Global-Aware Prototypical Network for Few-Shot Encrypted Traffic Classification

Jingyu Guo*†, Mingxin Cui*†, Chengshang Hou*†, Gaopeng Gou*† Zhen Li*† Gang Xiong*† Chang Liu*†✉
*Institute of Information Engineering, Chinese Academy of Sciences
†School of Cyber Security, University of Chinese Academy of Sciences
{guojingyu,cuimingxin,houchengshang,gougaopeng,lizhen,xionggang,liuchang}@iie.ac.cn

***Abstract*—Performing encrypted traffic classification under a few-shot scenario is vital because of labor-intensive labeling and intrinsically rare samples. Most existing methods apply metric learning to solve the problem of few-shot encrypted traffic classification. However, those methods only involve local information of traffic inputs to distinguish different traffic types, which weakens classification performance. In this paper, we devise Global-aware Prototypical Network (GP-Net) for few-shot encrypted traffic classification by aggregating the global information of the traffic inputs. Specifically, GP-Net firstly captures the relations between any two bytes of payload sequence, regardless of the spatial distance, and then utilizes the byte-wise relationships to aggregate the global information of traffic inputs. Moreover, we model the position information of bytes in payload sequence by leveraging the relative position mechanism, which enhances the express ability of GP-Net. We conduct extensive experiments on the real-world traffic dataset to evaluate the effectiveness of GP-Net. The experimental results demonstrate that GP-Net achieves high performance when recognizing a new traffic type even when the number of traffic samples is less than 20, outperforming state-of-the-art (SOTA) few-shot encrypted traffic classification methods.**

***Index Terms*—traffic classification, meta-learning, few-shot learning, prototypical network**

I. Introduction

Encrypted traffic classification is the process of mapping a traffic flow to its corresponding application, which is critical in network management and cyberspace security [1]. In recent years, Deep-Learning (DL) based methods have attracted more and more interest in the encrypted traffic classification field due to their excellent performance [2]. DL-based methods require a large number of traffic samples to optimize the algorithm to achieve high accuracy in traffic classification tasks. However, the new traffic types emerge constantly in the real network environment, making it impractical to collect numerous traffic samples in a short time for DL-based methods in some situations. Therefore, there is a need for few-shot encrypted traffic classification to classify new traffic types with a limited number of samples.

Meta-learning methods [3], [4] are proposed to effectively solve the problem of few-shot learning, which has been gradually applied in the encrypted traffic classification field. Metric-based methods are one of the most well-studied meta-learning algorithms in the encrypted traffic classification field because those methods are simple and effective. As shown in Fig.1, metric-based methods obtain universal traffic classification knowledge in the meta-training phase and generalize it to identify new traffic types in the meta-testing phase. Specifically, metric-based methods use feature extractor and similarity kernel to perform traffic classification tasks. Feature extractor generates embeddings for all traffic inputs, and then similarity kernel is used to give a recognition result via computing the similarity between the traffic embeddings.

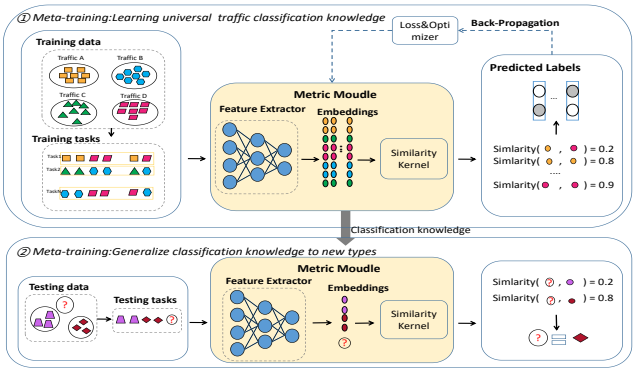


Fig. 1. The illustration of metric-based methods for few-shot encrypted traffic classification

Most existing metric-based methods for few-shot encrypted traffic classification use feature extractor to construct an embedding space that could better adapt to similarity kernel [5]–[7]. As noted by [8], the comparison ability of metric-based methods is closely related to the feature extractor since usually the similarity kernel is very simple. Previous work used the Convolutional Neural Network (CNN) to construct the feature extractor. Specifically, FC-Net [5] used four stacked 3D-CNN blocks to generate embeddings of all traffic samples by utilizing payload information. Triplet Fingerprinting (TF) [6] and Flow-Based Relation Network (RBRN) [7] used 1D-CNN to extract traffic features from the traffic flow sequences. However, the convolutional kernel is small while the traffic flow sequence (or payloads) typically contains hundreds of packets (or bytes). As a result, the feature extractor of those methods fails to involve the global information of the whole traffic input, leading to an insufficient comparison in subsequent similarity kernel module.

In this paper, we present a new method for few-shot en-

✉ Corresponding author. E-mail address: liuchang@iie.ac.cn
ISBN 978-3-903176-48-5

crypted traffic classification namely Global-Aware Prototypical Networks (GP-Net), which represents the traffic flow with the global information of the whole traffic inputs, improving the comparison ability of the existing deep models for few-shot encrypted traffic classification. Specifically, GP-Net consists of traffic normalization, global-aware representation, embedding generator, and similarity kernel four modules. Traffic normalization transforms the raw traffic data into traffic images. The key challenge of GP-Net is to learn a better representation with insufficient samples of novel traffic types. We aggregate the global information of the whole traffic inputs by leveraging the idea of self-attention mechanism in the global-aware representation module, which enhances the representation ability of the prototypical network. Moreover, we model the position information of bytes in payload sequence to strengthen the dependencies of several consecutive bytes in a traffic flow, which overcame the position-agnostic property in self-attention mechanism. In this way, each convolution operation in subsequent embedding generator module leverages the global information even the convolutional kernel is small. At last, the similarity kernel gives the classification results by properly performing comparison between those traffic embeddings.

