International Journal of Innovative
Computing, Information and Control
Volume **21**, Number **2**, April **2025**

ICIC International ©2025 ISSN 1349-4198
pp. 359–371

LETFORMER: LIGHTWEIGHT TRANSFORMER PRE-TRAINING WITH SHARPNESS-AWARE OPTIMIZATION FOR EFFICIENT ENCRYPTED TRAFFIC ANALYSIS

Zhiyan Meng[1], Dan Liu[1,*] and Jintao Meng[2]

[1]Research Institute of Electronic Science and Technology
University of Electronic Science and Technology of China
No. 2006, Xiyuan Avenue, West Hi-Tech Zone, Chengdu 611731, P. R. China
iconmzy@std.uestc.edu.cn; *Corresponding author: liudan@uestc.edu.cn

[2]National Key Laboratory of Security Communication
No. 35, Huangjing Road, Shuangliu County, Chengdu 610041, P. R. China
mengjintao01@126.com

Received July 2024; revised November 2024

Abstract. *Reliable encrypted traffic classification is fundamental for advancing cybersecurity and effectively managing exponentially growing data streams. The success of large language models in fields such as natural language processing demonstrates the feasibility of learning general paradigms from extensive corpora, making pre-trained encrypted traffic classification methods a preferred choice. However, attention-based pre-trained classification methods face two key constraints: the large number of neural parameters is unsuitable for low-computation environments like mobile devices and real-time classification scenarios, and there is a tendency to fall into local minima, leading to overfitting. We develop a shallow, lightweight Transformer model named LETformer. We utilize sharpness-aware optimization during pre-training to avoid local minima while capturing temporal features with relative positional embeddings and optimizing the classifier to maintain classification accuracy for downstream tasks. We evaluate our method on four datasets – USTC-TFC2016, ISCX-VPN2016, ICCXTOR, CICIOT2022. Despite having only 17.6 million parameters, LETformer achieves classification metrics comparable to those of methods with ten times the number of parameters.*

Keywords: Encrypted traffic classification, LETformer, Deep learning, Spare relative position embedding, Sharpness-aware optimization

1. **Introduction.** As network communication technologies and protocols advance rapidly, the complexity and variety of network traffic increase significantly. To safeguard data transmissions and prevent privacy intrusions, the majority of communication networks rely on advanced encryption algorithms. Paradoxically, this encryption creates anonymity that cyber attackers exploit to hide their identities and illegal activities. The concealment of malware and sensitive information within encrypted traffic leads to cybersecurity threats, such as viruses and unauthorized access. Conventional traffic classification techniques, like Deep Packet Inspection (DPI) [1], fall short in extracting robust traffic representations from data with concealed and unbalanced content. To address these challenges, there is a growing shift towards the adoption of deep learning approaches in traffic classification. Deep learning eliminates the need for manual feature selection from network traffic. It processes raw traffic data directly, automatically learning the complex, nonlinear relationships between raw inputs and desired output labels. Models using the Convolutional Neural Network (CNN) architecture [2] create a hierarchy of spatial features through

DOI: 10.24507/ijicic.21.02.359

their layered structure, which is extensively employed in recognizing Website Fingerprinting (WF), categorizing Internet of Things (IoT) traffic, and more. However, CNNs focus only on the local features of the traffic and fail to capture the long-term dependencies among packets in a traffic trace.

The remarkable successes of large language models in image classification and natural language understanding widely validate the approach of learning general knowledge patterns from extensive corpora and the effectiveness of attention mechanisms in uncovering deep semantic correlations. In particular, methods based on pre-training followed by fine-tuning [3] have become the preferred choice in recent years. Pre-trained methods for encrypted traffic consist of two main stages. First, the model learns the relationships between encrypted packets from a large set of unlabeled traffic data and extracts relevant features. In the second stage, the model's pre-trained weights are used to initiate a fine-tuning process. With a small set of labeled data, the model is adjusted to quickly and accurately perform classification tasks in various encrypted traffic scenarios. The quadratic computational complexity of self-attention excels at identifying potential correlations among sequences, yet it brings significant deployment costs. Stacking layers to process long inputs creates a memory bottleneck. This leads to a trade-off: either use a lot of memory or take longer to process. This issue is especially challenging with large volumes of encrypted traffic that require parallel processing.

Although GPU computing power is improving, it still cannot keep up with the rapid growth of data. Therefore, it is crucial to extract consistent and stable patterns from diverse encrypted traffic while efficiently minimizing resource use. This approach is essential for achieving optimal network security and efficient network management across all scenarios. In this paper, we design a more lightweight data preprocessing and encoding structure and introduce Sharpness-Aware Minimization (SAM) [4] to avoid local minima during pre-training. This significantly enhances the effectiveness of parallel processing for traffic packets. Furthermore, it is noted that tokens within encrypted messages inherently do not explicitly reveal semantic information, so the absolute positioning of tokens is not suitable for describing temporal representations. To improve accuracy, we adopt relative positional encoding embeddings that help better extract temporal features from encrypted traffic. Experimental results show that LETformer achieves the same classification performance as the full multi-head attention model while significantly improving efficiency. Table 1 summarizes the comparison of the proposed LETformer and other existing methods. MLM stands for Masked Language Modeling Prediction, SP stands for Sentence Prediction, RL represents Relative Attention Positional Encoding, and SAM indicates the Sharpness-Aware Minimization algorithm. Our contributions are summarized as follows.

