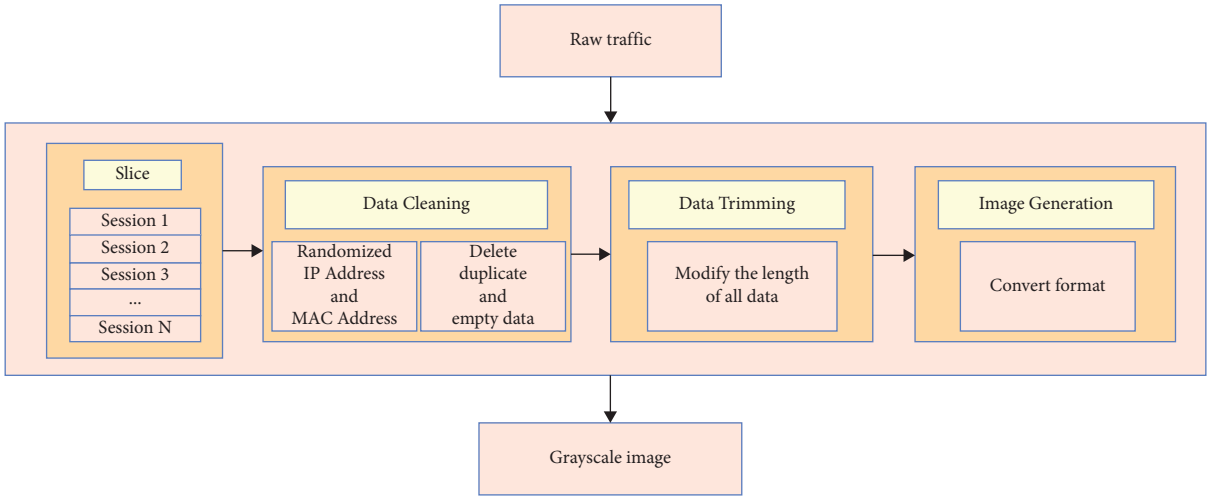

Figure 1: Data preprocessing process.

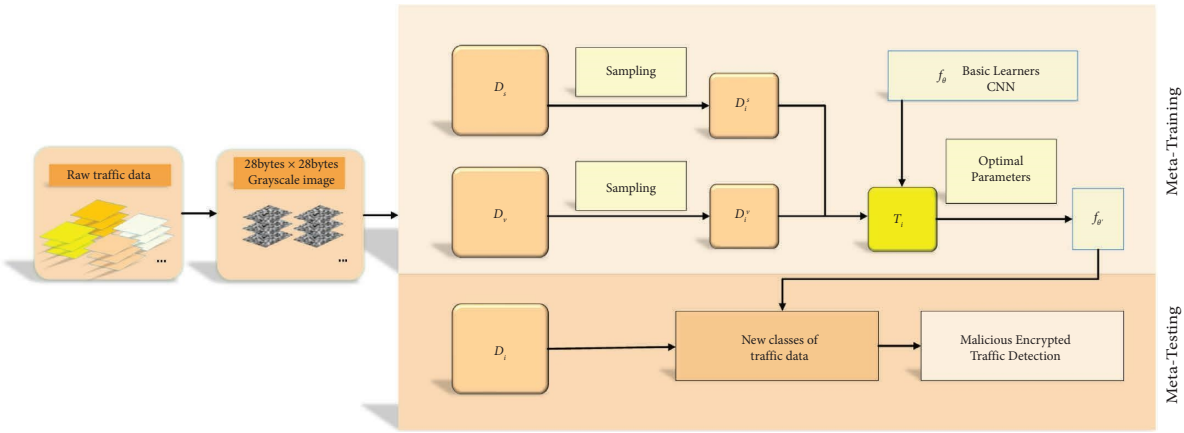

Figure 2: Overview of the FMETD approach.

model with this initialization can quickly adapt to a new task only using a few samples. During meta-training, multiple tasks are sampled from the dataset $\mathscr{D}$, and the model is trained with K samples from these tasks to obtain its parameters θ. Then, the model is tested on the new samples from all tasks, the test error is computed, and parameters θ are updated by gradient descent. The parameter θ obtained at this point is highly sensitive to the new task, and it is used as the initialization of the model, which can make the model adapt to the new task quickly with only a few samples.

