

Article

LAMBERT: Leveraging Attention Mechanisms to Improve the BERT Fine-Tuning Model for Encrypted Traffic Classification

Tao Liu ¹, Xiting Ma ¹, Ling Liu ¹, Xin Liu ², Yue Zhao ³ , Ning Hu ^{1,*}  and Kayhan Zrar Ghafoor ⁴ 

¹ Institute of Cyberspace Security, Guangzhou University, Guangzhou 510006, China; 2112233168@e.gzhu.edu.cn (T.L.); 2112233031@e.gzhu.edu.cn (X.M.); 2112106034@e.gzhu.edu.cn (L.L.)

² College of Computer Engineering and Applied Math, Changsha University, Changsha 410022, China; xinliu@ccsu.edu.cn

³ Science and Technology on Communication Security Laboratory, Chengdu 610041, China; yuezhaoy@foxmail.com

⁴ Department of Computer Science, Knowledge University, Erbil 44001, Iraq; kayhan.ghafoor@epu.edu.iq

* Correspondence: huning@gzhu.edu.cn; Tel.: +86-138-7311-2138

Abstract: Encrypted traffic classification is a crucial part of privacy-preserving research. With the great success of artificial intelligence technology in fields such as image recognition and natural language processing, how to classify encrypted traffic based on AI technology has become an attractive topic in information security. With good generalization ability and high training accuracy, pre-training-based encrypted traffic classification methods have become the first option. The accuracy of this type of method depends highly on the fine-tuning model. However, it is a challenge for existing fine-tuned models to effectively integrate the representation of packet and byte features extracted via pre-training. A novel fine-tuning model, LAMBERT, is proposed in this article. By introducing an attention mechanism to capture the relationship between BiGRU and byte sequences, LAMBERT not only effectively improves the sequence loss phenomenon of BiGRU but also improves the processing performance of encrypted stream classification. LAMBERT can quickly and accurately classify multiple types of encrypted traffic. The experimental results show that our model performs well on datasets with uneven sample distribution, no pre-training, and large sample classification. LAMBERT was tested on four datasets, namely, ISCX-VPN-Service, ISCX-VPN-APP, USTC-TFC and CSTNET-TLS 1.3, and the F1 scores reached 99.15%, 99.52%, 99.30%, and 97.41%, respectively.

Keywords: information security; encrypted traffic classification; privacy protection; fine-tuning model; attention mechanism

MSC: 68P25



Citation: Liu, T.; Ma, X.; Liu, L.; Liu, X.; Zhao, Y.; Hu, N.; Ghafoor, K.Z. LAMBERT: Leveraging Attention Mechanisms to Improve the BERT Fine-Tuning Model for Encrypted Traffic Classification. *Mathematics* **2024**, *12*, 1624. <https://doi.org/10.3390/math12111624>

Academic Editor: Antanas Cenys

Received: 29 March 2024

Revised: 6 May 2024

Accepted: 11 May 2024

Published: 22 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the rapid development of internet technology and increasing attention given to data privacy [1], the proportion of encrypted traffic in internet traffic is constantly increasing. Google's recent report shows that 96% of chrome web pages have SSL/TLS encryption enabled as of February 2024 [2]. Encryption is an important means of protecting privacy [3], protecting users' network traffic data from prying eyes, as well as preventing eavesdroppers from stealing users' passwords or application usage habits [4]. Because traffic encryption technology hides the data payload of the application protocol, traditional traffic classification technology based on data payload cannot analyse the encrypted field. While providing confidentiality for information systems, privacy protection solutions based on encryption technology also bring greater technical challenges to operators and security management departments, such as malicious app governance [5], and network attack behaviour detection [6]. Encrypted traffic classification is an important technology for supporting network management and behaviour analysis [7]. For real-world encrypted

traffic classification applications, classification can involve different scenarios, often depending on the intent of the network administrator. The classification of encrypted traffic can distinguish network traffic, user behaviour, and application services in the network environment [8]; detect whether illegal software encodes sensitive information through the encrypted channel [9]; help operators optimize network resource scheduling policies [10]; and assist network security departments to conduct network behaviour audits.

Encrypted traffic classification is an important part of information system privacy protection research. It can achieve highly accurate classification results without decrypting ciphertext [11,12]. Early encrypted traffic analysis techniques constructed packet fingerprints by matching the readable information of IP packets [13], and classifying encrypted traffic via statistical methods. With the development of privacy-enhanced encryption [14], the visible fields of IP packets can be disguised by technologies such as VPN and Tor, so they gradually lost their relevance and even become deceptive. Moreover, the encryption protocols (e.g., TLS 1.3) are constantly changing, and the amount of visible packet header information is decreasing [15]. Many scholars have used machine learning methods to extract and classify the statistical features (packet size, average packet arrival time, etc.) of encrypted traffic [11,16]. Compared with traditional statistical methods, machine learning can process large-scale and high-dimensional data. However, the feature extraction of machine learning requires professional knowledge and experience in the field as well as high human resource costs. Moreover, its limited generalization ability cannot be effectively applied to analysing complex encrypted traffic. For example, when the number of training samples is reduced by 80%, the classification accuracy of Van Ede et al.'s model decreases by 62.48% [17]. In response, many scholars have turned their research direction to deep learning [18–21]. Different from traditional machine learning methods, deep learning does not need to manually select network traffic feature sets and instead can directly use raw traffic as the input data and automatically learn the nonlinear relationship between the raw traffic input and expected output labels [22]. Such methods have achieved significant performance improvement. However, they require a large number of labelled datasets for training and cannot adapt well to emerging encrypted traffic types and classification scenarios. For these kinds of methods, when the training samples are reduced by 90%, the classification accuracy—for example, of Lotfollahi et al.'s model—decreases by 40.22% [17].

In recent years, pretrained models have made significant breakthroughs in natural language processing (NLP) [23], computer vision (CV) [24], and other fields, highlighting their great potential in artificial intelligence research and application. pre-training methods are also widely used in network communication scenarios [17,25,26]. The pre-training based encrypted traffic method is divided into two stages. In the first stage, the pretrained model autonomously learns the nonlinear relationship between encrypted packets and bytes in large-scale unlabelled encrypted traffic data and extracts the features of packets and bytes. In the second stage, the weight parameters learned by pre-training are used to initialize the fine-tuning model, and a small amount of labelled data is used to fine-tune the model parameters so that the model can be quickly and accurately applied to the classification target in various encryption scenarios. However, the rationality of the application of such models in the field of encrypted traffic classification still needs to be studied.

In this paper, we propose LAMBERT, an improved fine-tuning model based on bidirectional encoder representations from Transformers (BERT), and apply it to encrypted traffic classification. LAMBERT uses an attention mechanism to optimize long-term byte dependencies to classify encrypted traffic. The method includes several key steps. First, the packet feature extraction module is used to extract the features of the token sequence and convert them into the corresponding feature vectors. Second, the bidirectional gated recurrent unit (BiGRU) of the long-term byte sequence modelling module is used to model the context relationship of the feature vector sequence to capture the long-distance relationships between bytes. Then, the key enhancement module, through the self-attention mechanism, maps the weights and learns the parameter matrix to enhance the influence of key information to improve the loss of byte feature sequences to more accurately capture

the implicit features between long-distance byte sequences and improve the accuracy and generalization ability of the classification. Finally, the classification output module is used to complete the final encrypted traffic classification task. Our proposed classification model is not limited to any particular type of encrypted data and can handle encryption algorithms (such as AES and CHACHA20), encryption protocols (such as SSL and TLS), and encryption technologies (such as VPN and Tor). This is due to the fact that our model relies only on the general characteristics of encrypted traffic, namely, the transport layer payload, to classify it thus making it applicable to various types of encrypted communication without being limited by the underlying encryption details.

Our main contributions are as follows:

1. We comprehensively analyse the feasibility of applying artificial intelligence techniques and BERT models to encrypted traffic classification and reveal the important impact of long-distance dependencies between byte sequences on the accuracy of encrypted traffic classification.
2. We propose a new encrypted traffic classification model, LAMBERT, which uses the long-term byte sequence modelling to model the long and short distances of the byte feature sequences extracted by BERT and introduces the multiattention of the key enhancement module to solve the problem of sequence loss between long-distance dependencies. Furthermore, additive attention is used to reduce the dimensionality of the features output by multiattention while retaining the original features.
3. We test the capabilities of the LAMBERT model by comparing it with ten other advanced models using three types of typical datasets. These data include imbalanced sample datasets (USTC-TFC and VPN-App), no pretrained datasets (USTC-TFC) and large classification datasets (CSTNET-TLS 1.3). The experimental results show that the LAMBERT model can achieve high accuracy on these datasets. We have published the source code of the LAMBERT model on GitHub (<https://github.com/MysteryObstacle/Lambert>, accessed on 29 March 2024).