- In this paper, we present LETformer, a sparse attention mechanism designed for the efficient extraction of features from encrypted traffic using lightweight vocabulary granularity and channel attention with only 17.6m of the parameter size. We introduced the SAM strategy to address the local extrema problem commonly encountered with attention-based models. We release our work at https://github.com/iconmzy/LETformer, which is accessible after publication of the article.

Table 1. Comparison of the proposed LETformer and existing pre-training methods

Method	Tokenize method	Parameters	MLM	SP	SAM	RL
ET-BERT	Vocabulary	187.4m	✓	✓	✗	✗
YATC	Visualization	2.2m	✗	✓	✗	✗
LETformer	Vocabulary	17.6m	✓	✓	✓	✓

- Based on the non-significant semantics of encrypted traffic messages, we improved the position encoding method to construct several relative position matrices for the input sequence. The LETformer demonstrates its superiority in learning temporal features for the precise identification of anonymous network traffic.
- Finally, we validate the effectiveness of our approach through four publicly available datasets with various types of encrypted traffic.

The rest of this paper is organized as follows. In Section 2, we review the related work, discussing the progress made in encrypted traffic classification and attention-based models. Section 3 presents the preliminaries, covering essential background information and foundational concepts relevant to our approach. Section 4 details the proposed LETformer model, explaining its architecture and key components. Section 5 presents the datasets and comprehensive comparison experiments. Finally, Section 6 concludes the paper.

2. **Related Work.** Traffic classification has consistently been the focus of extensive attention due to its fundamental importance in network traffic management, the safeguarding of mobile and wireless security, and the analysis of user behavior dynamics [5]. In the early stages of network transmission, given the limited number of protocols, port number-based [6] and Deep Packet Inspection (DPI) [7] approaches were frequently employed for data matching and the identification of various protocol traffic. Owing to the widespread adoption of network encryption protocols, approaches relying on extracting substantial data from packet payload contents are steadily diminishing in effectiveness due to increasingly low data visibility. Hence, various research endeavors have been proposed to categorize network traffic even in situations where access to explicit data and port numbers remains unattainable. BLINC [8] is one of the representative works in the field that solely relies on host behavioral patterns for preliminary classification. Taylor et al. [9] developed an integrated framework that employs machine learning techniques to analyze side-channel data, including variables like packet size and direction, for enhanced data interpretation. Research studies [10, 11] within the domain of natural language processing indicate that, notwithstanding the absence of overt readability, deep learning algorithms possess the ability to discern underlying patterns within complex data streams.

In recent years, there has been a discernible pivot in research efforts toward employing deep learning techniques to extract representations with robust generalization potential from unlabeled or non-apparent traffic data. ET-BERT [12] implements a framework akin to that of BERT, undergoing pre-training with a voluminous collection of unmarked encrypted traffic data to derive a comprehensive representation that is adaptable to various encrypted traffic classification tasks. Diao et al. [13] utilize graph convolutional neural networks to generate data representations in traffic analysis, effectively transforming encrypted traffic into a series of distinct subgraphs for enhanced interpretability. In certain investigations, including CMTSNN [14], attributes such as behavioral dynamics and temporal characteristics are extracted independently. Zhou et al. [15] pioneeringly introduce the use of a relative position matrix [16] as a novel replacement for the established absolute position encoding in the Transformer model, aiming to capture the intricate temporal relationships among traffic data packets. Some approaches focus on enhancing downstream task classifiers by incorporating additional features using pre-trained neural network parameters. Bi-ETC [17] enhances the ability to capture long-range dependencies between byte feature sequences by incorporating a BiLSTM layer after BERT. LAMBERT [18] further employs relative attention to amplify the influence of key information and mitigate the impact of imbalanced data during pre-training.

3. **Preliminaries.** In this chapter, we will introduce the foundational concepts and background knowledge of several key components relied upon in this study, laying the groundwork for the detailed discussions in the subsequent chapters.

In order to maximize the compression of redundant parameters while retaining the core components of extracting potentially deep semantics, our proposed framework, LETformer, integrates channel-wise attention [19] and relative position embedding within the Transformer network architecture. This section will define and formulate the core concepts and components utilized within our LETformer framework. Since the advent of BERT [20], the Transformer series architecture has gradually emerged as the mainstream in the field of machine learning. This trend has persisted, extending to the large-scale models exemplified by GPT in contemporary times. We first dissect the self-attention mechanism, which is the cornerstone of the Transformer. The classic self-attention mechanism allocates distinct weights to individual elements within the input sequence by absolute position embedding, facilitating the model's ability to gauge the significance of each element in correspondence to the others. The calculation of attention weight can be expressed as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

where Q, K, and V correspond to query, key, and value matrices respectively, which are obtained by multiplying the embedding vector of the input sequence by the corresponding weight matrix. The scaling factor d_k is used to normalize the dot product of the query Q and the transpose of the key K, scaled down by the square root of d_k.