4.2. Task Definition. After the data preprocessing step, the raw traffic data in the dataset $\mathscr{D}$ consist of two partsthe image x in png format and label data a. The dataset $\mathscr{D}$ has H types of encrypted traffic. We divide the dataset $\mathscr{D}$ into three partsthe support set $\mathscr{D}_j$, the validation set $\mathscr{D}_v$, and the test set $\mathscr{D}_t$. The classes of these three parts of data $\mathscr{C}_s = \{1, \ldots, H_1\}$, $\mathscr{C}_v = \{H_1 + 1, \ldots, H_2\}$, and $\mathscr{C}_t = \{H_2 + 1, \ldots, H\}$ are disjoint. Our approach uses MAML algorithm to train CNN, and we will aim to learn a classification model on $\mathscr{D}_j$ and $\mathscr{D}_v$, which can make rapid progress on new tasks drawn from $\mathscr{T}$, without overfitting. During meta-training, our approach trains the classification model with the samples from $\mathscr{D}_j$, and then the model is tested on the new classes of samples from $\mathscr{D}_v$ (new classes of traffic) and the loss is fed back to update the parameters θ of the model by one or more steps of gradient descent. So, we can find model parameters that are highly sensitive to the new classes of traffic, and it is used as the initialization of the model, which can make the model adapt to the new classes of traffic quickly with only a few samples. Our approach trains the model to learn the features of traffic which are broadly applicable to all tasks from dataset $\mathscr{D}$ rather than one task. In the meta-test phase, the model initialized with parameter θ can quickly adapt to new malicious encrypted traffic in the test set $\mathscr{D}_t$.

Before the meta-training begins, we first explain the composition of the training task. Our approach updates the initialization parameters over a set of M-way-K-shot classification tasks $\mathscr{T} = \{\mathscr{T}_i = \{\mathscr{D}_i^j, \mathscr{D}_i^v\}\}_{\mathscr{N}}$ to quickly adapt to new tasks with only a small number of training samples. We define a task as $\mathscr{T}_i = \{\mathscr{D}_i^j, \mathscr{D}_i^v\}$, and each task $\mathscr{T}_i$ consists of a support set $\mathscr{D}_i^j$ and a validation set $\mathscr{D}_i^v$. In the meta-training phase, we randomly sample M classes $\mathscr{C}^{M1}$ from $\mathscr{C}_s$, and then we sample K labeled two-dimensional grayscale images from each class in $\mathscr{C}^{M1}$ from $\mathscr{D}_j$ to construct the support set $\mathscr{D}_i^j = \{(x_1, a_1), (x_2, a_2), \ldots, (x_{\mathscr{K}}, a_{\mathscr{K}})\}$ of task $\mathscr{T}_i$. Then, we randomly sample M classes $\mathscr{C}^{M2}$ from $\mathscr{C}_v$, and then we sample K labeled two-dimensional grayscale images from each class in $\mathscr{C}^{M2}$ from $\mathscr{D}_v$ to construct the validation set $\mathscr{D}_i^v = \{(x_1, a_1), (x_2, a_2), \ldots, (x_{\mathscr{K}}, a_{\mathscr{K}})\}$ of task $\mathscr{T}_i$. In the meta-training phase, we train the classification model with the support set $\mathscr{D}_i^j$.

4.3. Meta-Training. In the meta-training phase, the meta-learner updates the initialization parameters through the inner loop and the outer loop. In the inner loop, the meta-learner optimizes the weights of the neural network on the support set $\mathscr{D}_i^j$ of $\mathscr{T}_i$ by means of the gradient descent algorithm. In the outer loop, the loss values of each training task on the validation set are summed while the initialization parameters of the basic learner are updated by the gradient descent algorithm. We represent the basic learner algorithm with a parametric function f_θ with parameter θ. θ is actually the parameter of the two-dimensional convolutional neural network in our approach. The algorithm for the meta-training phase is outlined in Algorithm 1.

In the inner loop of the meta-training phase, the basic learner updates the local parameters θ by one or more steps of gradient descent on each task $\mathscr{T}_i$. First, we compute the loss function of the basic learner $\mathscr{T}_i$ according to the following formula:

$$\mathscr{L}_{\mathscr{T}_i}(f_\theta) = \sum_{(x,a)\in\mathscr{D}_i^j} L(f_\theta(x), a). \tag{1}$$

We use the cross-entropy error $L(f_\theta(x), a)$ between the predicted label $f_\theta(x)$ of the basic learner and the true label a as the loss function in our approach. The cross-entropy loss function of a single sample in the case of two-class classification is defined as

$$L(f_\theta(x), a) = -[x \bullet \log(p) + (1 - x) \bullet \log(1 - p)], \quad (2)$$

where p is the probability that the sample x is predicted to be positive by the basic learner.