The remainder of this paper is organized as follows. In Section 2, we review some of the most important and recent research on encrypted traffic classification. In Section 3, we introduce the paper's motivation and goals. Section 4 introduces our proposed approach, namely, LAMBERT. In Section 5, the experimental results are presented and analysed. Section 6 discusses the limitations of LAMBERT. Finally, we conclude the paper in Section 7.

2. Related Work

The current research methods for encrypted traffic classification can be divided into three main categories: machine learning, deep learning and pre-training. At present, many researchers use BERT pre-training technology for encrypted traffic classification, and our method mainly focuses on the fine-tuning stage of pre-training. However, many research works based on machine learning and deep learning are still representative in terms of ideas and practical applications. Therefore, we select representative works from these two categories to introduce and compare them with our method regarding performance in Section 5.2.

Based on machine learning: FlowPrint [27] is the first real-time system for building mobile application fingerprints. The framework deploys a semisupervised approach by clustering the target network and used cross-correlation methods to measure temporal correlations to fingerprints and classify the traffic of visible and invisible mobile applications. AppScanner [16] is a powerful and scalable framework for identifying smartphone apps from network traffic. We comprehensively evaluate the feasibility of smartphone app fingerprinting from several aspects, and the experimental results show that app fingerprinting is not significantly affected by the device on which the app is installed. BIND [28] is a novel encrypted network traffic data analysis method for end-node identification that exploits the correlation in packet sequences to extract features suitable for classification. It is suitable for website identification and fingerprint identification.

Based on deep learning: DF [29] is a new website fingerprinting attack against Tor. The architecture of the model is designed by using a CNN, and attacks against WTF-PADs and Walkie-Talkie are evaluated. FS-Net [19] is an end-to-end classification model that learns representative features from raw streams and then classifies them in a unified framework. Moreover, it adopts a multilayer codec structure to deeply mine the potential sequence features of flows. Deeppacket [21] adopts two deep neural network architectures (SAE and CNN) to classify network traffic, which can not only distinguish VPN and non-VPN network traffic but also identify end-user applications.

Based on pre-training: The current methods of applying pre-training technology to encrypted traffic classification mainly focus on two stages: pre-training and fine-tuning. For the pre-training stage, PERT [25] first introduces dynamic word embedding technology into pre-training and uses it for automatic traffic feature extraction. ET-BERT [17] first proposed two pre-training tasks, Mask BURST Prediction (MBM) and Homologous BURST Prediction (SBP), to help the pretrained model better learn the feature representations of packets and bytes. For the fine-tuning stage, BFCN [30] proposes the use of three 1D CNNs to capture the byte-level local features of BERT output to improve the expression ability of features. Bi-ETC [31] enhances the ability to capture long-distance dependencies between byte feature sequences by introducing BiLSTM after BERT. Moreover, by copying packet-level features to the tail of BiLSTM, the problem of forgetting packet-level features is mitigated.

3. Motivation and Objective

Encrypted Traffic Classification of Privacy-Preserving: Encrypted traffic classification aims to classify network traffic under the protection of various encryption technologies (such as VPN and TLS 1.3) and can be divided according to applications and network services. As shown in Figure 1, although the protocol header and the underlying network fields (such as port number and IP) are readable, with the development of privacy enhancement technology, these visible header fields may be deceptive. Therefore, with traditional traffic classification methods, it is difficult to identify encrypted traffic that is obfuscated by visible fields. Furthermore, encrypted traffic classification methods may violate users' privacy because they perform traffic detection based on constructing packet fingerprints in visible fields. To achieve privacy protection and improve classification accuracy, we choose to use the encrypted payload to classify encrypted traffic. However, is it possible to classify traffic with high accuracy without decrypting the ciphertext? We find that the ciphertext is not completely random, but it has implicit characteristics. Sengupta et al. [32] used the randomness difference between different ciphertexts to distinguish different applications, which showed that encrypted traffic is not completely random but that there is an implicit feature composed of the randomness difference between different ciphertexts. These implicit features can be used for the classification of encrypted traffic.

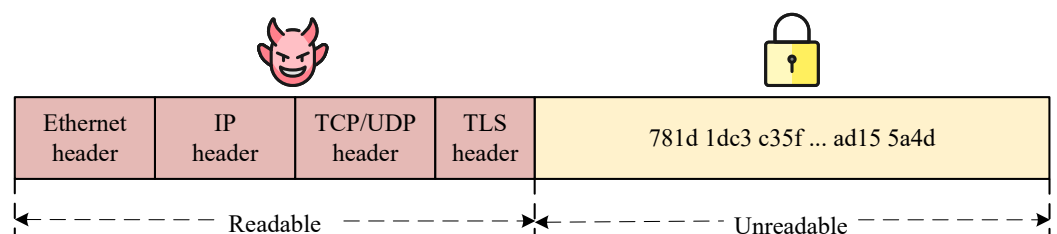


Figure 1. Schematic diagram of the composition of encrypted traffic packets.

Road to BERT: Due to the great success of BERT models in the field of natural language processing, we analyse the applicability of BERT models in the field of encrypted traffic classification. The ciphertext in encrypted traffic is encrypted data, which are unreadable and random [33], and have obvious differences from natural language in semantic structure. In natural language, words or characters are organized into sentences and paragraphs

according to grammatical rules and semantic meanings, while in cryptography, characters or bits in plaintext are converted into ciphertext according to a certain encryption algorithm. Since ciphertext is also a kind of sequence data and has implicit features, advanced pre-training models in the field of NLP, such as BERT, can be used to learn a universal representation of ciphertext [17,25]. Success has been achieved in this regard, and the well-performing pretrained models have been validated through comparative trials by analysing the differences in randomness in different ciphertexts.

Improved Fine-tuning: However, PERT, ET-BERT and other researchers directly use the original BERT model for fine-tuning and simply use the packet features extracted by BERT for classification while ignoring the importance of other byte features. Therefore, when faced with new encrypted traffic classification scenarios and objectives, existing fine-tuning models perform poorly when dealing with encrypted traffic with a large distribution difference between pre-training data and fine-tuning data (Section 5.6). The difficulty of current research is how to effectively integrate the implicit features between other byte feature sequences without overwriting the original packet features to improve the expression ability of the model. By comparing natural language with ciphertext, we find that many literal combinations are impossible in natural language due to the lack of actual semantics. However, the randomness of the encryption system makes any combination of bytes possible. Therefore, we believe that it is difficult to capture the implicit characteristics of ciphertext by analysing only the distribution of local bytes. In addition, secure cryptographic systems are characterized by certain desirable properties, namely, completeness [32]. Even a small change in the plaintext will affect all the bits of the ciphertext, which means that a change in the original text will be imposed on all the ciphertext bytes. Therefore, capturing long-distance dependencies between bytes is more helpful for accurate classification. This is also proven by the rationality analysis of long-distance dependencies between encrypted traffic bytes in Section 5.3. BFCN [30] attempt to capture local byte feature sequences through CNNs, but with CNNs, it is difficult to capture long-distance dependencies between byte feature sequences. Bi-ETC [31] realizes the importance of long-distance dependencies between bytes. However, when BiLSTM (bidirectional long short-term memory) is used to process byte feature sequences that are too long, sequence loss occurs, making it difficult for the model to model the structural information between sequence data [34]. As a result, the long-distance dependence between bytes is still difficult to model, and our experiments in Section 5.7 also prove that this problem has a serious impact on the classification accuracy (Figure 2).

Objective: The discussion of related research shows that the existing BERT-based fine-tuning models for encrypted traffic have shortcomings in capturing long-distance dependencies between bytes [35]. Through the above analysis, we find that the introduction of the long-term byte sequence modelling module and key enhancement module will improve the capture of long-distance dependencies. Based on this fine-tuned model, the main goal of this study is to achieve superior performance under various classification scenarios and objectives, outperforming other state-of-the-art models on different classification tasks (Section 5.2). In addition, compared with the original BERT fine-tuning model, we aim to address a variety of complex data distributions, including imbalanced datasets (Section 5.5), unpretrained datasets (Section 5.6), and datasets with large classification samples (Section 5.7). Our goal is to achieve high-accuracy encrypted traffic classification tasks in these cases to fully demonstrate the significant advantages of our model in terms of generalization performance and classification accuracy.