Self-attention lacks an intrinsic mechanism to process sequential positional information; thus, absolute positional encoding is generally incorporated directly into the embedding vector of each element. The calculation of this positional encoding can be characterized by the following expression:

$$\overrightarrow{p_t}^{(i)} = \begin{cases} \sin\left(\dfrac{1}{10000^{2k/d}} \cdot t\right), & \text{if } i = 2k \\ \cos\left(\dfrac{1}{10000^{2k/d}} \cdot t\right), & \text{if } i = 2k+1 \end{cases} \tag{2}$$

where t represents the position index in a sequence, k represents the dimension index, and d is the total number of dimensions in the position encoding vector. This method of positional encoding allows the model to comprehend the discrete position of an individual word or constituent within the input sequence. However, it does not directly encode and capture the relational positional information among various elements. Particularly in the context of network traffic packet sequences, employing relative position embedding can effectively depict the temporal features between elements. For instance, within the Ethernet protocol framework, the source MAC address is positioned between the destination MAC address and the type field, but the overall length of each data frame is not fixed.

Consequently, we adopt a more fitting approach to relative position representations, envisioning the input sequence as a degenerate graph. For each pair of relative position tokens, we propose introducing a relative position vector to encapsulate the feature representation of the interconnecting edge, as illustrated in Figure 1. Mathematically, the relative position between the input variables x_i and x_j can be characterized as

$$a_{ij}^K = w_{\text{clip}(j-i,k)}^K \tag{3}$$

$$a_{ij}^V = w_{\text{clip}(j-i,k)}^V \tag{4}$$

$$\text{clip}(x, k) = \max(-k, \min(k, x)) \tag{5}$$

where a_{ij}^K and a_{ij}^V are the relative position variables added to the Key and Value matrices, respectively. These relative representations depend only on the distance between tokens, independently of their positions in the sequence. The clip function limits the interaction between markers that are excessively distant, thus handling exceptionally long sequences effectively. The computation of multi-head attention, incorporating relative positional embedding, proceeds as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{Q\left(K^T + A^K\right)}{\sqrt{d_k}}\right)\left(V + A^V\right) \tag{6}$$

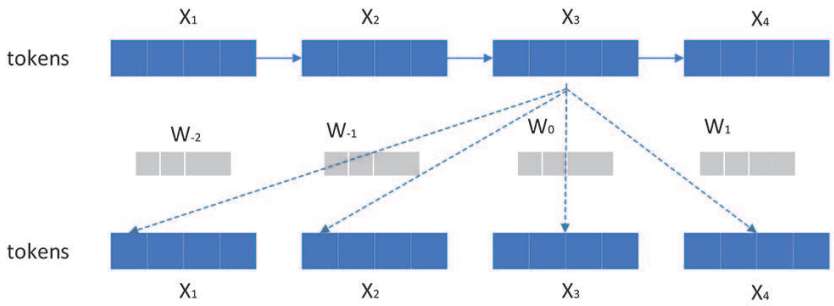


FIGURE 1. Visual representation of relative position representations

After computing the single-head attention, we employ a Squeeze-and-Excitation (SE) block [19] to adaptively recalibrate the channel-wise feature responses by explicitly modeling the interdependencies between channels. This operation first "squeezes" the global spatial information into a channel descriptor and then "excites" each channel by weighting its importance. The process is outlined as follows.

First, we perform global average pooling to squeeze the spatial dimensions, producing a channel-wise descriptor S, where X represents the input feature maps reshaped to (batch_size, channels, seq_len):

$$S = \text{F.adaptive\_avg\_pool1d}\left(X^T, 1\right) \tag{7}$$

This generates a tensor S of shape (batch_size, channels), where each channel is represented by a single scalar value summarizing its global information.

Next, we pass the descriptor S through two fully connected layers to form the excitation operation. The first fully connected layer reduces the channel dimensionality by a factor of r, and the second layer restores it to the original size. Between these layers, we apply the ReLU activation function to introducing non-linearity, and the Sigmoid activation function at the end scales the channel-wise importance to the range of $[0, 1]$:

$$E = \text{Sigmoid}(\text{fc2}(\text{ReLU}(\text{fc1}(S)))) \tag{8}$$

Here, E is the excitation vector with dimensions (batch_size, channels, 1), representing the learned importance of each channel.

Finally, we recalibrate the original feature maps X by multiplying each channel's feature map with its corresponding weight from the excitation vector E. The resulting output retains the same dimensions as the input feature map X:

$$\text{Channel\_Attention}(X) = X \times \text{Sigmoid}(E) \tag{9}$$

This channel-wise scaling ensures that the network emphasizes informative features while suppressing less useful ones. The SE block thus enhances the representational power of the network by adaptively adjusting the feature responses based on their global importance.

Finally, during the pre-training process, we use the AdamW algorithm as the base optimizer and apply the SAM strategy, which is designed to address the challenge of sharp minimizers, which are often associated with poor generalization.

SAM modifies the optimization process to consider the sharpness of the loss landscape around the current parameter point. The core idea is to minimize not only the loss function but also the sharpness of the minimizer. This is achieved by perturbing the current parameters in the direction that increases the loss the most and then minimizing the perturbed loss. Formally, the SAM optimization problem can be expressed as

$$\theta^* = \arg\min_{\theta} \left[\max_{\|\Delta\theta\| \leq \rho} \mathcal{L}(\theta + \Delta\theta) \right] \tag{10}$$

where θ denotes the model parameters, $\mathcal{L}(\cdot)$ represents the loss function, and ρ is a hyperparameter controlling the radius of the perturbation. The inner maximization finds the worst-case loss within a neighborhood around θ, while the outer minimization optimizes over the parameter space to reduce the worst-case loss. By optimizing this robust objective, SAM aims to obtain parameter settings that are less sensitive to perturbations, leading to better generalization.