When the basic learner adapts to a new task $\mathcal{T}_i$, the parameter θ is updated to θ_i'. When using one-step gradient descent, the parameter θ varies as follows:

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta), \quad (3)$$

where α is the step size of local parameter update, which is a fixed hyperparameter.

The purpose of the outer loop is to compute the loss value on the validation set for all tasks. Further summing all loss values utilizes a gradient descent algorithm to update the parameter θ of the base learner. For each task $\mathcal{T}_i$, we define a loss function on the validation set, where $L(f_{\theta_i}(x), a)$ is also the cross-entropy loss function:

$$\mathcal{L}_{\mathcal{T}_i}(f_{\theta_i}) = \sum_{(x,a) \in \mathcal{D}_i^v} L(f_{\theta_i'}(x), a). \quad (4)$$

Then, we sum the loss values on the validation set for all tasks and update the parameter θ using one or more of gradient descent steps. β is the meta-step size of global parameter update, which is a fixed hyperparameter.

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{i=1}^{N} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}). \quad (5)$$

4.4. Meta-Testing. After the meta-training phase, our approach finds an optimal initialization parameter θ for the basic learner. The test set $\mathcal{D}_t$ is composed of new classes of malicious encrypted traffic that do not appear in the support set and validation set. During the meta-test phase, the basic learner initialized with parameter θ can quickly adapt to new malicious encrypted traffic in the test set.

4.5. Architecture of CNN. The deep learning model of our approach is CNN, which has four modules. The convolutional layer of each module consists of 3×3 convolutions and 48 filters, followed by a nonlinear activation function ReLU and a batch normalization operation. 2×2 max-pooling operation is used in the pooling layer of our model after the convolutional layer. The input of the first module includes raw traffic data from 28 bytes $\times$ 28 bytes two-dimensional grayscale images. A softmax function follows the last layer to output the probability of each class.

5. Experiment Setup

5.1. Experiment Environment. Pytorch [35] was used as deep learning framework on a server with Ubuntu 18.04.5 64 bit OS. A server with CPU model AMD Ryzen 9 3950X 16-Core Processor and GPU NVIDIA GeForce RTX 3090 was used in the experiments. The programming language was Python 3.8.

5.2. Dataset. To evaluate the capability of the FMETD approach in malicious encrypted traffic detection, we applied our approach to two public datasets, ISCXVPN2016 and CICAndMal2017. ISCXVPN2016 dataset has twelve classes of normal encrypted raw traffic data, and the malicious traffic data of CICAndMal2017 dataset came from 42 malware families.

(1) *ISCXVPN2016 Dataset*. Lashkari et al. [36] released the ISCXVPN2016 dataset in 2016. Fourteen classes of traffic data were in the ISCXVPN2016 dataset, including seven regular encrypted traffic and seven protocol encapsulated traffic. The dataset has two formatstraffic characteristics and raw traffic (packets in pcap form). Our approach selected the raw traffic for experiments. The ISCXVPN2016 dataset has twelve classes of accurately labeled raw traffic data, including six classes of conventional encrypted traffic data and six classes of protocol encapsulated traffic. The specific content of the dataset is shown in Table 2.

(3) *CICAndMal2017 Dataset*.The CICAndMal2017 dataset was published by Lashkari et al. [37] in 2018. The authors collected malicious and normal encrypted network traffic data by running more than 5000 applications (426 malware and 5065 benign software) on real devices. The benign software among them was from the Google Play Store from 2015 to 2017. The malware was divided into four categoriesadware, ransomware, scareware, and SMS malware, from 42 malware families. The traffic data of malware in the dataset were used in this experiment.

5.3. Evaluation Metrics. The FMETD approach is evaluated using the following four evaluation metricsaccuracy, precision, recall, and F1 score.