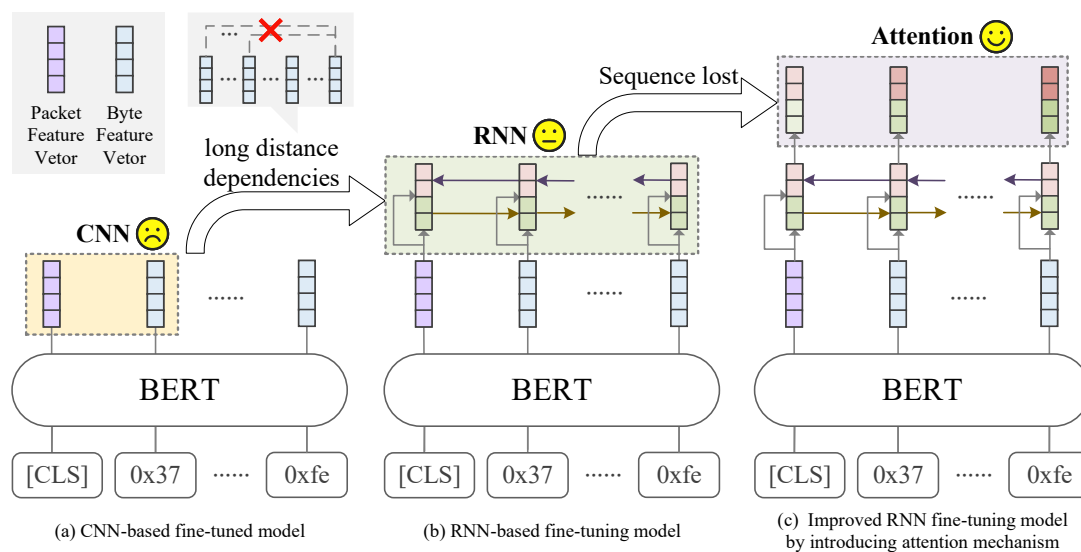


Figure 2. Three different fine-tuning ideas.

4. Methodology

4.1. Workflow

The pre-training-based encrypted traffic classification method is divided into three stages. The first is the data preprocessing stage, which aims to convert the original encrypted traffic data into unlabelled and labelled traffic data that can be input to the model for training. Next, there is a pre-training phase, in which a large amount of unlabelled traffic data are input into the pretrained model for learning to obtain a common representation of encrypted traffic. Finally, in the fine-tuning stage, the pretrained model weights are used to initialize the fine-tuning model, and a small amount of labelled traffic data is input for fine-tuning to train and generate a model that can be used to predict traffic categories. In this paper, we aim to further improve the accuracy and generalizability of encrypted traffic classification by innovating a fine-tuning process based on pretrained BERT (Figure 3).

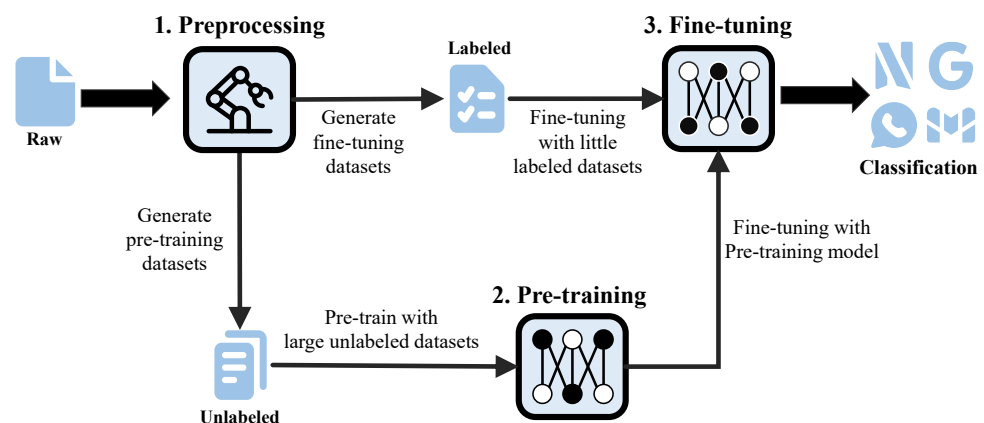


Figure 3. Flow chart of encrypted traffic classification based on pre-training.

4.2. Preprocessing

To improve the data quality and training efficiency, retransmission and out-of-order data packets as well as network query packets (e.g., ICMP and DNS) are filtered out, and only IP packets are retained. Because retransmissions and out-of-order packets are often caused by anomalous network transmissions, they can distort traffic patterns and affect classification accuracy [21]. By removing these packets, the traffic pattern in the training and test datasets can be ensured to be more accurate and consistent. In addition,

network query packets are usually caused by network requests such as domain name resolution, advertisement tracking, and statistical analysis, which are not related to the goal of traffic classification. Reserving only IP packets can reduce the size of the dataset and reduce unnecessary processing and analysis during training and testing, thereby improving efficiency.

Since LAMBERT does not rely on any visible plaintext information, only the transport layer payload of each packet is intercepted, and the remainder (e.g., Ethernet headers, IP headers, and transport layer headers) are discarded. The payload is then converted to hexadecimal encoding for processing to ensure data consistency and commonality and to avoid character encoding issues. The hexadecimal payload is then encoded using a bigram model to help capture the context between adjacent bytes.

4.3. Pre-Training Model

In the pre-training stage, the masked BURST model and same-origin BURST prediction are adopted as the pre-training methods. However, this is not the focus of our paper [17]. In the pre-training process, approximately 30 GB of unlabelled traffic data is used. This dataset contains two parts: (1) 15 GB of traffic from public datasets, including ISCX-VPN-Service, ISCX-VPN-App, CIC-IDS 2017 and 2018; and (2) 15 GB of traffic from the China Science and Technology Network (CSTNET-TLS 1.3).

4.4. Fine-Tuning Model

Recent studies have shown that the recurrent neural network (RNN) model can effectively capture the context between byte-level feature vectors output by the BERT model [31]. However, when dealing with longer sequences, BiLSTM, BiGRU and other recurrent neural networks have the problem of sequence loss [36]. To overcome this challenge, this study proposes an innovative fine-tuning model, which improves upon the original BERT fine-tuning model. Three attention mechanisms—self-attention, multihead attention, and additive attention [34]—are introduced after the improved long-term byte sequence modelling module structure based on BiGRU. This innovation helps to avoid the loss of key information when dealing with long sequences. It also enables the model to more accurately learn the long-distance interdependence between sequences. Furthermore, it solves the problem of attention weight dimensionality reduction classification to improve the performance of encrypted traffic classification [37] (Figure 4).

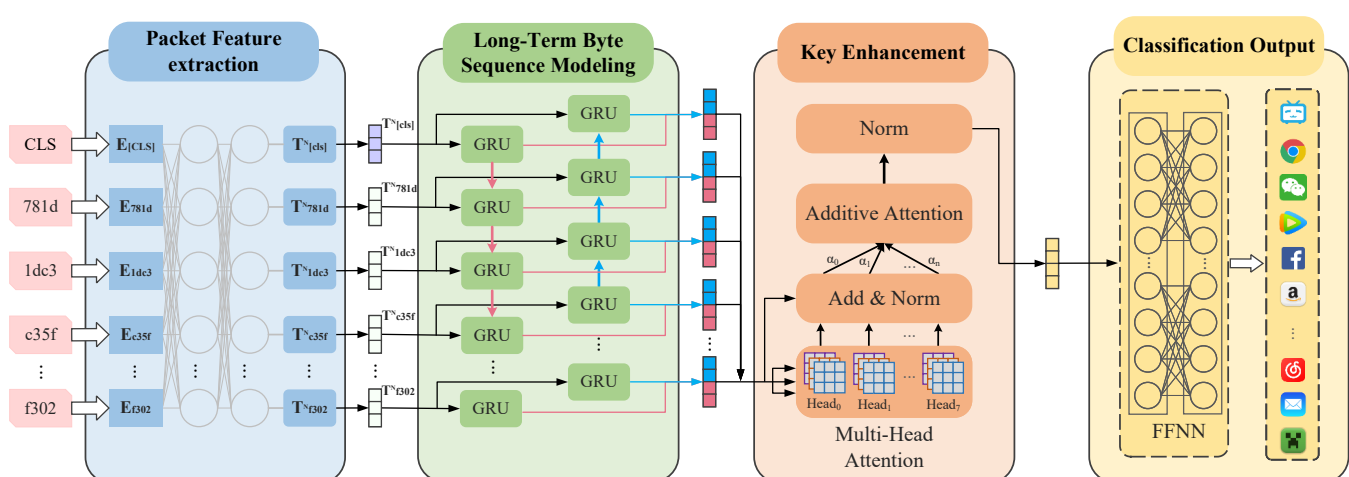


Figure 4. Structure diagram of the LAMBERT model.