4. **LETformer.** The proposed approach in this paper is called LETformer, which adopts relative positional embedding and simplified channel-wise attention to better suit high-concurrency traffic traces. In Figure 2, the architecture of the proposed LETformer network is shown in detail. It primarily consists of three modules: 1) converting the raw encrypted traffic into fine-grained token representations and performing multi-dimensional embeddings to learn deep associative features; 2) executing BURST Masked Prediction and Same-origin BURST Prediction tasks on massive pre-training data within the encoder consisting of six Channel-Wise attention blocks; 3) leveraging attention mechanisms and GRU to construct a classifier for downstream tasks.

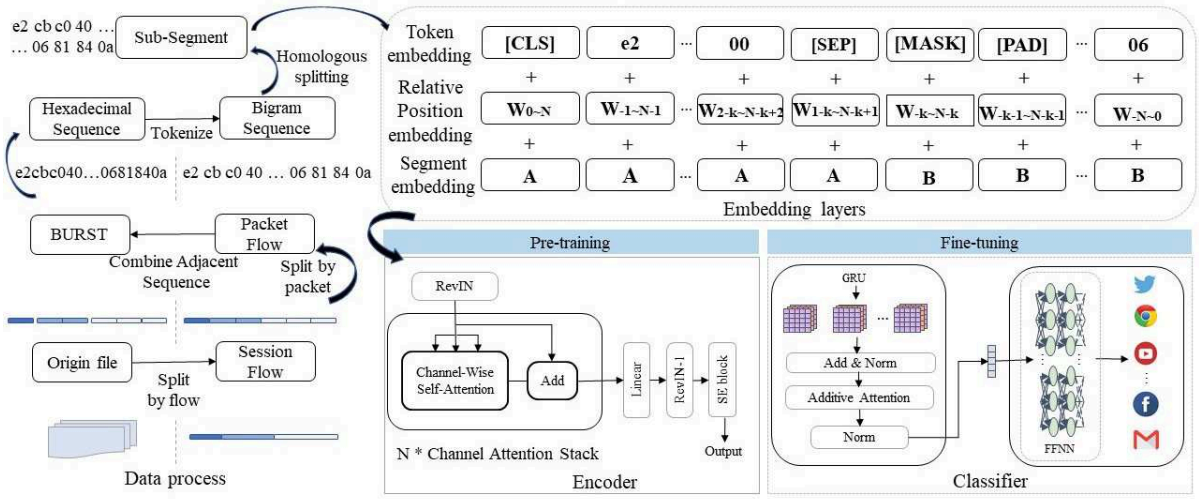


FIGURE 2. Overview of LETformer framework

4.1. **Input traffic representation.** Similar to ET-BERT [12], we first extract a set of sessions from the raw pcap files and then partition the homologous sequences into packet-level BURST (BERT-based Unified Representation for Scalable Traffic). Divergently, we set the token granularity to 2^2 during sequence tokenization. This is because, with a standard hidden layer size of 768, constructing a vocabulary from "0000" to "$ffff$" results in an embedding parameter size of 50 million. By reducing the data granularity of

the smallest unit, this part of the parameter can be completely ignored to reduce the high requirements for VRAM (Video Random Access Memory) and boost training efficiency. Three types of embedding layers are applied to the input sequence in the embedding layers to achieve light BURST vectorization:

- **Token Embedding**: The token representation learned from the lookup table, ranging from "`00`" to "`ff`", is set to a final hidden vector of 768 dimensions to match the input size of the encoder.
- **Relative Position embedding**: In addition to annotating the specific position of each token within the sequence, we employed relative positional embeddings to capture the relative positional relationships in encrypted messages. The relative distance between each pair of elements is represented by an $N \times N$ matrix, where N denotes the sequence length.
- **Segment embedding**: To achieve the Next Sentence Prediction objective, each input packet sequence is assigned segment identifiers.

4.2. **Traffic Transformer encoder for pre-training.** During pre-training, the embedded sequence first undergoes Reversible Instance Normalization (RevIN [21]), followed by a stack of N channel-wise attention layers, where the number of channels is simplified to match the sequence length. The size of each attention hidden layer remains at 768, as in the standard Transformer architecture, and the reduction ratio of the channel attention is set to 6. The pre-training process employs SAM (Sharpness-Aware Minimization) to optimize the model, guiding it towards a flatter local minimum. The overall structure of the pre-trained encoder is illustrated in Table 2. In this setup, the maximum distance L relative to attention capture can be dynamically adjusted; in our method, it is set to 128.

4.3. **LAM classifier for fine-tuning.** We also observed that some peer works improved downstream task classifiers without altering the pre-trained models, demonstrating that capturing long-distance dependencies can enhance classification performance [17, 18]. Therefore, we adopted the fine-tuning approach from Lamber, incorporating a BiGRU-based long-term byte sequence modeling module and attention mechanism (hereinafter referred to as LAM) for fine-tuning on downstream tasks. Finally, a fully connected layer is used for softmax probability prediction.

5. **Experiments and Results Analysis.** In this section, we gain a thorough understanding of LETformer's performance and efficiency in encrypted traffic classification through a multi-faceted evaluation. First, we describe the experimental setup, covering data preparation, metrics, implementation details. We then compare LETformer with several competitive baselines, including both state-of-the-art pretrained models and non-pretrained methods in traffic classification. Finally, we conduct ablation studies to further examine the effect of key components on the model's performance.