A confusion matrix is a specific matrix used to present a visualization of algorithm performance. It is very easy to

Require: distribution over tasks
Input: α, β Step size hyperparameter
(1) Randomly initialize the parameter θ for the basic learner f_θ
(2) **while** not done **do**
(3) Sample batch of tasks $\mathcal{T}_i \sim \mathcal{T}$
for all $\mathcal{T}_i$ **do**
(4) Compute the loss function $\mathcal{L}_{\mathcal{T}_i}(f_\theta)$ on the support set $\mathcal{D}_i^s$
(5) Update adapted parameters by gradient descent algorithm $\theta_i' = \theta - \alpha\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
(6) **end for**
(7) Compute the loss value on the validation set $\mathcal{D}_i^v$ for all tasks $\sum_{i=1}^{N} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$
(8) Update initialization parameters $\theta \leftarrow \theta - \beta\nabla_\theta \sum_{i=1}^{N} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$
(9) **end while**

Algorithm 1: MAML for few-shot malicious encrypted traffic detection.

show whether multiple classes are confounded (one class is predicted to be another). In Table 3, each column represents the predicted value and each row represents the actual value. Confusion matrix reports the number of false positives, false negatives, true positives, and true negatives.

TN is the number of samples for which the actual negative prediction is also negative. TP is the number of samples for which the actual positive prediction is also positive. FN is the number of samples for which the actual positive prediction is negative. FP is the number of samples for which the actual negative prediction is positive.

$$\begin{aligned} \text{Accuracy} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \\ \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ F_1 &= 2\frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \end{aligned} \tag{6}$$

The F1 score represents the harmonic mean of precision and recall, which can indicate the classification performance of a deep learning model relatively accurately. It is expressed in the range of 0 to 1, where the best value is 1. We compute the recall, precision, and F1 score of all application labels and then finally average them to get a single overall F1 score measurement for performance.

5.4. Baseline. The algorithms we chose to compare with our approach were support vector machine (SVM), long shot-term memory (LSTM), and convolutional neural network (CNN). Based on the analytical study in the related work section, we can know that these approaches are the most effective cryptographic malicious traffic detection algorithms available today. This method unified the session length to 784 bytes in the data preprocessing part, referring to the method proposed by Wang et al. [17]. They used CNN to classify the encrypted traffic after data preprocessing. So, comparing our approach with these methods can better reflect the effectiveness and superiority of our approach.

Table 2: ISCXVPN2016 dataset.

Label	Content
E-mail	E-mail, Gmail (SMPT, POP3, IMAP)
Chat	ICQ, AIM, Skype, Facebook, Hangouts
Streaming	Vimeo, YouTube, Netflix, Spotify
File transfer	Skype, FTPS, SFTP
VoIP	Facebook, Skype, Hangouts, VoipBuster
P2P	uTorrent, Bittorrent

6. Experiment Result and Analysis

In this section, we compared our approach with existing state-of-the-art approaches, verifying whether the FMETD approach outperforms existing state-of-the-art approaches for malicious encrypted traffic detection.

We set up two experiments, one to verify the FMETD approach's capacity to detect malicious encrypted network traffic, and the other to verify its ability to detect new class malicious encrypted network traffic using only a few training samples. The hyperparameters in the experiment were set as followsthe inner loop step size was 0.01, the outer loop step size was fixed to 0.001, the batch size was 5, and the basic learner was trained for 100 epochs, each epoch consisting of 600 internal iterations. The model obtained in each round of training was saved. After the training phase, the five best-performing models on the validation set were selected for testing on the test set, and the average of the five models was used as the final experimental result.

6.1. Malicious Encrypted Traffic Detection. First, experiments were performed under traditional conditions (the classes $\mathcal{C}_s$ of test data were the same as the support data, and the training set contained a large number of samples) to verify the capacity of FMETD approach to detect and classify malicious encrypted traffic. In the two-class classification experiment, because we needed to detect malicious encrypted traffic from normal traffic, we ran the experiments on 1-way and 5-shot settings on the dataset to make sure all samples in one task were from the same malicious class.

When we ran the two-class classification experiment, we integrated ISCXVPN2016 and CICAndMal2017 to construct a raw traffic dataset in which normal encrypted traffic came from ISCXVPN2016 and malicious encrypted traffic came from CICAndMal2017. Then, we ran the multi-class experiments on the CICAndMal2017 dataset. The experimental results of two-class classification are shown in Figure 3 and Table 4, and the multi-class classification is shown in Figure 4.

The experimental results showed that when the training set has a large number of samples, the two-class classification accuracy of this approach reaches 99.8%, which was higher than the 98.7% classification accuracy of the CNN. FMETD approach has a multi-classification accuracy of 98.5% on the CICAndMal2017 dataset for malicious encrypted traffic. The experimental results demonstrated that our approach has the same good detection capability as the existing state-of-the-art approaches in traditional malicious encrypted traffic detection.