4.4.1. Packet Feature Extraction Module

The preprocessed encrypted traffic byte sequence (Section 4.2) is input into the BERT model. BERT performs position embedding, segment embedding, and token embedding on these section sequences to convert them into byte-level token sequences [23]. The Token

sequence can be expressed as $T = (C_1, C_2, \dots, C_n)$, which refers to the special character [CLS], which is the classification feature symbol. In encrypted traffic classification, a [CLS] Token is a packet-level feature representation, and the corresponding final hidden state is often used as an aggregated sequence representation for classification tasks [23]. The remaining $C_i (i \in [2, n - 1])$ represents the i -th Token in the initial traffic byte sequence. This set of Token sequences serves as the original input of BERT. The first set of embedding vectors in BERT is randomly initialized, expressed $X_{input} = (x_1, x_2, \dots, x_n), x_i \in R^{d \times 1}$, and the embedding dimension d is set to 768. After N rounds of transformer encoding, packet-level and byte-level feature representations can be obtained.

Moreover, the sequence information is captured according to the byte context of the packet and the architecture of the model, and various features are encoded to form a byte-level feature vector. This multilevel token design enables the model to understand and process sequence information more comprehensively and perform better in classification tasks.

4.4.2. Long-Term Byte Sequence Modeling Module

The byte sequence output by the packet feature extraction module is a kind of time series data, and an RNN can model this time series well. To capture this long-distance dependence between time series data, we use BiGRU to model it. This gated recurrent unit (GRU) is a special kind of RNN, and BiGRU helps to alleviate the vanishing gradient problem of RNNs; moreover, its convergence speed is higher than that of LSTM, and the difference in accuracy is smaller [38,39].

The GRU includes an update gate and an output gate, and the state of the current cell of the model is obtained by calculating and summing the state of the previous cell. The specific gate units in the GRU model are calculated using the following formula:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (1)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (2)$$

$$\tilde{h}_t = \tanh(W \cdot [r_t \times h_{t-1}, x_t]) \quad (3)$$

$$h_t = (1 - z_t) \times h_{t-1} + z_t \times \tilde{h}_t \quad (4)$$

where z_t , r_t , $\tilde{h}_t$, and h_t represent the update gate, reset gate, candidate hidden state, and output of the GRU, respectively. W_z , W_r , and W represent the corresponding state weight matrices. σ represents the sigmoid function. To fully obtain the context information, this paper uses the BiGRU model, the two single-layer GRU models are superimposed in reverse, and the output is also determined by the state of the two superimposed GRUs:

$h_t^{(i)} = [h_t^{\rightarrow(i)}, h_t^{\leftarrow(i)}]$. Where, $h_t^{(i)}$ is the output information of the i -th token in the model, and $h_t^{\rightarrow(i)}$ is the forwards GRU information of the i -th Token in the model, and $h_t^{\leftarrow(i)}$ is the backwards GRU information of the i -th Token in the model. Therefore, the entire BiGRU model output can be expressed as:

$$O_{GRU} = (h_t^{(1)}, h_t^{(2)}, \dots, h_t^{(T)}) \quad (5)$$

However, BiGRU suffers from in-sequence loss when dealing with overly long byte sequences [36], especially as the number of classification targets increases (Section 5.7). When modelling end-to-end sequential relationships, information needs to be passed one by one along a time step [40]. As the sequence of bytes increases, the weight updates at each time step can lead to a gradual dilution of the previous information. This gradual dilution may cause some parts of the sequence, especially those distant from the current time, to lose critical information [40].

4.4.3. Key Enhancement Module

To solve the problem of sequence loss, we introduce the key enhancement module, which is composed of three key parts: a self-attention mechanism, a multihead attention mechanism and an additive attention mechanism [41].

The self-attention mechanism enables the model to learn the byte relationships and features between different time steps in the same time step and selectively assigns higher weights to some key time steps to increase their influence. This helps the model to more comprehensively understand the implicit characteristics of the entire ciphertext byte sequence and more comprehensively model the complex dependencies in encrypted traffic data to improve the sequence loss problem of BiGRU. This approach improves the modelling ability of the model for long-distance sequences. The self-attention input is a matrix consisting of 3 vectors: (Q, K, V) , where $Q = X_{input} \cdot W^Q$ denotes the query vector matrix; $K = X_{input} \cdot W^K$ represents the key vector matrix; and $V = X_{input} \cdot W^V$ denotes the value vector matrix. The output of the self-attention is:

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

In addition, since the randomness of the encryption system increases the complexity of long-distance dependencies, the key classification information may be distributed in different locations and subspaces. The multihead attention mechanism allows the model to focus on information from different representation subspaces at different locations simultaneously [42]. To address the complex implicit features of the encryption system, this model adopts the multihead attention mechanism. By setting eight attention heads, the attention weights generated by each head can capture the relevance and importance of different parts in the sequence and then achieve more comprehensive and multiangle feature extraction at the model level. This multihead structure helps to cope with complex relationships and diverse features in the sequence and subsequently improves the performance of the model in encrypted traffic classification tasks.

After the input feature vector sequence is calculated by the BiGRU network layer, we can obtain the following output vector: $O_{GRU} = (h_t^{(1)}, h_t^{(2)}, \dots, h_t^{(T)})$. T is the length of the sequence, and γ is the weighted average of the sequence vectors. The attention model is constructed as follows:

$$\alpha = \text{softmax}(w^T \tanh(H)) \quad (7)$$

$$\gamma = H\alpha^T \quad (8)$$

where $H \in R^{d^w \times T}$, d^w is the dimension of the word embedding, w is the parameter vector during training, and w^T is its transpose. After the single-head attention calculation, we can obtain a single self-attention output feature value:

$$h^* = \tanh(\gamma) \quad (9)$$

The structure of the multihead attention mechanism allows the model to derive more layers of features from different representation subspaces, thereby enabling the model to capture more contextual information of byte feature sequences. Its essence is to perform k self-attention calculations. In a single self-attention calculation, the dimensions of H are compressed, that is, H is linearly transformed to obtain $W_i^h H$, and $W_i^h \in R^{d^h/k \times d^h}$, $i \in \{1, 2, \dots, k\}$. Then, highly optimized matrix multiplication is achieved by using the multiplicative attention mechanism. We calculate Equations (7)–(9) K times, using different w each time, splice the results and linearly map them to obtain the final result:

$$h_s = w_s \otimes \text{concat}(h_1^*, h_2^*, \dots, h_k^*) \quad (10)$$

The dimension of w_s is $k \times d^h$, and $\otimes$ represents element-wise multiplication.

After multihead attention, additive attention plays a key role in effectively reducing the dimensionality of the multihead attention output. This step not only reduces model complexity but also improves computational efficiency while retaining key information:

$$e_t = \tanh(W_e O_t + b_e) \quad (11)$$

O_t represents the output of the fully connected layer at each time step; W_e and b_e are training and learning parameters, representing parameter vectors and biases, respectively:

$$\alpha_t = \frac{\exp(e_t)}{\sum_t \exp(e_t)} \quad (12)$$

α_t represents the weight value in the byte sequence, the weight value is between $[0, 1]$, and $\exp$ is an exponential function:

$$R = \sum_t \alpha_t O_t \quad (13)$$

O_t of each time step is multiplied by the corresponding weight α_t , and then the weighted results of all time steps are added to obtain the attention feature R , which can be used for classification.

4.4.4. Classification Output Module

In the classification output layer, we directly feed the feature vectors obtained by additive attention into a multilayer perceptron (MLP) with two fully connected layers [43]. This MLP structure has two key advantages. First, it allows the model to better capture features and perform nonlinear mapping, thus improving the encrypted traffic classification performance. Second, through these two-layer MLPs, we make the model more flexible for adapting to different encrypted traffic classification tasks and enhancing its generalizability.