Our main contributions are summarized as follows:

- We propose GP-Net to improve few-shot encrypted traffic classification, which enhances the representation ability by aggregating the global information of encrypted flows.
- GP-Net captures and models the relative position information of bytes in payload sequences to strengthen the dependencies of consecutive bytes in encrypted traffic.
- We compare GP-Net with three SOTA methods on real-world dataset, and the superior performance of GP-Net proved its effectiveness.

The rest of the paper is organized as follows. We first retrospect the most relevant work to our methods in Section II. Then, the preliminaries are described in Section III. In Section IV, we provide the details of our proposed GP-Net. Then, in Section V, we present the experimental results and analysis. Finally, the paper is concluded in Section VI.

II. Related Work

Previous works on traffic classification involve many different techniques. DL-based methods are the mainstream in this area [2]. To help those methods to learn new traffic types with only limited samples, researchers propose a new topic — few-shot encrypted traffic classification. We retrospect the most relevant work to our methods in those two parts.

A. Conventional DL-based methods for traffic classification

Deep neural networks have achieved remarkable success on various tasks like computer version and natural language processing [9]. In the encrypted traffic classification field, many studies have shown that it is possible to design DL algorithms with the purpose of classifying different traffic types [10]–[22]. Wang et al. leveraged the 2D-CNN to learn the representation of raw traffic flow, achieving excellent performance on real-world datasets [11]. In study [14], researchers attempted to utilize 1D-CNN to extract the feature of traffic data because it is more reasonable to consider the traffic data as sequential data. In order to capture both spatial and temporal features of traffic flow, Wang et al. proposed a mixed model which contains CNN and Recurrent Neural Networks (RNN) to identify mobile traffic [17]. On the other hand, applying RNN alone for encrypted traffic classification could also achieve excellent performance [12], [20]–[22]. Liu et al. proposed FS-Net, which is an encoder-decoder structure only using RNN to learn representative features from the raw flows and classify them [12]. In recent years, many researchers tend to apply self-attention mechanism to capture more long-range dependencies of traffic flows to implement encrypted traffic classification [15], [16]. Moreover, auto-encoder [13] also is a frequently used structure in DL-based traffic classification methods. However, those methods require large amounts of traffic data to achieve excellent performance, which excludes some applications where traffic data is rare. Therefore, there is a need for few-shot encrypted traffic classification methods.

B. Few-shot encrypted traffic classification

Few-shot learning aims to solve classification problems with a small number of examples and make the deep learning model generalize better to test classes [3], [4]. Meta-learning is an effective technique to realize the aim of few-shot learning. In this paper, we focus on metric-based meta-learning methods, where our methods belong to. Koch et al. proposed siamese neural network to compute the representation of input samples and distance vector between them [23]. Vinyals et al. proposed matching network, which maps a small labeled support set and an unlabelled example to its labels without fine-tuning [24]. In order to reduce the complexity of existing methods for meta-learning, prototypical network [25] was proposed by Snell et al. Prototypical network learns an embedding space where the model generates a prototype representation of each class and perform classification via computing distance between the query examples to them. Prototypical network is a simple but effective method. Therefore, we choose prototypical network as our basic meta-learning framework.

In encrypted traffic classification field, there are three representative metric-based methods for few-shot classification. TF [6] and RBRN [7] use 1D-CNN as the feature extractor to generate embeddings for input traffic and then given classification results by similarity kernel, while FC-Net [5] utilizes 3D-CNN to capture both spatial and temporal features of traffic flow. However, the comparison ability of those methods is still limited because the representation of the traffic flow fails to reflect the global information of the whole input.

III. Preliminaries

In this section, we first introduce the basic concept of meta-learning. Then, we give the definition of the few-shot encrypted traffic classification problem, which needs to be solved in our study.

A. Meta-learning

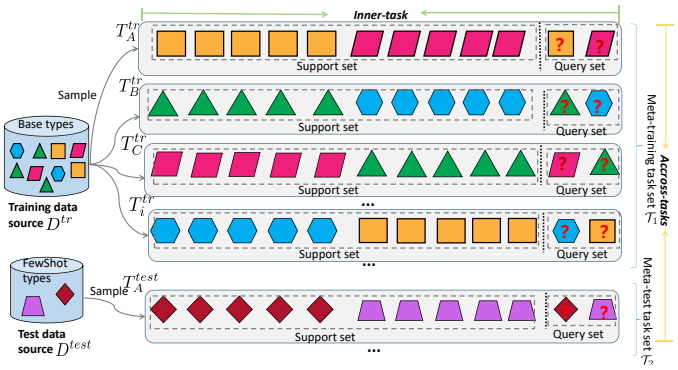


Fig. 2. The illustration of the division of meta-learning tasks

In conventional machine learning, the deep model requires a large number of training samples to achieve good performance. This requirement severely limits the ability of deep models to learn a new concept quickly, which contrasts with true intelligence as humans learn rapidly from very little data. To address this problem, meta-learning based methods are proposed [3], [4]. Different from conventional machine learning, the basic unit of meta-learning is a task instead of a training example. The main goal of meta-learning is to obtain the universal meta-knowledge from meta-training tasks and adopt it to rapidly learn meta-testing tasks with just a few samples.