5.1. **Experiment settings.**

5.1.1. *Datasets preparation.* In the experiment, we used four types of public encrypted traffic data: ISCX-VPN2016 [22], USTC-TFC2016 [23], ICCXTOR [24], and CICIOT2022 [25]. For all pre-training tasks, we utilized around 30GB of data, the first three datasets included, while the CICIOT2022 dataset was only used for downstream tasks to test the transferability of different methods. In the downstream tasks, a limit of 5,000 samples is set for each label in each dataset. All datasets are split in an $8:1:1$ ratio. To prevent bias and interference, our method excludes the IP header and Ethernet header. Detailed statistical information is provided in the following Table 3.

TABLE 2. Network architecture of the proposed model in pre-training

LETformer				
Component	**Type**	**Input Dim**	**Output Dim**	**Activation**
Embedding Layer	WordPosSegRelPosEmbedding	128	768	–
Embedding Details				
Word Embedding	Embedding	261	768	–
Position Embedding	Embedding	512	768	–
Segment Embedding	Embedding	3	768	–
Relative Position Embedding	RelativePositionEmbedding	L	768	–
Dropout	Dropout	–	–	$p = 0.1$
Layer Norm	LayerNorm	–	768	–
Encoder				
Transformer Layer	SAMTransformerLayer	6 Layers	768	–
Transformer Details				
Compute Keys	Linear	$768 \to 768$	768	–
Compute Queries	Linear	$768 \to 768$	768	–
Compute Values	Linear	$768 \to 768$	768	–
Linear Forecaster	Linear	$768 \to 768$	768	–
Layer Norm 1	LayerNorm	–	768	–
Layer Norm 2	LayerNorm	–	768	–
RevIN	RevIN	–	768	–
Channel Attention FC1	Linear	$768 \to 128$	128	–
Channel Attention FC2	Linear	$128 \to 768$	768	–
Channel Attention Sigmoid	Sigmoid	–	768	–
Target				
MLM Linear 1	Linear	$768 \to 768$	768	–
MLM Linear 2	Linear	$768 \to 261$	261	–
Softmax	LogSoftmax	–	261	–
NSP Linear 1	Linear	$768 \to 768$	768	–
NSP Linear 2	Linear	$768 \to 2$	2	–

TABLE 3. Statistics of the four datasets in downstream tasks

Dataset	Label	Packet count	Encryption algorithm
USTC-TFC2016	20	97,115	AES RC4 3DES
ISCX-VPN2016	14	64,240	AES
ICCXTOR	8	40,000	AES RSA
CICIOT2022	7	35,000	AES AES-128 ECC

5.1.2. *Implementation details.* For fairness, we set the batch size to 64 and the training steps to 15,000 for all pre-training methods. We use the AdamW optimizer as the base optimizer for the SAM optimization strategy, initializing the learning rate at 2×10^{-5} and dynamically adjusting it during training. The dropout rate is set to 0.5, and the ratio of warmup is 0.1. All experiments are conducted on two A100 GPUs.

5.1.3. *Evaluation metrics.* To evaluate the performance of encrypted traffic classification, we use the following metrics.

- **Precision (PR)**: Precision measures the proportion of true positive results among the total number of positive predictions. It is defined as

$$\mathrm{PR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

where TP is the number of true positives and FP is the number of false positives.

- **Recall (RC)**: Recall, also known as sensitivity, measures the proportion of true positive results among the total number of actual positives. It is defined as

$$\mathrm{RC} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

where FN is the number of false negatives.

- **Accuracy (AC)**: Accuracy measures the proportion of correctly classified instances out of the total number of instances. It is defined as

$$\mathrm{AC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

where TN is the number of true negatives.

- **F1 Score (F1)**: The F1 Score is the harmonic mean of precision and recall, providing a single metric that balances both concerns. It is defined as

$$\mathrm{F1} = 2 \cdot \frac{\mathrm{PR} \cdot \mathrm{RC}}{\mathrm{PR} + \mathrm{RC}}$$

5.2. **Comparison with state-of-the-art methods.** To comprehensively evaluate our method, LETformer, we compare it with two deep learning-based baselines and two state-of-the-art attention-based pretrained methods. The details of these methods are listed below.

- **Deep Learning-Based Baselines:**
  - **FS-Net** [26] uses a flow sequence encoder to capture the temporal information in origin traffic data.
  - **Deep Packet** [27] first utilizes a Stacked Autoencoder (SAE) and Convolutional Neural Network (CNN) to extract deep features for flow characterization.
- **State-of-the-Art Attention-Based Pretrained Methods:**
  - **YATC** [28] transforms the raw packets into grayscale images during preprocessing.
  - **ET-BERT** [12] constructs context prediction and token prediction tasks similar to text sequences within the original BERT architecture.

As is shown in Table 4, non-pretrained deep learning methods are susceptible to the challenges of data imbalance, leading to significant variations in classification performance across different datasets. Despite using less than one-tenth of the parameter size, LETformer maintains a precision loss of less than 0.02 on the USTC-TFC2016 and ISCX-VPN2016 datasets. It even achieves the best performance on the ICCXTOR dataset, which has the smallest sample size. However, it is noteworthy that LETformer does not perform as well on the CICIOT2022 dataset, which was not included in the pre-training data. This may be due to the large-scale contraction of the vocabulary size, which impacts the learning of transferable complicated semantic information in long sequences. In contrast, YATC, which employs image-like preprocessing methods, shows superior transferability, indicating that methods not reliant on token mapping offer better transferability.