In this paper, the softmax classifier is selected to calculate the conditional probability of each relation type, and the relation category corresponding to the maximum conditional probability is selected as the output of the prediction result. This is because softmax classifiers are generally easier to interpret than other classifiers, such as ReLU, which can directly provide probabilities for each category as output. Finally, the output of the attention mechanism layer is R input into the softmax classifier to calculate the relation classification probability:

$$p(r | Y) = \text{softmax}(W_s \cdot R + b_s) \quad (14)$$

$$\hat{r} = \arg \max(p(r|R)) \quad (15)$$

where W_s is the weight matrix of the classifier, b_s is the bias parameter of the classifier, r is the relationship category, and $\hat{r}$ indicates that the classifier calculates the probability distribution of the input relationship and selects the maximum value as the relationship extraction result.

4.4.5. Model Enhancement

In addition, to improve the convergence speed of the model and avoid the vanishing gradient problem, we introduce a residual connection [44], which is applied to the BiGRU for multihead attention transitions while using layer normalization between each subblock [45]. Such a design helps to accelerate the training convergence of the model and improve the overall learning efficiency.

To solve the problem of gradient disappearance and optimal solution convergence during model training, the output of the self-attention mechanism is processed by residual

and normalization and finally input into the fully connected feedforward network (FFNN), which contains 2 fully connected layers. The FFNN output Y is calculated as follows:

$$Y = \max(0, ZXW_1^F + b_1) \cdot W_2^F + b_2 \quad (16)$$

After the residual and layer normalization operation, the output of the FFNN network is input into the next set of transformers, and the above process is repeated.

5. Experiments and Results Analysis

In this section, a comprehensive understanding of the performance of the LAMBERT model in encrypted traffic classification tasks is obtained through multiaspect evaluation. Four specific encrypted traffic classification tasks (Section 5.1) are demonstrated, highlighting the effectiveness of LAMBERT in different encryption scenarios. We then compare LAMBERT with 10 existing encrypted traffic classification methods (Section 5.2) to reveal its relative advantages in the field. Next, the long-distance byte sequence modelling layer is divided into different uniform paragraphs, and the dependencies between long-distance bytes are justified (Section 5.3). Subsequently, we conduct ablation experiments on LAMBERT components to gain insight into the contribution of each component to the performance (Section 5.4). When focusing on the applicability of LAMBERT in real application scenarios, we first consider the case of imbalanced collected samples from different classification targets in real environments and evaluate the performance of LAMBERT in this case to reveal its robustness (Section 5.5). Second, in practice, the traffic distribution in the classification scenario applied by the pretrained model is not yet known by pre-training. Hence we evaluate this situation in our experiments (Section 5.6). Finally, considering that there may be many classification targets in real applications, we also investigate the performance of LAMBERT when dealing with large-scale classification tasks (Section 5.7).

5.1. Experimental Setup

5.1.1. Experimental Environment

A one Tesla T4 GPU was used for this experiment. The model parameters were set as follows: the number of epochs was set to 20, the initial learning rate was set to 2×10^{-5} , and the learning rate was dynamically adjusted during the training process. The warmup ratio was set to 0.1, the batch size to 32, and the dropout to 0.1. The BERT input sequence length was set to 128, the BERT hidden size and BiGRU hidden size were set to 768, the number of BiGRU layers was set to 2, the number of attention heads was 8, and the cross-entropy loss function was used for loss calculation [41]. AdamW was used as the optimizer [36].

5.1.2. Datasets and Downstream Tasks

To evaluate the effectiveness and generalizability of LAMBERT on encrypted traffic classification tasks, four different encrypted traffic classification tasks were set for four public datasets. The following table shows the statistics of these tasks and their corresponding datasets (Table 1).

Table 1. The four datasets corresponding to the statistical information of the four classification tasks.

Task	Dataset	Packet	Label	Encryption Algorithm
ETVS	ISCX-VPN-Service	60,000	12	AES
ETVA	ISCX-VPN-App	77,163	17	AES
ETMA	USTC-TFC	97,115	20	AES RC4 3DES
ETTA	CSTNET-TLS 1.3	581,709	120	AES CHACHA20

Task 1: Encrypted Traffic Classification on the VPN-Service(ETVS) Task

This task classifies the encrypted traffic of network services that use virtual private networks (VPNS) to communicate. We used a commonly used dataset for current research, namely, the ISCX-VPN-Service [46]. It is publicly available at the University of New

Brunswick, and classified network packets captured at the data link layer according to the different activities performed during network communication by the application they were generated with (e.g., Chat, file transfer, or video call), divided into different pcap files, resulting in the ISCX-VPN-Service dataset containing 12 categories.

Task 2: Encrypted Traffic Classification on the VPN-App(ETVA) Task

This task classifies encrypted traffic from applications that use virtual private networks (VPNS) to communicate. We used a commonly used dataset for current research, ISCX-VPN-App [46], which is also publicly available from the University of New Brunswick, and classified network packets captured at the data link layer according to the application they were generated with (e.g., Gmail, Skype, and Facebook), divided into different pcap files, resulting in the ISCX-VPN-App dataset containing 17 categories.

Task 3: Encrypted Traffic Classification on the Malware App(ETMA) Task

This task aims to classify encrypted traffic consisting of malware and benign apps. The USTC-TFC dataset was used and it consists of two parts [47]. The first part contains ten kinds of public website malware traffic collected from a real network environment by CTU researchers from 2011 to 2015 [48]. The second part includes ten kinds of normal traffic collected using an IXIA BPS [49], including eight common applications.

Task 4: Encrypted Traffic Classification on the TLS1.3 App(ETTA) Task

This task aims to classify the encrypted traffic of applications that use the novel encryption protocol TLS 1.3. It uses the CSTNET-TLS 1.3 dataset which is a collection of encrypted traffic from 120 applications collected under CSTNET from March to July 2021 by researchers from the Institute of Information Technology of the Chinese Academy of Sciences [17]. These applications were acquired from Alexa Top-5000 [50] with TLS 1.3 deployed, and each session flow is tagged by the server name indication (SNI).

5.1.3. Evaluation Metrics

We use four evaluation metrics: precision (P_c), recall (R_c), F1 score ($F1_c$), and accuracy (acc). Since encrypted traffic classification is a multicategory task, we need to calculate the above metrics for each category separately. Specifically, we use N as the total number of training samples; TP_c is used to indicate the quantity that originally belongs to category c , and is predicted by the model as c ; FP_c indicates the quantity that originally does not belong to category c but is predicted by the model as c ; TN_c indicates the quantity that does not belong to category c , and is not predicted to be class c ; and FN_c indicates the quantity that belongs to class FN_c but is misclassified to another class. Hence, the definitions of the aforementioned four evaluation metrics can be given as follows:

$$P_c = \frac{TP_c}{TP_c + FP_c} \quad (17)$$

$$R_c = \frac{TP_c}{TP_c + FN_c} \quad (18)$$

$$F1_c = \frac{2P_cR_c}{P_c + R_c} \quad (19)$$

$$acc = \frac{TP_c + TN_c}{TP_c + TN_c + FP_c + FN_c} \quad (20)$$

5.2. Comparison with State-of-the-Art Methods

The proposed LAMBERT model was compared with various state-of-the-art methods, including (1) a fingerprint-based construction method, FlowPrint [29]; (2) machine learning-based methods, AppScanner [16] and BIND [31]; (3) deep learning-based methods, DF [15], FS-Net [19], and DeepPacket [21]; and (4) pre-training-based methods, PERT [25], ET-BERT [17], BFCN [30], and Bi-ETC [31].

In the comparison, four encrypted traffic classification tasks (ETVS, ETVA, ETMA, and ETTA) were used as benchmarks, and the corresponding datasets (ISCX-VPN-Service,

ISCX-VPN-APP, USTC-TFC, and CSTNET-TLS 1.3) were used for evaluation. Table 2 lists the performance comparisons between LAMBERT and each other on different tasks.

Table 2. Comparison of LAMBERT with other state-of-the-art methods.