B. Problem definition

In order to perform encrypted traffic classification under few-shot scenario, we assume there has two traffic data source $D^{tr} = \{(x_1^{tr}, y_1^{tr}), (x_1^{tr}, y_1^{tr}), \cdots, (x_K^{tr}, y_K^{tr})\}, x_i^{tr} \in \mathbb{R}^d, y_i^{tr} \in \{0, ...C\}$ and $D^{test} = \{(x_1^{test}, y_1^{test}), (x_1^{test}, y_1^{test}), \cdots, (x_M^{test}, y_M^{test})\}, x_i^{test} \in \mathbb{R}^d, y_i^{test} \in \{C+1, \cdots C+N\}$. There are C traffic types in D^{tr} and each traffic type has numerous samples, which are defined as base type. However, the traffic types in D^{test} are few-shot types, only has limited samples for each class. As shown in Fig.2, we define each task in our study as a binary encrypted traffic classification task. Base types as the data source to conduct meta-training task set $\mathcal{T}_1 = \{T_A^{tr}, T_B^{tr}, T_C^{tr}, \cdots\}$, and the meta-testing task set $\mathcal{T}_2 = \{T_A^{test}, T_B^{test}, T_C^{test}, \cdots\}$ is constructed from few-shot types similarly. Each specific task T_i^{tr} (or T_i^{test}) includes support set and query set two components, which are sampled form D^{tr} (or D^{test}). The sample process follow $N-way\ k-shot$ principle, which states that support set in each task contains N traffic types and k examples per type. Our aim is to obtain the universal meta-knowledge about traffic classification from the meta-training task set $\mathcal{T}_1$, and adopt it to rapidly learn new tasks in $\mathcal{T}_2$ which contains the new types with just a few samples. As a result, even though the traffic types in D^{tr} and D^{test} are disjoint, the classifier could also assign the class label $\hat{y}$ to the query sample x^q with only limited labelled samples in the support set when performing meta-testing tasks.

IV. GP-Net Framework

In this section, we give the details of the proposed GP-Net for few-shot encrypted traffic classification. At first, we will present the overall framework of GP-Net, and then each component of GP-Net will be introduced in detail.

A. GP-Net Overview

The GP-Net architecture is given in Fig.3, consisting of traffic normalization, global-aware representation, embedding generator and similarity kernel four modules. As illustrated in Section III, the basic training unit of the GP-Net is a task. The traffic data source used by each task (also known as episode) contains support set and query set. The classifier sees the support set and extracts information from it to guide its predictions on the query set. The key issue for GP-Net is how to measure the similarity between the representations of each class in support set and query sample.

At first, the traffic examples in support set and query set are fed into the traffic normalization module. With the traffic normalization module, all input traffic samples are transformed from the raw traffic payload byte sequence to a 28*28 traffic image, which is the required format of GP-Net.

Then, in order to make a more comprehensive comparison between the traffic inputs from query set and support set (of which we know the labels), we apply multi-head self-attention mechanism over them to enhance the global feature of traffic inputs via capturing long-range dependencies of the byte sequence in the global-aware representation module. Therefore, the output feature map contains the global information of the whole traffic image. In this way, even if the subsequent embedding generator module contains convolution operations to produce the final representation of the traffic image, it also involves the global information of the two input features.

Next, GP-Net computes the prototype of each class in support set. At last, the classification results of new input traffic is performed by measuring the distance between the embedding of new traffic input and prototypes in the similarity kernel. More concretely, the new traffic input is classified as the class whose prototype has the nearest distance with the new traffic input.

B. Traffic Normalization

The definition of flow is the packets that have the same quintuple [source IP, destination IP, source Port, destination Port, protocol]. Following the previous works [11], we use the first 784 bytes to represent the traffic flow. The details of the three procedures are described as follows:

- ***Traffic Segmentation***: In the first step, we split continuous raw traffic which contains data from all layers to multiple discrete traffic flows. The tool we used to divide the raw traffic into flows is SplitCap, which take a PCAP format file as input and output smaller PCAP files based on quintuple [source IP, destination IP, source Port, destination Port, protocol].
- ***Address Anonymization***: The goal of the second step is to randomize the Media Access Control (MAC) address and