6.2. Few-Shot Malicious Encrypted Traffic Detection. In addition to enabling the detection of malicious encrypted traffic when the training set contains a large number of labeled samples, our approach can quickly detect new class malicious encrypted traffic with only a few samples. We compared the FMETD approach with the state-of-the-art approaches, which are RF-based, LSTM-based, and CNN-based.

First, we ran the experiments on the 5-way and 1/5-shot settings on the ISCXVPN2016 dataset. For the 20 classes of traffic data in the ISCXVPN2016 dataset, we randomly selected 12 classes of them as the support set, 4 classes as the validation set, and the remaining 4 classes as the test set, each class containing 20 grayscale images. This experiment validated the capacity of the FMETD approach to detect new encrypted traffic with only small training samples. In all comparison experiments, the dataset was divided into a training set and a test set. The training set consisted of 16 classes of traffic data, and the test set had the remaining 4 classes of traffic data. Every class also has 20 grayscale images. Due to the number of samples in the training set being too small, we conducted comparison experiments until we got the optimal classification accuracy as the final result. The experimental results are shown in Table 5.

We ran the experiments on the 5-way and 5/10-shot settings on the CICAndMal2017 dataset. For the 42 classes of malicious data in the CICAndMal2017 dataset, we selected 23 as the support set, 8 as the validation set, and the remaining 11 as the test set, each class containing 33 grayscale images. In each comparison experiment, the dataset was divided into a training set (31 classes) and a test set (11 classes). Figure 5 shows the trend of the accuracy rate as the number of training rounds increases when the experiment setting is 5-way-5-shot. The accuracy of FMETD and other comparison approaches on CICAndMal2017 is shown in Table 6.

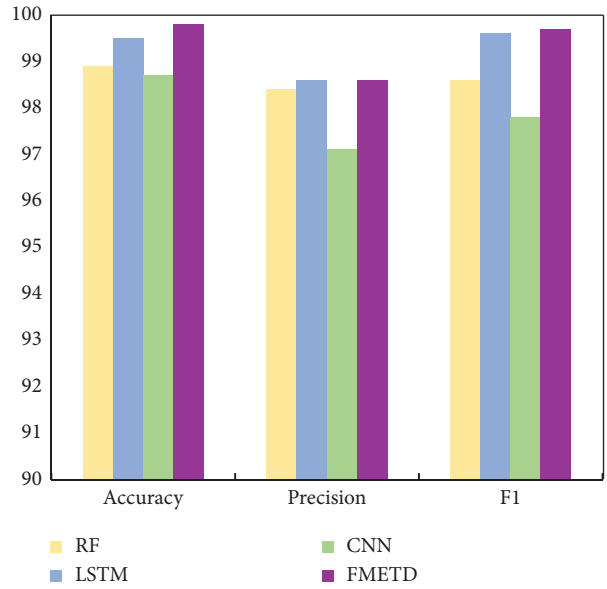

Figure 3: Two-class classification.

Table 3: Confusion matrix.

Confusion matrix		Actual class	
		Positive	Negative
Predictive class	Positive	TP	FP
	Negative	FN	TN

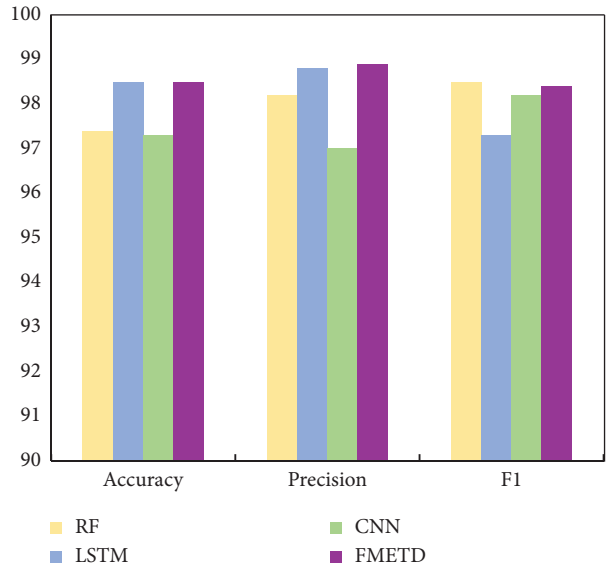

Figure 4: Multi-class classification.

As shown in Figure 5, the accuracy of our approach on the support set and validation set is all over 90% just after two epochs. In the fifth epoch, the accuracy of FMETD on the validation set is already at the best level. The FMETD

Table 4: Confusion matrix of two-class classification.