Task	ETVS				ETVA				ETMA				ETTA			
	AC	PR	RC	F1	AC	PR	RC	F1	AC	PR	RC	F1	AC	PR	RC	F1
FlowPrint	0.7962	0.8042	0.7812	0.7820	0.8767	0.6697	0.6651	0.6531	0.8146	0.6434	0.7002	0.6573	0.1261	0.1354	0.1272	0.1116
AppScanner	0.7182	0.7339	0.7225	0.7197	0.6266	0.4864	0.5198	0.4935	0.8954	0.8984	0.8968	0.8892	0.6662	0.6246	0.6327	0.6201
	0.7534	0.7583	0.7488	0.7420	0.6767	0.5152	0.5153	0.4965	0.8457	0.8681	0.8382	0.8396	0.7964	0.7605	0.7650	0.7560
	0.7154	0.7192	0.7104	0.7102	0.6116	0.5706	0.4752	0.4799	0.7787	0.7883	0.7819	0.7593	0.7936	0.7721	0.7573	0.7602
FS-Net	0.7205	0.7502	0.7238	0.7131	0.6647	0.4819	0.4848	0.4737	0.8846	0.8846	0.8920	0.8840	0.8639	0.8404	0.8349	0.8322
Deep Packet	0.9329	0.9377	0.9306	0.9321	0.9758	0.9785	0.9745	0.9765	0.9640	0.9650	0.9631	0.9641	0.8019	0.4315	0.2689	0.4022
PERT	0.9352	0.9400	0.9349	0.9368	0.8229	0.7092	0.7173	0.6992	0.9909	0.9911	0.9910	0.9911	0.8915	0.8846	0.8719	0.8741
ET-BERT	0.9887	0.9888	0.9887	0.9887	0.9959	0.9932	0.9922	0.9927	0.9909	0.9910	0.9911	0.9910	0.9714	0.9716	0.9719	0.9717
BFCN	-	-	-	-	0.9965	0.9936	0.9947	0.9941	-	-	-	-	-	-	-	-
Bi-ETC	-	-	-	-	0.9970	0.9934	0.9951	0.9943	-	-	-	-	-	-	-	-
LAMBERT	0.9915	0.9917	0.9915	0.9915	0.9970	0.9952	0.9952	0.9952	0.9930	0.9931	0.9930	0.9930	0.9743	0.9742	0.9742	0.9741

Bold represents the best classification.

The experimental results show that the LAMBERT model performs well in extracting, learning and discriminating traffic features efficiently, and its classification performance on four different datasets is significantly better than that of other models. In particular, the pretrained BERT-based LAMBERT model achieves significant improvement in classification accuracy on both imbalanced sample datasets and nonpretrained sample datasets.

Task ETVS: Although this dataset has been used in the pre-training stage and the pretrained model has learned the traffic representation well under this data distribution, in the fine-tuning stage, the LAMBERT model can still further improve the performance.

Task ETVA: This dataset has obvious data imbalance challenges, but LAMBERT is able to successfully capture the implicit features between bytes through its unique BiGRU-Attention mechanism on AIM_Chat and ICQ two unbalanced sample labels (Section 5.5), which effectively alleviates the impact of unbalanced data on the classification effect.

Task ETMA: Since this dataset has not been used in the pre-training stage, it is more difficult for the native BERT fine-tuning model to learn the implicit features compared with other dataset classification tasks (ETVS, ETVA, etc.) used in pre-training. In contrast, the LAMBERT model makes full use of byte-level feature vectors, so its effect is the most significant improvement in this task, with all four indicators reaching more than 99.30%.

Task ETTA: Compared with those of the other three datasets, the sample base (465,367) and the number of classification targets (120) of this dataset are significantly improved. Compared with the native BERT fine-tuning model, LAMBERT shows more stable performance advantages, with all four indicators reaching more than 97.41%. This highlights its performance improvement in the face of larger scale and diverse data.

5.3. Rationality Analysis of Long-Distance Dependency between Bytes

To investigate the impact of long-distance dependencies on classification accuracy, we used LAMBERT as the baseline model (1Seg) and divided the long-distance byte sequence modelling layer into different uniform paragraphs (8Seg and 32Seg) to model long-distance dependencies between different paragraph bytes.

We employed t-distributed random neighbourhood embedding (t-SNE) to plot the images as 2D images, as shown in Figure 5. We chose the USTC-TFC dataset used in Section 5.6 show the model classification results of three different Seg partitions.

It can be seen that 1Seg (a) shows the best classification performance. The results of Zeus and Skype show the smallest overlap, with a clear boundary between the two clusters. However, as the number of divided paragraphs in 8Seg (b) and 32Seg (c) increases, the classification overlap for Zeus and Skype also increases, the boundaries become increasingly less clear, and the classification accuracy decreases (1.5% and 4.7%, respectively). This result is consistent with what is described in Section 5.5. This decrease can be attributed

to the inability of the model to capture long-distance dependencies between bytes in different paragraphs, which affects the classification performance. Therefore, this study highlights the importance of considering long-distance byte dependencies in encrypted traffic classification to improve classification accuracy and robustness.

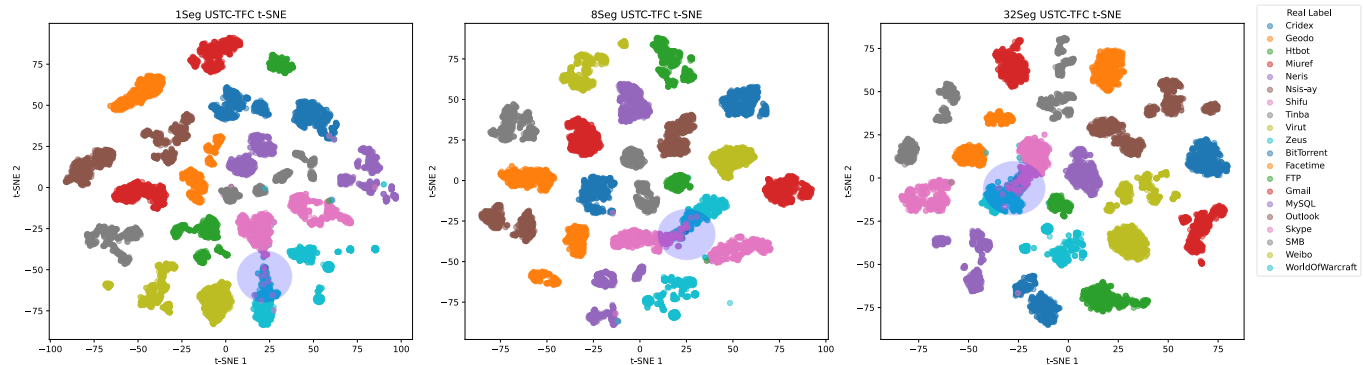


Figure 5. The impact of long-distance byte dependencies on classification accuracy.

5.4. Ablation Study

We conducted ablation experiments to gain insight into the contribution of the components of the LAMBERT model to its performance. The ablation experiments covered the ISCX VPN-Service and VPN-App datasets used in pre-training as well as the USTC-TFC dataset without pre-training.

- **Without BiGRU:** LAMBERT model without BiGRU, where the output of BERT is directly fed to the attention block for fully connected classification, aiming to evaluate the effect of BiGRU.
- **Without Multi-head Self-Attention (MA):** LAMBERT model without multihead self-attention passes the time step output by the BiGRU to the full connection for classification after additive attention dimensionality reduction and is used to evaluate the effect of multihead self-attention.
- **Without Additive Attention (AA):** LAMBERT model without additive attention takes the horizontal average of the feature vectors obtained by multihead self-attention and then passes it to the fully connected layer for classification, which is used to evaluate the effect of additive attention.
- **Without Fully Connected Layer (FC):** A baseline model with only BERT and FC, which is a native BERT fine-tuned classification model. It feeds the [CLS] vector output by BERT directly into FC for classification, which is used as a baseline control along with LAMBERT (Table 3).

Table 3. Comparison of ablation experimental data.

Class	Index				VPN-Service				VPN-App				USTC-TFC			
	BiGRU	MA	AA	FC	AC	PR	RC	F1	AC	PR	RC	F1	AC	PR	RC	F1
LAMBERT (full mode)	✓	✓	✓	✓	0.9915	0.9917	0.9915	0.9915	0.9970	0.9952	0.9952	0.9952	0.9930	0.9931	0.9930	0.9930
Without BiGRU	✗	✓	✓	✓	0.9905	0.9906	0.9905	0.9905	0.9965	0.9929	0.9941	0.9935	0.9917	0.9918	0.9918	0.9917
Without Multi-head Self-Attention	✓	✗	✓	✓	0.9898	0.9899	0.9898	0.9898	0.9966	0.9936	0.9948	0.9942	0.9914	0.9915	0.9915	0.9914
Without Additive Attention	✓	✓	✗	✓	0.9905	0.9907	0.9905	0.9905	0.9968	0.9924	0.9955	0.9939	0.9920	0.9921	0.9920	0.9920
Without FC	✗	✗	✗	✓	0.9887	0.9888	0.9887	0.9887	0.9959	0.9932	0.9922	0.9927	0.9909	0.9910	0.9911	0.9910

✓, ✗ represents whether the model has this module. Bold represents the best classification.