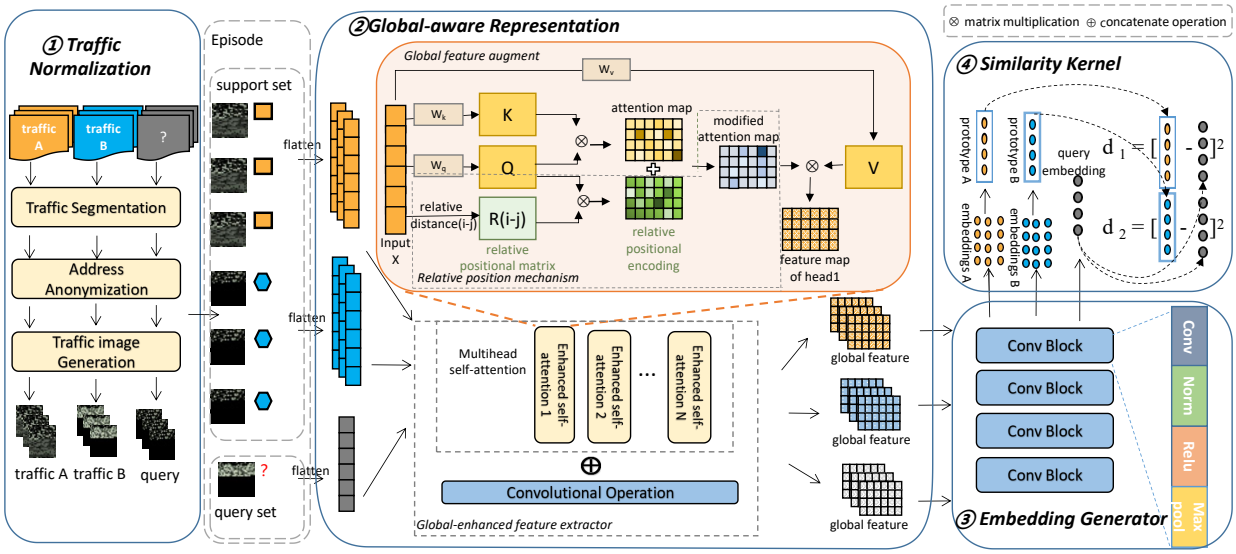


Fig. 3. The overview of GP-Net

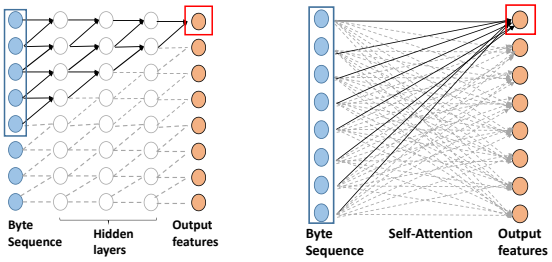
IP address in the data link layer and IP layer, respectively. IP address and MAC address are available in Local Area Network (LAN), while they should not be used for features in a real network environment where the traffic are collected from many different network environments [26]. Therefore, we randomize the MAC address and IP address to make it closer to a real network environment.

- ***Traffic Image Generation***:In this step we transform the payload sequences to gray traffic images for few-shot encrypted traffic classification. Square image of size 28*28 is used for batch integration, consistent with [10], [11]. Specifically, we select the first 784 bytes of the application layer data of each flow. The part exceeding 784 bytes is truncated, and the insufficient part is filled with 0x00. Then, the payload sequences will be reshaped to 2D square matrices. At last, those matrices will be converted to grayscale images, and each pixel of the image represents a byte in the original file.

C. Global-aware representation

As shown in Fig.3, the key component of the global-aware representation is the global-enhanced feature extractor. Global-enhanced feature extractor leverages the global feature augment module to aggregate the information of the whole traffic inputs. Moreover, we introduce relative position mechanism to model the position information of bytes in the global feature augment module. The detailed information about the global-aware representation is described as follows.

1) Global feature augment: Previous metric-based studies just used CNN as the feature extractor to produce the representation of the flow, failing to involve the global information of the whole traffic image. This phenomenon is illustrated in Fig.4(a). Even if there stack four convolution blocks, the first element of the output only modeled five bytes in payload sequence, indicating that only using CNN as a feature extractor



(a) Local information (b) Global information

Fig. 4. The local and global information in byte sequence

can not capture the long-range dependencies between payload bytes. As a result, in subsequent comparison, only local information is used to measure the similarity between the two examples. To deal with this problem, the one immediate idea is to use a large convolution kernel or stack more convolutional layers. However, those methods will introduce more parameters for the model, leading to overfitting problem especially under the few-shot scenario.

Recently, there have been some studies using self-attention mechanism to complement the convolution in Natural Language Processing (NLP) [27] and Reinforcement Learning (RL) field [28]. Inspired by that, the global feature augment module is proposed in the global-aware representation to capture long-range dependencies for traffic bytes in payload. The global feature augment module just uses a small number of the parameters to capture the relationship between any two bytes, regardless of their spatial distance, and then aggregate the global information at each element of outputs as shown in Fig.4(b). In this way, even if the subsequent convolutional kernel in the embedding generator is small, each convolution operation involves the global information of the traffic image, improving the performance of the similarity kernel.

For a given traffic image x, we first flatten it to a vector $X \in \mathbb{R}^{H \times W}$, and then perform self-attention mechanism as proposed in Transformer architecture [29] to enhance the global features of the traffic inputs. Following the self-attention operation, feature map A is produced, which can be formulated as follows:

$$A = \text{Softmax}\left(\frac{(XW_q)(XW_k)^T}{\sqrt{d_k^h}}\right)(XW_v), \tag{1}$$

where $W_q, W_k \in \mathbb{R}^{d_k^h}$ and $W_v \in \mathbb{R}^{d_v^h}$ are learned linear transformations that map the traffic input X to queries $Q = XW_q$, keys $K = XW_k$ and values $V = XW_v$. K can be seen as the bytes in payload sequence, and each byte has a value that corresponds to an element of V. Q is also the bytes in payload sequence, which are expected to calculate the relationships between the key-value pairs. The output A contains the global information in all bytes of traffic inputs, including two calculation steps. First, we capture the relationships between bytes in payload sequences by computing a compatibility function of the Q with the corresponding K. Then the weighted sum of the values V was computed to aggregate the global information by utilizing the byte-wise relationships captured in the first step.

2) Relative position mechanism: However, original self-attention mechanism in the global feature augment module is position-agnostic. This property loses the location information of the bytes, leading to an incomprehensive representation of traffic flow. Therefore, we introduce the relative position mechanism, which is proposed in [30] to enhance the representation of traffic flow. The output of self-attention becomes:

$$A = \text{Softmax}\left(\frac{(XW_q)(XW_k)^T + P^{rel}}{\sqrt{d_k^h}}\right)(XW_v), \tag{2}$$

where $P^{rel} \in \mathbb{R}^{HW \times HW}$ is the matrix of relative position of all bytes in payload sequence. The relative position of byte i to byte j can be written as follows:

$$P^{rel}[i,j] = q_i r_{j-i}, \tag{3}$$

where q_i is the query vector for byte i (the i_{th} element of Q) and r_{j-i} is the learned embedding for relative position between byte i and byte j.

3) Global-enhanced feature extractor: In global-enhanced feature extractor, we fuse the global information from different subspaces generated by global feature augment and relative position mechanism. Then, the output global feature maps from different subspaces are concatenated and projected as follows:

$$MA = Concat[A_1, A_2, \cdots, A_N]W_O. \tag{4}$$

At last, we concatenate the feature maps generated by multi-head attention mechanism and standard convolution as the final output of the global-aware representation module:

$$O = Concat[MA, Conv(X)] \tag{5}$$

The output O contains global information aggregated by a multi-head attention mechanism. After that, O will be passed through the embedding generator to produce the final representation of the traffic flow.