Confusion matrix		Actual class	
		Positive	Negative
Predictive class	Positive	119	0
	Negative	1	120

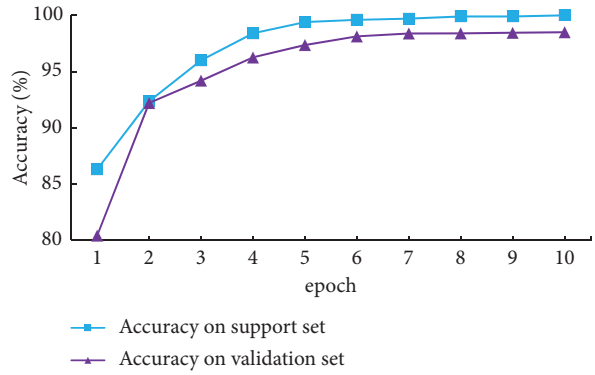

Figure 5: The accuracy on CICAndMal2017 and 5-way-5-shot setting.

Table 5: Few-shot malicious encrypted traffic detection on ISCXVPN2016.

ISCXVPN2016		Accuracy	
		5-way-1-shot (%)	5-way-5-shot (%)
1	**FMETD**	**94.3**	**97.9**
2	RF	74.9	
3	LSTM	84.2	
4	CNN	75.6	

The bold values given in above Table emphasize the higher accuracy of our method.

Table 6: Few-shot malicious encrypted traffic detection on CICAndMal2017.

CICAndMal2017		Accuracy	
		5-way-5-shot (%)	5-way-10-shot (%)
1	**FMETD**	**95**	**97.8**
2	RF	65.7	
3	LSTM	76.2	
4	CNN	67.3	

The bold values given in above Table emphasize the higher accuracy of our method.

approach can quickly adapt to new classes of encrypted traffic, which proves the high efficiency of this approach.

Table 6 shows that none of the existing approaches have an accuracy rate of 90%. What we know is that both traditional machine learning-based and deep learning-based approaches do not detect new encrypted malicious traffic well with only a small number of samples. The multi-class classification accuracy of our approach was 94.3% on the 5-way-1-shot settings (each task contains five classes of malicious encrypted traffic, with one sample in each class) on the ISCXVPN2016 dataset. When the setting was changed to 5-way-5-shot, the accuracy of our approach increased to 97.9%. The classification accuracy of the FMETD approach was 95% when tested on the CICAndMal2017 dataset in the 5-way-5-shot tasks, and in the 5-way-10-shot tasks, the accuracy reached 97.8%.

Experimentally, it can be found that when there are only a few samples that can be used to train the classification model, the detection accuracy of the FMETD approach for new classes of malicious encrypted traffic is much higher than that of the existing state-of-the-art approaches. Many existing approaches need to be trained on a large amount of traffic data to accurately detect specific classes of encrypted network traffic. However, in real network environments, the classes of malicious encrypted traffic are not static and few are labeled. So, the performance of existing approaches drops dramatically on these new classes of traffic. Our approach uses MAML to learn an optimal set of initialization parameters for CNN on a set of classification tasks. The model with these initialization parameters can quickly adapt to new class data with few training samples. The experimental results show that the FMETD approach can effectively make up for the shortcomings of the existing approaches and significantly improve the classification effect of the new class of traffic data under a low label rate of training samples.

7. Conclusion

The effectiveness of the existing malicious encrypted traffic detection approaches is deteriorated due to the rapid development of the Internet and the wide application of encryption technology. Existing approaches have low detection accuracy and poor generalization when only a few labeled traffic is available. In this paper, we propose a few-shot malicious encrypted traffic detection (FMETD) approach based on MAML, which integrates feature selection and classification into an end-to-end framework. In our approach, we take the two-dimensional grayscale images converted from the raw traffic data as the input of the deep learning model. We train the CNN on the classification tasks consisting of support set and validation set using MAML for efficient malicious encrypted traffic detection. Experimental results show that the FMETD approach can quickly classify malicious encrypted traffic with high multi-class classification accuracy. Our approach achieves efficient and accurate detection of new classes of malicious encrypted traffic which is invisible in the training phase when there are only a few kinds of labeled traffic, demonstrating the strong generalization of this approach.