Table 4. Comparison of LETformer with other state-of-the-art methods

	USTC-TFC2016				ISCX-VPN2016				ICCXTOR				CICIOT2022			
	F1	RC	AC	PR	F1	RC	AC	PR	F1	RC	AC	PR	F1	RC	AC	PR
FS-Net	0.884	0.892	0.8637	0.8846	0.8632	0.8557	0.8664	0.8602	0.5362	0.4592	0.6074	0.608	0.8562	0.8468	0.8637	0.8544
Deep Packet	0.8883	0.8788	0.8849	0.8786	0.7881	0.6102	0.7883	0.6006	0.2704	0.3476	0.3682	0.3268	0.8808	0.8604	0.8808	0.8626
YATC	0.9607	0.9607	0.9613	0.9623	0.8807	0.8879	0.8879	0.8879	0.8234	0.8104	0.8104	0.8231	0.9201	**0.9210**	0.9201	**0.9270**
ET-BERT	**0.9905**	**0.9904**	**0.9904**	**0.9905**	**0.9934**	**0.9933**	**0.9946**	**0.9935**	0.9997	0.9998	0.9998	0.9998	**0.9209**	0.9209	**0.9209**	0.9215
LETformer (ours)	0.9871	0.9870	0.9869	0.9872	0.9806	0.9837	0.9892	0.9778	**1.000**	**1.000**	**1.000**	**1.000**	0.8866	0.8868	0.8868	0.8867

5.3. **Ablation study.** We present ablation results to validate the effectiveness of different components in our model design, as shown in Table 5. The results from the ablation experiments indicate that, regardless of whether the downstream task data is included in the pre-training process, both the Lambert classifier and the relative position embeddings contribute to improving the final classification accuracy. SAM does not directly improve the metrics of the USTC-TFC2016 dataset in downstream tasks; rather, its role is more focused on mitigating overfitting caused by imbalanced data distribution during the pre-training phase. We recorded the loss values of LETformer during pre-training with and without the SAM strategy across two pre-training tasks to substantiate this point as Figure 3. When the SAM algorithm is employed, the model's initial data fitting speed slows down. However, by avoiding local optima during training, the model ultimately achieves better data fitting on both tasks, learning a more comprehensive data distribution. Consequently, it yields improved performance on the CICIOT2022 dataset, which is a transfer-learning task.

Table 5. Ablation study of key components on USTC-TFC2016 and CICIOT2022

	Index			USTC-TFC2016				CICIOT2022			
	RL	LAM	SAM	F1	RC	AC	PR	F1	RC	AC	PR
LETformer (full)	✓	✓	✓	**0.9871**	0.9870	**0.9869**	**0.9872**	**0.8866**	**0.8868**	**0.8868**	**0.8867**
w/o RL	✗	✓	✓	0.9857	0.9854	0.9854	0.9858	0.8825	0.8823	0.8823	0.8833
w/o LAM	✓	✗	✓	0.9848	0.9849	0.9846	0.9848	0.8823	0.8825	0.8823	0.8833
w/o SAM	✓	✓	✗	0.9870	**0.9870**	**0.9871**	**0.9872**	0.8849	0.8851	0.8849	0.8859

(✓) indicates the inclusion of the component, while (✗) indicates its exclusion. RL represents the use of relative attention positional encoding in pre-training. LAM refers to the use of the Lambert structure during fine-tuning. When removed, it is replaced with a fully connected (FC) layer. SAM indicates whether the Sharpness-Aware Minimization algorithm is used during pre-training to prevent overfitting.