D. Embedding Generator

After obtaining the globally related features from the global-aware representation module, we stacked four identical convolution blocks to obtain more comprehensive embeddings of traffic samples. Each convolution block contains 3×3 convolutional layer (with 64 filters), batch normalization layer, Rectified Linear Units (ReLu) nonlinear activation function and max-pooling layer four components. Despite the small kernel size, each convolution operation also involves global information because the features of the whole traffic inputs are aggregated in the global-aware representation module. We denote such an embedding generator module as the follows:

$$e = f_\phi(O). \tag{6}$$

After that, we obtain the final representation of each traffic input which contains the global information of the whole traffic image, and then pass through a similarity kernel module to give the recognition result of the query example.

E. Similarity Kernel

In the similarity kernel, GP-Net performs the classification for the query sample x^q by comparing the distance between the embedding of query sample e^q and each class prototype c_i in support set, which is a single vector representation of each class in support set. The computation of the class prototype c_i is to average all the embeddings from the same traffic types in support set. The formal definition of prototype c_i can be written as follows:

$$c_i = \frac{1}{K}\sum_{j=1}^{K} e_i^j, \tag{7}$$

where e_i^j is the j^{th} traffic sample form the i^{th} class in support set and K is the number of samples in per support set class. Finally, we compute class probabilities as follows:

$$p_{\boldsymbol{\theta}}(y = i \mid x, S) = \frac{\exp\left[-d\left(e^q, c_i\right)\right]}{\sum_N^{k=1} \exp\left[-d\left(e^q, c_k\right)\right]}, \tag{8}$$

where d is a distance function for two given vectors and N is the total traffic types in support set. There has multiple choice for the distance function. We found that the performance of Euclidean distance is the best, which is consistent with prototypical network. Finally, the query traffic sample is classified as the traffic type with the nearest prototype distance.

V. Experimental evaluation

Our experiments mainly answer the following research questions:

- **Question 1:** Could GP-Net generalize the encrypted traffic classification knowledge obtained from base types to few-shot types, and how does the performance of GP-Net compare to the other representative baseline methods?

TABLE I
THE STATISTICAL INFORMATION OF USTC-TFC2016 DATASET

Apps	Flows	Apps	Flows
BitTorrent	14722	Cridex	16023
Facetime	5885	Geodo	13540
FTP	13342	Htbot	12425
Gmail	16880	Miuref	9576
MySQL	14181	Neris	16708
Outlook	14704	Nsis-ay	10917
Skype	11772	Shifu	15468
SMB	12416	Tinba	15879
Weibo	12765	Virut	12928
WoW	15454	Zeus	11823

TABLE II
THE DIVISION OF BASE TYPES AND FEW SHOT TYPES

Traffic types	#Classes	#Samples per class	Generate tasks
BaseTypes	12	5000	meta-training
FewShotTypes	8	300	meta-testing

- **Question 2:** Can global feature augment module and relative position mechanism improve the comparison ability of the prototypical network?
- **Question 3:** How do different choices of parameters affect the performance of GP-Net?

We construct extensive experiments to answer those questions. In the following, we present the details of the dataset, experimental setup, experiment results of comparison, absolution results and sensitivity analysis.

A. Dataset and Metric

1) Dataset: USTC-TFC2016 [11] is a public widely-used encrypted traffic dataset that contains original raw traffic for traffic classification tasks. We take the USTC-TFC2016 dataset as our data source to conduct few-shot encrypted traffic classification tasks. USTC-TFC2016 dataset contains ten types of malware traffic collected by Cal Tech University (CTU) researchers and ten types of normal traffic collected by IXIAS BPS. Those traffic types cover various areas such as email, music, video and searching, which is rich enough to evaluate the effectiveness of GP-Net.

We provide the statistics information of USTC-TFC2016 in Table I. As can be seen, USTC-TFC2016 is a relative balance dataset where 19 out of 20 classes have more than 10000 flows. As mentioned in Section II, the traffic types in USTC-TFC2016 are supposed to be divided into base types for meta-training and few-shot types for meta-testing, respectively. Therefore, we randomly sample 12 traffic types out of USTC-TFC2016 as base types and the rest 8 traffic types as few-shot types. For each traffic class of base types, we randomly select 5000 samples as the data source to conduct training tasks in the meta-training phase. For each traffic class of few-shot types, we randomly sample 300 samples to construct the few-shot encrypted traffic classification tasks in the meta-testing phase. The description of the division of base types and few-shot types is summarized in Table II.

TABLE III
HYPERARPARAMETER SETTINGS

Hyperarparameter	Value
Batch	500
Episode	10000
#Headers	2
ratio of attention channnels	0.2
#Conv kerenls	64
Similarity Metrics	Euclidean
Optimizer	SGD
Embedding Size	64