From the ablation experimental results, it can be seen that the contributions of BiGRU and multihead attention are significant. In the absence of these two modules, all four indicators showed a significant decline. The main effect of additive attention is to reduce the dimension of the output of the multihead self-attention while retaining the characteristics

of the output matrix of the multihead self-attention as much as possible, which makes a certain contribution to the performance improvement of LAMBERT.

5.5. Unbalanced Sample Analysis

To verify the effectiveness and robustness of LAMBERT in the case of an uneven distribution of sample numbers, unbalanced sample analysis was carried out. Up to 5000 samples from each label in the ISCX VPN-APP and USTC-TFC datasets were selected for evaluation (Tables 4 and 5). Specifically, for the ISCX VPN-APP dataset, we set the number of samples in the AIM_Chat and ICQ categories to 1340 and 823, respectively. For the USTC-TFC dataset, the number of samples in the FTP category was set to 1903.

Table 4. Categories of VPN-App.

Label	Class	Count
1	AIM_Chat	1340
2	Email	5000
3	Facebook	5000
4	Gmail	5000
5	Hangout	5000
6	ICQ	823
7	Netflix	5000
8	SCP	5000
9	Skype	5000
10	Spotify	5000
11	Tor	5000
12	Torrent	5000
13	Vimeo	5000
14	VoipBuster	5000
15	VPN-FTPS	5000
16	VPN-SFTP	5000
17	YouTube	5000

The bold represents the unbalanced sample category and its number.

Table 5. USTC-TFC categories.

Label	Class	Count
1	Cridex	5000
2	Geodo	5000
3	Htbot	5000
4	Miuref	5000
5	Neris	5000
6	Nsis-ay	823
7	Shifu	5000
8	Tinba	5000
9	Virut	5000
10	Zeus	5000
11	BitTorrent	5000
12	FaceTime	5000
13	FTP	1903
14	Gmail	5000
15	MySQL	5000
16	Outlook	5000
17	Skype	5000
18	SMB	5000
19	Weibo	5000
20	WorldOfWarcraft	5000

The bold represents the unbalanced sample category and its number.

When evaluating model performance, we choose F1 score as our primary evaluation metric. The F1 score combines precision and recall and can accurately measure the performance of the model on each class to avoid the class with a large sample size dominating the evaluation results. Hence this is especially suitable for the case of unbalanced samples [34]. From the ISCX-APP dataset, it can be observed that compared with other balanced sample labels, LAMBERT has a significant improvement in performance on ICQ and AIM_Chat, two categories with fewer samples. On the USTC-TFC dataset, due to the relatively clear features of the category with fewer FTP samples, the accuracy reached 99.52%, and further improvement was difficult. However, LAMBERT shows significant performance improvement on Zeus and Skype, which are easily confused categories (Figures 6 and 7).

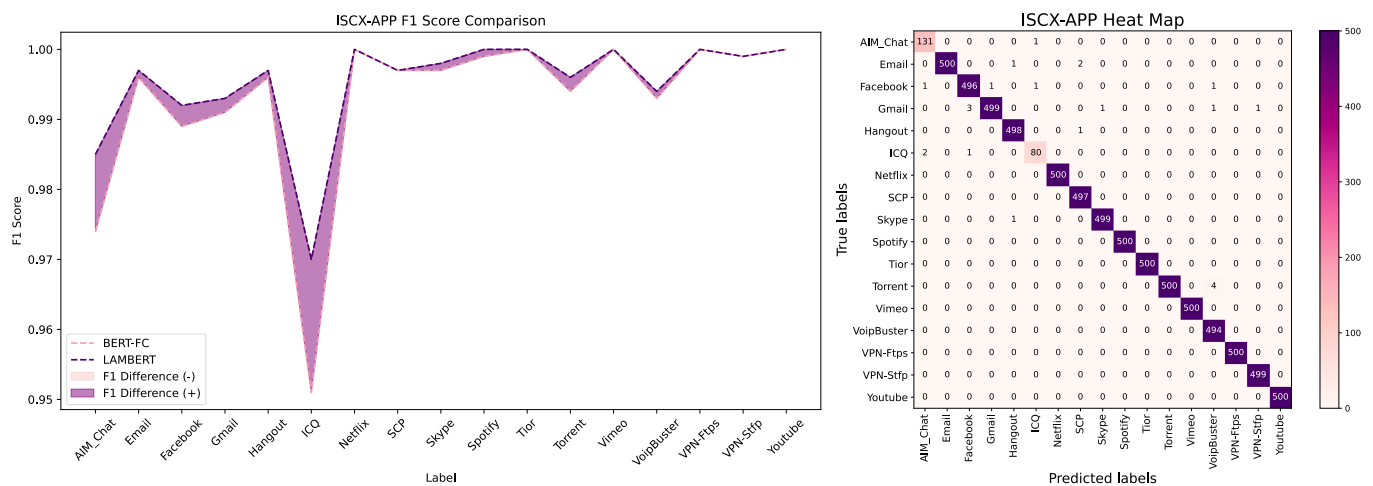


Figure 6. The LAMBERT F1 score comparison data and confusion matrix in the scenario where the number of samples is unevenly distributed (ISCX-APP).

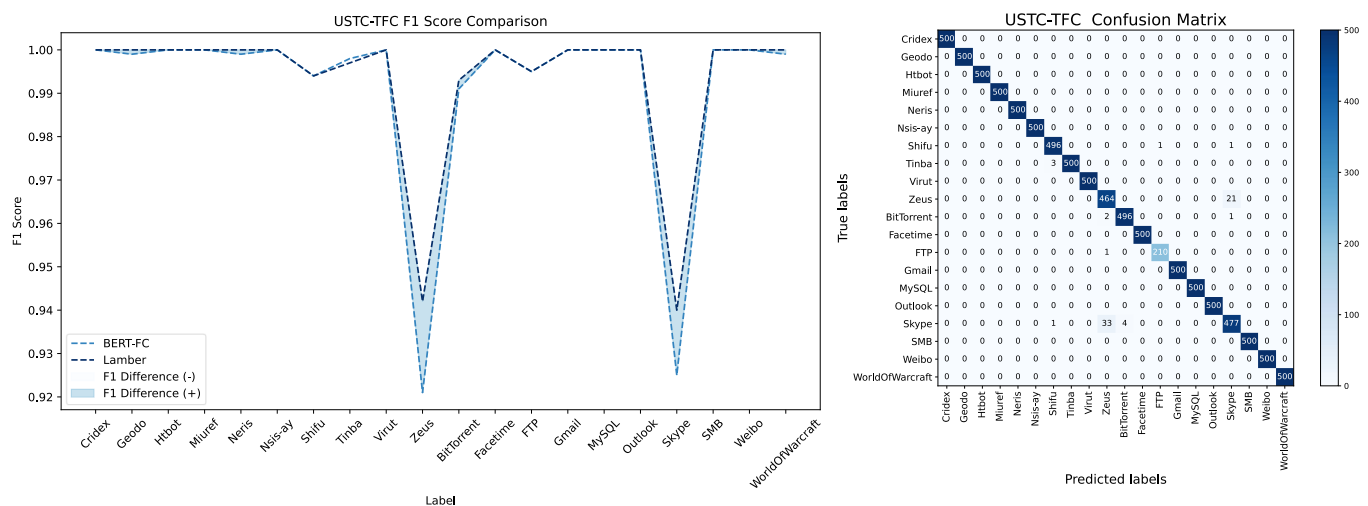


Figure 7. The LAMBERT F1 score comparison data and confusion matrix in the scenario where the number of samples is unevenly distributed (USTC-TFC).

5.6. Analysis of Samples without Pre-Training

To verify whether LAMBERT is effective in processing datasets without pre-training, we conducted comparative experiments on the USTC-TFC dataset without pre-training. In this experiment, we compared LAMBERT with BERT-FC, BERT-Attention-FC, and BERT-BiGRU-FC to evaluate the generalization ability of LAMBERT on samples not pretrained (Figure 8).