This paper converts the raw traffic data into grayscale images for feature extraction and malicious traffic classification. Only the spatial features of the traffic data are used in

this process. The next step of the work is to investigate how to add temporal features to the detection to achieve a more efficient detection of malicious encrypted traffic. In future work, we will research how to add these temporal features in our few-shot learning approach, using recurrent neural network (RNN) to replace the CNN model, following, e.g., [38].

Acronyms

FMETD: Few-shot malicious encrypted traffic detection
MAML: Model-agnostic meta-learning
DT: Decision tree
NB: Naive Bayes
SVM: Support vector machine
RF: Random forest
CNN: Convolutional neural network
LSTM: Long short-term memory
Resnet: Residual network
KNN: K-nearest neighbor
FOMAML: First-order MAML
IP: Intellectual property
MAC: Media access control
SMS: Short message service
RNN: Recurrent neural network
Tree-RNN: Tree structural recurrent neural network.

Data Availability

The data used to support the findings of this study can be obtained from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Man Li and Zhiqiang Wang contributed equally to this work.

Acknowledgments

This study was supported by the National Key Research and Development Plan of China (grant no. 2018YFB0803401), Special Fund on First-Class Discipline Construction Project of Beijing Electronic Science and Technology Institute (grant no. 3201012), and China Postdoctoral Science Foundation (grant no. 2019M650606).

References

[1] Https encryption on the web, "Https encryption on the web," 2022, https://transparencyreport.google.com/https/overview.

[2] G. De La Torre Parra, P. Rad, and K. K. R. Choo, "Implementation of deep packet inspection in smart grids and industrial internet of thingschallenges and opportunities," *Journal of Network and Computer Applications*, vol. 135, pp. 32–46, 2019.

[3] M. Sainz, I. Garitano, M. Iturbe, and U. Zurutuza, "Deep packet inspection for intelligent intrusion detection in softwaredefined industrial networksa proof of concept," *Logic Journal of IGPL*, vol. 28, no. 4, pp. 461–472, 2020.

[4] S. Rezaei and X. Liu, "Deep learning for encrypted traffic classificationan overview," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 76–81, 2019.

[5] S. Wang, Z. Chen, L. Zhang et al., "TrafficAVan effective and explainable detection of mobile malware behavior using network traffic," in *Proceedings of the 24th International Symposium on Quality of Service (IWQoS)*, pp. 1–6, Beijing, China, June, 2016.

[6] Q. Shang, L. Feng, and S. Gao, "A hybrid method for traffic incident detection using random forest-recursive feature elimination and long short-term memory network with bayesian optimization algorithm," *IEEE Access*, vol. 9, pp. 1219–1232, 2021.

[7] M. Alrowaily, F. Alenezi, and Z. Lu, "Effectiveness of machine learning based intrusion detection systems," in *Proceedings of the International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, pp. 277–288, Springer, Atlanta, GA, USA, June, 2019.

[8] J. A. K. Suykens, L. Lukas, P. Van Dooren, B. De Moor, and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

[9] E. Min, J. Long, Q. Liu, J. Cui, and W. Chen, "TR-IDSanomaly-based intrusion detection through text-convolutional neural network and random forest," *Security and Communication Networks*, vol. 2018, Article ID 4943509, 9 pages, 2018.

[10] T. Stöber, "Who do you sync you are? smartphone fingerprinting via application behaviour," in *Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and mobile Networks*, Budapest, Hungary, April, 2013.

[11] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescape, "Multi-classification approaches for classifying mobile app traffic," *Journal of Network and Computer Applications*, vol. 103, pp. 131–145, 2018.

[12] H. F. Alan and J. Kaur, "Can Android applications be identified using only TCP/IP headers of their launch time traffic?" in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and mobile Networks*, pp. 61–66, Darmstadt, Germany, July, 2016.

[13] M. Abbasi, A. Shahraki, and A. Taherkordi, "Deep learning for network traffic monitoring and analysis (ntma)a survey," *Computer Communications*, vol. 170, pp. 19–41, 2021.

[14] J. Cheng, R. He, and E. Yuepeng, "Real-time encrypted traffic classification via lightweight neural networks," in *Proceedings of the GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pp. 1–6, Taipei, Taiwan, March, 2020.

[15] S. Rezaei, B. Kroencke, and X. Liu, "Large-scale mobile app identification using deep learning," *IEEE Access*, vol. 8, pp. 348–362, 2020.

[16] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," in *Proceedings of the 2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 43–48, Beijing, China, July, 2017.