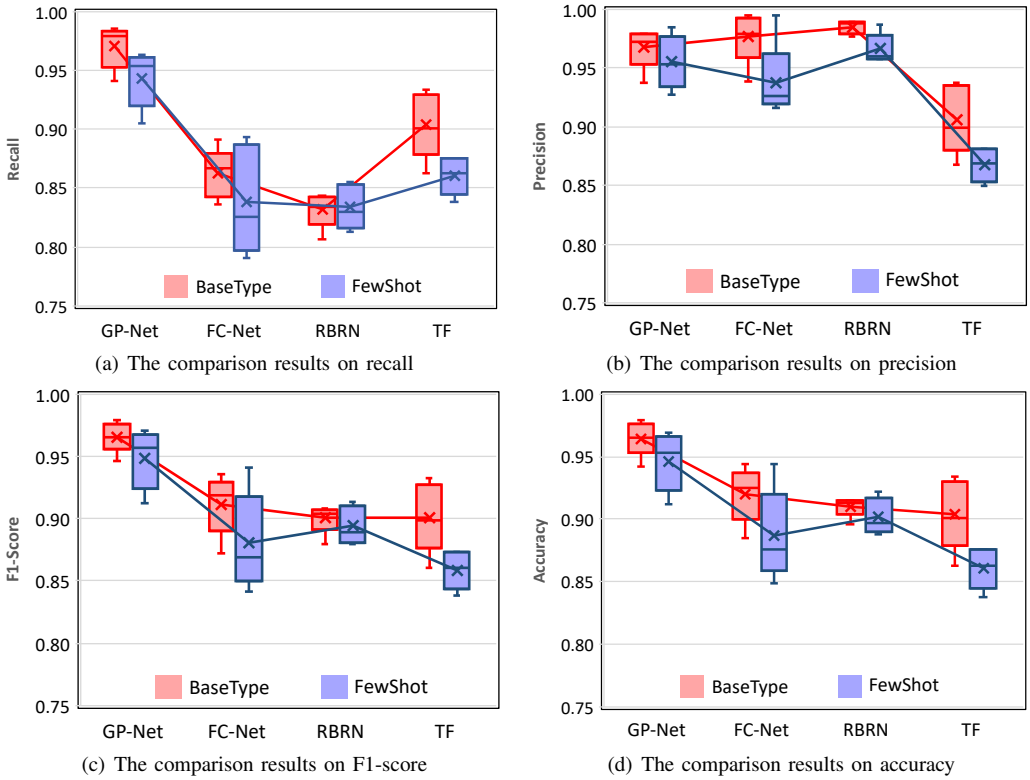
2) Metric: As mentioned in Section III, the basic unit of meta-learning is a task. In our study, each meta-learning task is defined as a binary traffic classification task involving the same amount of positive samples and negative samples. Therefore, ground on the estimation of the ability to accomplish the binary tasks, we choose the recall, precision, F1-score and accuracy as the metrics to evaluate the performance of GP-Net and the other representative SOTA methods.

However, in few-shot encrypted traffic classification field, the number of traffic samples involved in a single task is small, making it lack statistical significance. To generate the rigor of experiments, we generate 1000 tasks in the meta-testing phase and repeat each experiment multiple times. We average the metrics of all experiments to evaluate the overall performance of GP-Net and the other representative SOTA methods.

B. Experimental Setup

1) Hyperparameter Tuning: Our model is implemented with Tensorfolw 1.15 and trained on a Tesla P100. To develop GP-Net, we follow the episode training mechanism from the previous work to implement few-shot classification tasks [24]. Moreover, we carefully select the hyperparameters of GP-Net to achieve the best performance, which is summarized in Table III. In most cases, we use two headers to capture the global information from different subspaces, set the ratio of attention channels (the number of output channels in the global-aware representation module divided by attention channels) to 0.2 in the global-aware representation module, set the number of convolution kernels to 64. We train GP-Net by Adam Optimizer with 0.001 learning rate and 500 epoch which contains 10000×500 episodes in total.

2) Few-shot Encrytpted Traffic Classification Tasks: In our study, we focus on binary traffic classification tasks with a few samples. Following [24], we adopt the episode training mechanism to generate tasks and train the model with those tasks. In order to conduct a binary task for meta-training, we randomly sample two classes from the base types and thirty labeled samples per class. Twenty out of those labeled samples in each class are used as support set (of which we know the labels). The left ten examples per class form query set, which are the testing data source of this task. The meta-training task set is generated by repeating the above process thousands of times. The construction of the meta-testing task set is similar to the meta-training task set. Note that base types and few-



(a) The comparison results on recall

(b) The comparison results on precision

(c) The comparison results on F1-score

(d) The comparison results on accuracy

Fig. 5. The distribution of GP-Net results on five parallel experiments compared to other approaches

TABLE IV
OVERALL GP-NET EXPERIMENTAL RESULTS COMPARED TO OTHER APPROACHES

Method	Types	Metric			
		Recall	*Precision*	*F1-score*	*Accuracy*
GP-Net	Base	**0.9696**	0.9667	**0.9648**	**0.9640**
	FewShot	**0.9423**	0.9544	**0.9474**	**0.9458**
FC-Net	Base	0.8617	0.9759	0.9109	0.9193
	FewShot	0.8381	0.9373	0.8803	0.8862
RBRN	Base	0.8310	**0.9841**	0.8999	0.9092
	FewShot	0.8331	**0.9656**	0.8936	0.9014
TF	Base	0.9029	0.9055	0.9006	0.9029
	FewShot	0.8600	0.8671	0.8582	0.8600

shot types are both included in the meta-testing task set. The purpose of this setting is to evaluate the fitting ability and generalization ability of GP-Net at the same time. The division of base types (12 classes) and few-shot types (8 classes) are random. We conduct five parallel experiments on different divisions of base types and few-shot types and present their result in our paper.

C. Comparison Study(Response to Question1)

We compare GP-Net to three SOTA metric-based methods for few-shot learning, including FC-Net [5], RBRN [7] and TF [6]. RBRN and FC-NET used relation network to implement few-shot encrypted traffic classification, achieving good performance in their experimental setting. TF leveraged the triplet network to perform website fingerprinting attack with only a few samples. We re-implement them and carefully tune the parameters of compared baselines for comparison. The overall comparison result on five parallel experiments is shown in Table IV. Moreover, Fig.5 shows the distribution of experimental results of GP-Net and the other three baselines on four metrics. We have the following findings from the results:

1. GP-Net achieves excellent performances on both base types and few-shot types. Table IV shows that the GP-Net achieves over 90% on recall, precision, F1-score and accuracy four metrics for identifying all traffic types, which indicates that GP-Net has the good fitting ability and generalization ability. Fitting ability refers to the ability to recognize the base traffic types shown in the meta-training phase. Generalization ability refers to the ability to generalize the meta-knowledge learned from base traffic types to newly emerge traffic types with few samples. This phenomenon demonstrated that GP-Net is able to learn universal traffic classification knowledge from base types and generalize it to few-shot types, enabling it to distinguish different traffic types, regardless of whether it has been seen during the training phase.