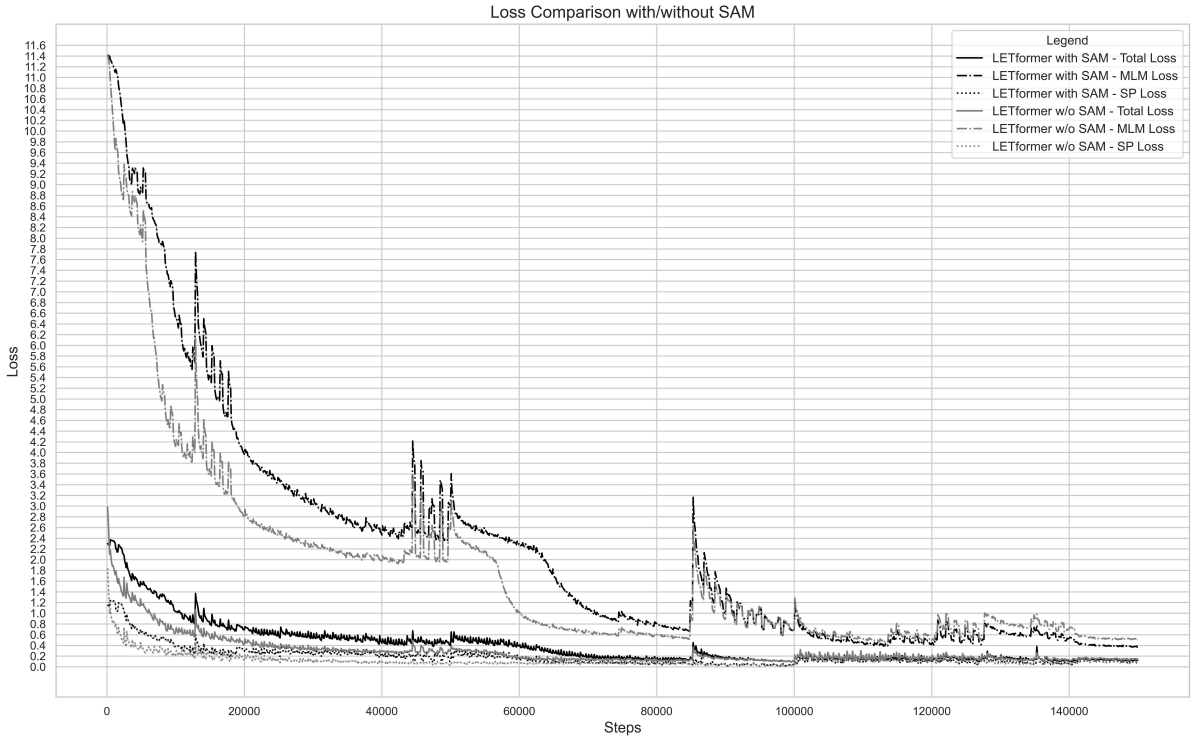


Figure 3. Training loss changes with and without the SAM strategy

6. **Conclusions.** In this paper, we proposed LETformer, a novel approach for encrypted traffic classification. Our study focused on leveraging advanced lightweight attention mechanisms while maintaining accuracy. We achieved several key results that significantly contribute to the field of efficient network traffic classification, particularly in scenarios with limited computational resources. Specifically, we opted to reduce the granularity of traffic sequence partitioning to eliminate the extensive vocabulary parameters. We utilized complete relative attention to capture the temporal relationships between traffic sequences and designed a lightweight encoder based on channel attention, which constrained the model parameters to 17.6 million while maintaining classification performance consistent with state-of-the-art methods of similar types. These modifications make LETformer more suitable for deployment in computationally constrained environments such as mobile devices. In the future, we consider combining preprocessing methods such as visualization to construct multi-modal feature extraction to enhance the accuracy of LETformer and improve its performance in downstream migration tasks.

Acknowledgment. This work was supported by the Sichuan Science and Technology Program (2023YFG0146) and the Stability Program of National Key Laboratory of Security Communication (M3023Y327).

REFERENCES

[1] E. Papadogiannaki and S. Ioannidis, A survey on encrypted network traffic analysis applications, techniques, and countermeasures, *ACM Computing Surveys (CSUR)*, vol.54, no.6, pp.1-35, 2021.

[2] J.-Y. Zhao, J. Gong, S.-T. Ma and Z.-M. Lu, Curvature gray feature decomposition based finger vein recognition with an improved convolutional neural network, *International Journal of Innovative Computing, Information and Control*, vol.16, no.1, pp.77-90, 2020.

[3] B. Richardson and A. Wicaksana, Comparison of IndoBERT-lite and RoBERTa in text mining for Indonesian language question answering application, *International Journal of Innovative Computing, Information and Control*, vol.18, no.6, pp.1719-1734, 2022.

[4] P. Foret, A. Kleiner, H. Mobahi and B. Neyshabur, Sharpness-aware minimization for efficiently improving generalization, *arXiv Preprint*, arXiv: 2010.01412, 2020.

[5] A. Candra, Wella and A. Wicaksana, Bidirectional encoder representations from transformers for cyberbullying text detection in Indonesian social media, *International Journal of Innovative Computing, Information and Control*, vol.17, no.5, pp.1599-1615, 2021.

[6] A. Vlăduţu, D. Comăneci and C. Dobre, Internet traffic classification based on flows' statistical properties with machine learning, *International Journal of Network Management*, vol.27, no.3, e1929, 2017.

[7] W. Song, M. Beshley, K. Przystupa, H. Beshley, O. Kochan, A. Pryslupskyi, D. Pieniak and J. Su, A software deep packet inspection system for network traffic analysis and anomaly detection, *Sensors*, vol.20, no.6, 1637, 2020.

[8] T. Karagiannis, K. Papagiannaki and M. Faloutsos, BLINC: Multilevel traffic classification in the dark, *Proc. of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp.229-240, 2005.

[9] V. F. Taylor, R. Spolaor, M. Conti and I. Martinovic, Robust smartphone app identification via encrypted network traffic analysis, *IEEE Transactions on Information Forensics and Security*, vol.13, no.1, pp.63-78, 2017.

[10] G. Liu, G. Yang, S. Bai, H. Wang and Y. Xiang, FASE: A fast and accurate privacy-preserving multi-keyword top-k retrieval scheme over encrypted cloud data, *IEEE Transactions on Services Computing*, vol.15, no.4, pp.1855-1867, 2020.

[11] S. Yin, H. Li, L. Teng, A. A. Laghari and V. V. Estrela, Attribute-based multiparty searchable encryption model for privacy protection of text data, *Multimedia Tools and Applications*, pp.1-22, 2023.

[12] X. Lin, G. Xiong, G. Gou, Z. Li, J. Shi and J. Yu, ET-BERT: A contextualized datagram representation with pre-training transformers for encrypted traffic classification, *Proc. of the ACM Web Conference 2022*, pp.633-642, 2022.