[17] W. Wang, Z. Ming, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *Proceedings of the 2017 International Conference on Information Networking (ICOIN)*, pp. 712–717, Da Nang, Vietnam, January, 2017.

[18] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Mobile encrypted traffic classification using deep

learningexperimental evaluation, lessons learned, and challenges," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 445–458, 2019.

[19] M. Lotfollahi, M. Jafari Siavoshani, R. Shirali Hossein Zade, and M. Saberian, "Deep packeta novel approach for encrypted traffic classification using deep learning," *Soft Computing*, vol. 24, no. 3, pp. 1999–2012, 2020.

[20] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescape, "MIMETICmobile encrypted traffic classification using multimodal deep learning," *Computer Networks*, vol. 165, Article ID 106944, 2019.

[21] Y. Zeng, H. Gu, W. Wei, and Y. Guo, "Deep-full-rangea deep learning based network encrypted traffic classification and intrusion detection framework," *IEEE Access*, vol. 7, pp. 45182–45190, 2019.

[22] C. Liu, L. He, G. Xiong, Z. Cao, and Z. Li, "Fs-NetA flow sequence network for encrypted traffic classification," in *Proceedings of the IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, Paris, France, May, 2019.

[23] H. Lim, J. Kim, and J. Heo, "Packet-based network traffic classification using deep learning," in *Proceedings of the 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pp. 46–51, Okinawa, Japan, February, 2019.

[24] X. Hu, C. Gu, Y. Chen, X. Chen, and F. Wei, "OpenCBDa network-encrypted unknown traffic identification scheme based on open-set recognition," *Wireless Communications and Mobile Computing*, vol. 202218 pages, Article ID 1746373, 2022.

[25] B. Sun, W. Yang, and M. Yan, "An encrypted traffic classification method combining graph convolutional network and autoencoder," in *Proceedings of the 2020 IEEE 39th International Performance Computing and Communications Conference (IPCCC)*, pp. 1–8, Austin, TX, USA, November, 2020.

[26] S. Rezaei and X. Liu, "How to achieve high classification accuracy with just a few labelsa semi-supervised approach using sampled packets," in *Proceedings of the 19th Industrial Conference on Data Mining (ICDM 2019)*, New York, NY, USA, July, 2018.

[27] J. Bronskill, J. Gordon, and J. Requeima, "Tasknormrethinking batch normalization for meta-learning," in *Proceedings of the International Conference on Machine Learning. PMLR*, pp. 1153–1164, Vienna, Austria, July, 2020.

[28] S. Flennerhag, A. A. Rusu, and R. Pascanu, "Meta-learning with warped gradient descent," in *Proceedings of the 3rd Workshop on Meta-Learning at NeurIPS*, Vancouver, BC, Canada, August, 2019.

[29] M. Yin, G. Tucker, and M. Zhou, "Meta-learning without memorization," in *Proceedings of the International Conference on Learning Representations(ICLR)*, Vancouver, BC, Canada, April, 2019.

[30] M. Goldblum, S. Reich, L. Fowl, and R. Ni, "Unraveling meta-learningunderstanding feature representations for few-shot tasks," in *Proceedings of the 37th International Conference on Machine Learning*, pp. 3607–3616, Vienna, Austria, July, 2020.

[31] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning*, pp. 1126–1135, PMLR, Sydney, Australia, August, 2017.

[32] A. Antoniou, H. Edwards, and A. Storkey, "How to train your MAML," in *Proceedings of the International Conference on Learning Representations(ICLR)*, New Orleans, LA, USA, June, 2019.

[33] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *Proceedings of the International Conference on Learning Representations(ICLR)*, pp. 1–11, Toulon, France, May, 2017.

[34] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," 2018, https://arxiv.org/abs/1803.02999.

[35] A. Paszke, S. Gross, and F. Massa, "Pytorchan imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[36] A. H. Lashkari, G. Draper-Gil, and M. Mamun, "Characterization of encrypted and VPN traffic using time-related features," in *Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP)*, pp. 407–414, Rome, Italy, February, 2016.

[37] A. H. Lashkari, A. Kadir, and L. Taheri, "Toward developing a systematic approach to generate benchmark android malware datasets and classification," in *Proceedings of the 2018 International Carnahan Conference on Security Technology (ICCST)*, pp. 1–7, Montreal, Canada, April, 2018.

[38] X. Ren, H. Gu, and W. Wei, "Tree-RNNtree structural recurrent neural network for network traffic classification," *Expert Systems with Applications*, vol. 167, Article ID 114363, 2021.