2. Overall, GP-Net achieves the best performance on recall, F1-score and accuracy for all traffic type and perform second best on precision, which is only 1.74% and 1.12% worse than RBRN respectively on the base types and few-shot types. Benefiting from the global-aware representation module, GP-Net improves the comparison ability of similarity kernel by complementing the information of the whole traffic inputs, making GP-Net outperforms the other three baseline methods.

TABLE V
PERFORMANCE COMPARISON BETWEEN GP-NET AND TWO VARIANTS

Method	Component		Metrics			
	GAR	*RP*	*Precision*	*Recall*	*F-Score*	*Accuracy*
GP-Net w/o GAR	×	×	0.9442	0.9205	0.9306	0.9283
GP-Net w/o RP	✓	×	0.9423	0.9415	0.9409	0.9420
GP-Net	✓	✓	**0.9591**	**0.9676**	**0.9524**	**0.9597**

Moreover, such good results demonstrate that complementing global information of traffic inputs helps the deep models better represent traffic flows, especially in the encrypted traffic classification task where the traffic samples are insufficient.

3. Compared to TF, GP-Net has more consistent performance on few-shot encrypted traffic classification tasks. Fig.5 shows that the distribution of experimental results on GP-Net is more concentrated than TF on four metrics. TF learns the representation of traffic flows by using triplet to compare matching/nonmatching traffic types. Therefore, the selection of triplets is crucial. One possible reason to explain the phenomenon in Fig.5 is that improper triplet selection politic of TF may lead to unstable embedding distribution of traffic samples. As a result, TF fails to perform well on all parallel experiments. However, GP-Net learns discriminative features by aggregating global information, which avoids sampling traffic inputs and keeps the stability of the training process, leading to a good performance on all parallel experiments.

4. Compared to RBEN and FC-Net, GP-Net is able to simultaneously perform well on both precision and recall. RBRN and FC-NET choose relation network as the few-shot learning model. As can be seen, the performance of RBRN and FC-NET are obviously inferior to GP-Net on F1-score. They only averagely reach 89.36% and 88.03% F1-score on few-shot types, respectively. One possible reason to explain this phenomenon is that RBEN and FC-Net fail to conduct a comprehensive comparison between traffic samples due to insufficient use of information. However, GP-Net is able to capture the global information of the traffic image and generate comprehensive embeddings for traffic samples, which improves the comparison ability of the prototypical network, making GP-Net perform well on both precision and recall.

D. Absolution Study(Response to Question2)

In this section, we conduct ablation experiments to evaluate the effectiveness of the global-aware representation module and relative position mechanism. We consider two variants as follows:

- GP-Net w/o GAR (global-aware representation module): The original networks proposed in study [25] only contain four convolution blocks to extract the feature of input samples, without global-aware representation module and relative position mechanism.
- GP-Net w/o RP (relative position mechanism): Global-aware representation module (without relative position) is added to original networks to capture the global information of traffic images. However, the relative position mechanism is removed to investigate its effectiveness.

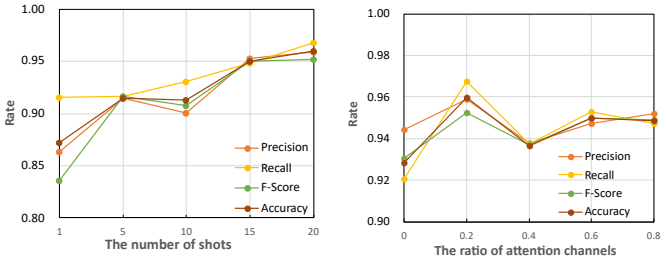
Effectiveness of global-aware representation: To validate the effectiveness of the global-aware representation module, we observe the average of GP-Net w/o GAR and GP-Net w/o RP in Table V. It can be seen that the global-aware representation improves the performance of the GP-Net w/o GAR from 93.06% and 94.09% on F1-score, demonstrating that the global-aware representation is an indispensable module for improving the performance of the prototypical network.

Effectiveness of relative position mechanism: Using only global-aware representation module will lose the position information of bytes, which weakens the dependencies of neighbor bytes. In order to address this problem, we introduce relative position mechanism to implement position information. As shown in Table V, compared to GP-Net w/o RP, GP-Net improved the F1-score from 94.09% to 95.24% on average, demonstrating that implementing position information will improve the performance of GP-Net.

E. Sensitivity Analysis(Response to Question3)

In this part, we investigate the impacts of two parameters, including the number of samples in support set and the ratio of attention channels in global-aware representation module.

Impact of the number of samples in support set: We test the performance of different $k \in [1, 5, 10, 15, 20]$ which indicate the number of samples in support set per class and show the results in Fig.6(a). Intuitively, more samples in support set contain more abundant information, which helps GP-Net generate a more accurate prototype to represent traffic type. As can be seen, more samples in support generally achieve higher performance. Moreover, Fig.6(a) shows that GP-Net recognizes new traffic types well with only less than 20 samples.



(a) Impact of the number of shots (b) Impact of the ratio of attention channels

Fig. 6. The performances of GP-Net as support samples and attention channels vary

Impact of attention channels: We denote the ratio of attention channels to the number of output channels in global-aware representation module as $v = \frac{d_k}{F_{out}}$, the larger v indicate that the ratio of feature maps generated by self-attention mechanism is more. We test $v \in [0, 0.2, 0.4, 0.6, 0.8]$ and show the results in Fig.6(b). It is clear that the performance of GP-Net is worst when setting $v = 0$, which indicates that combating self-attention and convolution outperforms convolution only due to the global information captured by multi-head self-attention. Another surprising finding is that increasing the ratio of attention channels does not always improve the performance of GP-Net. In our experiments, GP-Net achieves the best performance when $v = 0.2$.