[13] Z. Diao, G. Xie, X. Wang, R. Ren, X. Meng, G. Zhang, K. Xie and M. Qiao, EC-GCN: A encrypted traffic classification framework based on multi-scale graph convolution networks, *Computer Networks*, vol.224, 109614, 2023.
[14] S. Zhu, X. Xu, H. Gao and F. Xiao, CMTSNN: A deep learning model for multi-classification of abnormal and encrypted traffic of Internet of Things, *IEEE Internet of Things Journal*, 2023.
[15] Q. Zhou, L. Wang, H. Zhu, T. Lu and V. S. Sheng, WF-Transformer: Learning temporal features for accurate anonymous traffic identification by using transformer networks, *IEEE Transactions on Information Forensics and Security*, 2023.
[16] P. Shaw, J. Uszkoreit and A. Vaswani, Self-attention with relative position representations, *arXiv Preprint*, arXiv: 1803.02155, 2018.
[17] X. Ma, T. Liu, N. Hu and X. Liu, Bi-ETC: A bidirectional encrypted traffic classification model based on BERT and BiLSTM, *2023 8th International Conference on Data Science in Cyberspace (DSC)*, pp.197-204, 2023.
[18] T. Liu, X. Ma, L. Liu, X. Liu, Y. Zhao, N. Hu and K. Z. Ghafoor, LAMBERT: Leveraging attention mechanisms to improve the BERT fine-tuning model for encrypted traffic classification, *Mathematics*, vol.12, no.11, 1624, 2024.
[19] J. Hu, L. Shen and G. Sun, Squeeze-and-excitation networks, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.7132-7141, 2018.
[20] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, *arXiv Preprint*, arXiv: 1810.04805, 2018.
[21] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi and J. Choo, Reversible instance normalization for accurate time-series forecasting against distribution shift, *International Conference on Learning Representations*, 2021.
[22] G. D. Gil, A. H. Lashkari, M. Mamun and A. A. Ghorbani, Characterization of encrypted and VPN traffic using time-related features, *Proc. of the 2nd International Conference on Information Systems Security and Privacy (ICISSP)*, Setúbal, Portugal, pp.407-414, 2016.
[23] W. Wang, M. Zhu, X. Zeng, X. Ye and Y. Sheng, Malware traffic classification using convolutional neural network for representation learning, *2017 International Conference on Information Networking (ICOIN)*, pp.712-717, 2017.
[24] A. H. Lashkari, G. D. Gil, M. S. I. Mamun and A. A. Ghorbani, Characterization of tor traffic using time based features, *International Conference on Information Systems Security and Privacy*, vol.2, pp.253-262, 2017.
[25] S. Dadkhah, H. Mahdikhani, P. K. Danso, A. Zohourian, K. A. Truong and A. A. Ghorbani, Towards the development of a realistic multidimensional IoT profiling dataset, *2022 19th Annual International Conference on Privacy, Security & Trust (PST)*, pp.1-11, 2022.
[26] C. Liu, L. He, G. Xiong, Z. Cao and Z. Li, FS-Net: A flow sequence network for encrypted traffic classification, *IEEE Conference on Computer Communications (IEEE INFOCOM 2019)*, Paris, France, pp.1171-1179, 2019.
[27] M. Lotfollahi, M. J. Siavoshani, R. S. H. Zade and M. Saberian, Deep Packet: A novel approach for encrypted traffic classification using deep learning, *Soft Computing*, vol.24, no.3, pp.1999-2012, 2020.
[28] R. Zhao, M. Zhan, X. Deng, Y. Wang, Y. Wang, G. Gui and Z. Xue, Yet another traffic classifier: A masked autoencoder based traffic transformer with multi-level flow representation, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol.37, no.4, pp.5420-5427, 2023.

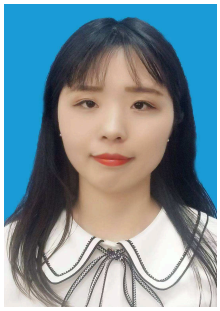
Author Biography

Zhiyan Meng received the academic degree in IoT Engineering from Southwest University for Nationalities, China, in 2022; currently pursuing a master's degree in Computer Science and Technology at the University of Electronic Science and Technology of China.

His research interests include artificial intelligence, cybersecurity, and large language models.

Dan Liu received the B.Sc. degree in Radio Engineering from Southeast University, Nanjing, China, in July 1990; the M.Sc. degree in Computer Applications from Sichuan University, China, in July 1995; and the Ph.D. degree in Computer Applications from the University of Electronic Science and Technology of China in November 2005. He was appointed as an Associate Professor in 2006. He is currently a graduate supervisor of Research Institute of Electronic Science and Technology, University of Electronic Science and Technology of China.

Dr. Liu's research focuses on computer system architecture and high-performance computing. He has published over ten papers related to information security in major domestic and international journals and conferences.



Jintao Meng is currently a researcher at the National Key Laboratory of Security Communication. She received her B.Sc. degree in Mathematics from Beijing Technology and Business University in 2014; her M.Sc. degree in Mathematics from the University of Science & Technology Beijing (USTB) in 2016; and her Ph.D. degree in Engineering from USTB in 2020.

Her research focuses on machine learning and cyber security.