VI. Conclusion and Future Work

In this paper, we proposed GP-Net for few-shot encrypted traffic classification. GP-Net contains traffic normalization, global-aware representation, embedding generator and similarity kernel four modules. GP-Net captures the relations between any two bytes of payload sequence and utilizes the byte-wise relationships to aggregate the global information of traffic inputs in the global-aware representation module. Moreover, we complement the position information of bytes to enhance the express ability of GP-Net. Our extensive experiments show that GP-Net recognizes novel traffic types with just a few samples and outperforms the other SOTA few-shot encrypted traffic classification methods. In the future, we will further explore more meta-learning techniques (e.g., zero-shot learning) to be applied in encrypted traffic classification field.

Acknowledgment

This work is supported by The National Key Research and Development Program of China No. 2020YFB1006100 and the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDC02040400. We are grateful to anonymous reviewers for their fruitful comments to improve this paper. We also sincerely appreciate the moral support from Rongchang Zhao.

References

[1] Pescape, A., Dainotti, Claffy, and C. K., "Issues and future directions in traffic classification," *IEEE Network: The Magazine of Computer Communications*, vol. 26, no. 1, pp. 35–40, 2012.

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

[3] M. Huisman, J. N. van Rijn, and A. Plaat, "A survey of deep meta-learning," *Artificial Intelligence Review*, pp. 1–59, 2021.

[4] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, pp. 1–34, 2020.

[5] C. Xu, S. J, and X. Du, "A method of few-shot network intrusion detection based on meta-learning framework," *IEEE Transactions on Information Forensics and Security*, vol. PP, no. 99, pp. 1–1, 2020.

[6] P. Sirinam, N. Mathews, M. S. Rahman, and M. Wright, "Triplet fingerprinting: More practical and portable website fingerprinting with n-shot learning," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 1131–1148.

[7] W. Zheng, C. Gou, L. Yan, and S. Mo, "Learning to classify: A flow-based relation network for encrypted traffic classification," in *WWW '20: The Web Conference 2020*, 2020.

[8] R. Salakhutdinov and G. E. Hinton, "Learning a nonlinear embedding by preserving class neighbourhood structure," *Journal of Machine Learning Research*, vol. 2, pp. 412–419, 2007.

[9] M. R. Minar and J. Naher, "Recent advances in deep learning: An overview," *arXiv preprint arXiv:1807.08169*, 2018.

[10] W. Wei, Z. Ming, 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)*, 2017.

[11] W. Wei, Z. Ming, 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)*, 2017.

[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*, 2019.

[13] J. Hchst, L. Baumgrtner, M. Hollick, and B. Freisleben, "Unsupervised traffic flow classification using a neural autoencoder," in *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*, 2017.

[14] M. Lotfollahi, R. Zade, M. J. Siavoshani, and M. Saberian, "Deep packet: A novel approach for encrypted traffic classification using deep learning," *Soft Computing*, 2017.

[15] B. Gxa, C. Qlb, and J. Yong, "Self-attentive deep learning method for online traffic classification and its interpretability," *Computer Networks*, 2021.

[16] J. Cheng, Y. Wu, E. Yuepeng, J. You, T. Li, H. Li, and J. Ge, "Matec: A lightweight neural network for online encrypted traffic classification," *Computer Networks*, vol. 199, p. 108472, 2021.

[17] X. Wang, S. Chen, and J. Su, "App-net: a hybrid neural network for encrypted mobile traffic classification," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020, pp. 424–429.

[18] A. S. Iliyasu and H. Deng, "Semi-supervised encrypted traffic classification with deep convolutional generative adversarial networks," *IEEE Access*, vol. 8, pp. 118–126, 2019.

[19] W. Shen, H. Zhang, S. Guo, and C. Zhang, "Time-wise attention aided convolutional neural network for data-driven cellular traffic prediction," *IEEE Wireless Communications Letters*, vol. 10, no. 8, pp. 1747–1751, 2021.

[20] 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.

[21] A. Rago, G. Piro, G. Boggia, and P. Dini, "Multi-task learning at the mobile edge: An effective way to combine traffic classification and prediction," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 10 362–10 374, 2020.

[22] C. Li, C. Dong, K. Niu, and Z. Zhang, "Mobile service traffic classification based on joint deep learning with attention mechanism," *IEEE Access*, vol. 9, pp. 74 729–74 738, 2021.

[23] G. Koch, R. Zemel, R. Salakhutdinov *et al.*, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2. Lille, 2015.

[24] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," *Advances in neural information processing systems*, vol. 29, pp. 3630–3638, 2016.

[25] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 4080–4090.

[26] J. Fan, J. Xu, M. H. Ammar, and S. B. Moon, "Prefix-preserving ip address anonymization: Measurement-based security evaluation and a new cryptography-based scheme," *Computer Networks*, vol. 46, no. 2, pp. 253–272, 2004.

[27] B. Yang, L. Wang, D. Wong, L. S. Chao, and Z. Tu, "Convolutional self-attention networks," *arXiv preprint arXiv:1904.03107*, 2019.

[28] V. Zambaldi, D. Raposo, A. Santoro, V. Bapst, Y. Li, I. Babuschkin, K. Tuyls, D. Reichert, T. Lillicrap, E. Lockhart *et al.*, "Deep reinforcement learning with relational inductive biases," in *International Conference on Learning Representations*, 2018.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[30] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," *arXiv preprint arXiv:1803.02155*, 2018.