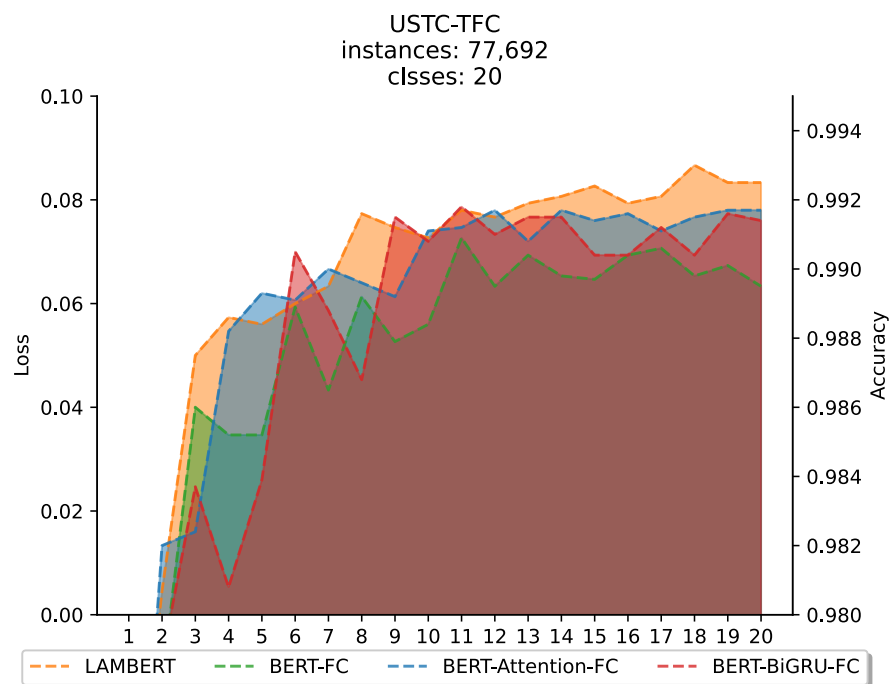


Figure 8. Comparative experiments of LAMBERT for processing datasets without pre-training.

The comparison results show that LAMBERT performs best on the nontrained datasets. The second best are BERT-Attention-FC and BERT-BiGRU-FC without BiGRU or attention blocks. Since this dataset is not pretrained, the ability of the pretrained BERT model to express encrypted traffic in this dataset is relatively insufficient, so the effect of the traditional BERT-FC method is not satisfactory. In contrast, the other three methods further learn the features on the dataset based on the encrypted traffic representation output by the original BERT, so the effect is better than that of the native BERT-FC method. This shows that LAMBERT has good generalizability on nonpretrained datasets and can effectively classify encrypted traffic.

5.7. Large Classification Sample Analysis

To verify the ability of LAMBERT to perform large sample classification tasks, we designed a comparative experiment in which LAMBERT was compared with BERT-FC, BERT-Attention-FC and BERT-BiGRU-FC. Experiments were conducted on the CSTNET-TLS 1.3 dataset, which contains 120 classification targets and 465,367 selected samples, which is 6 to 7 times greater than that of the other three datasets and hence the performance of LAMBERT in large sample classification could be evaluated.

According to the results, LAMBERT still performs best. BERT-FC performs similarly to BERT-Attention-FC and inferior to LAMBERT, while BERT-BiGRU-FC performs the worst. In addition, compared with Section 5.5, LAMBERT shows more stable performance on datasets with a large number of samples.

There are two reasons for this result. On the one hand, the CSTNET-TLS 1.3 dataset is pretrained, and the pretrained BERT already has good expressive power. Thus good results could be obtained by simple BERT-FC. When only adding an attention block after BERT but lacking BiGRU, the content that can be noticed is limited due to the loss of the context modelling ability of BiGRU, so the results show that BERT-Attention-FC has similar effects to BERT-FC. On the other hand, for BERT-BiGRU-FC, due to the sequence loss phenomenon of BiGRU at long distances, this disadvantage is significantly amplified for the multiclassification target and large sample size of the CSTNET-TLS 1.3 dataset, unlike the effect that can surpass BERT-FC on the USTC-TFC dataset (Figure 9).

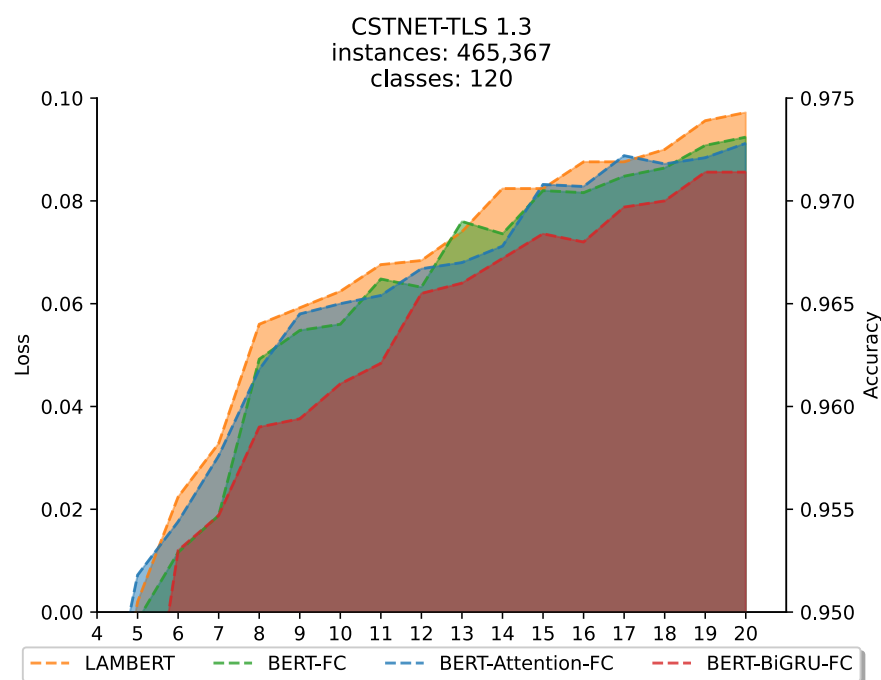


Figure 9. Comparative experiments of LAMBERT for processing large sample classification tasks.

In summary, the experimental results show that simply adding attention or BiGRU after BERT does not significantly improve performance and is sometimes even worse than simple BERT-FC. However, LAMBERT appropriately chooses its combination mode and shows excellent performance in large sample classification tasks.

6. Discussion

By comparing the results of the experiments and ablation experiments, we observe that LAMBERT performs excellently on encrypted traffic classification tasks. It integrates the structure of BiGRU, multihead self-attention and additive attention so that the model can better capture the long-distance dependence at the byte level and improve the classification accuracy.

Although the experimental results show that LAMBERT achieves excellent performance on multiple tasks and datasets, there are some potential limitations and room for improvement, which we discuss in the following sections.

1. Improve the fine-tuning method

In this paper, we have mainly focused on the structural improvement of the fine-tuning model, but it has been shown that improving the fine-tuning method can also help to improve the classification performance. Some possible directions for improvement are as follows:

- (1) Further pre-training: For those scenarios where there is a lack of pre-training data, consider further pre-training on the original pretrained model. This can make the model adaptable to new domain data and improve its generalizability.
- (2) Multitask tuning: Multitask learning has shown promising performance in deep learning. Applying LAMBERT to multiple encrypted traffic classification tasks without pre-training datasets and sharing knowledge between these tasks may further improve the classification performance.

2. Research on classification under a single complex encryption system

Different cryptographic implementations have different degrees of randomness [11]. According to the interpretability analysis of ET-BERT, the classification accuracy decreases significantly on the encryption system with a single encryption algorithm

and strong randomness of the encryption algorithm. The experimental results of LAMBERT further confirm this view. The distribution of ciphertext is mainly affected by the original text, the encryption algorithm and the encryption configuration. It is a task full of potential and challenges to realize classification by perceiving the change law of the original text in the ciphertext distribution without using the randomness difference between different encryption algorithms.

3. Classification research in superlarge target classification scenarios

From the experimental results, we find that the classification accuracy decreases for datasets with more classification targets. On three classification tasks, ETVS, ETVA and ETMA, with 10 to 20 classification targets, the F1 score reaches more than 99%, while on the ETMA with 120 classification targets, the F1 score decreases to 97%. However, in some real classification scenarios, the number of classification targets often reaches thousands. It can be assumed that in this case, the accuracy of classification will decrease significantly. Therefore, it is a great challenge to maintain high classification accuracy in such superlarge object classification scenarios.

When discussing the performance of the LAMBERT model, we specifically emphasize security as a factor that cannot be ignored. Considering the sensitivity of encrypted traffic and privacy protection requirements, the current model adopts the ciphertext part of the transport layer as the input and classifies the traffic without decrypting the ciphertext. By accurately classifying encrypted traffic, LAMBERT helps classify requirements in a variety of scenarios while protecting the privacy of user data. Future research will continue to explore ways to improve the accuracy and security of encrypted traffic classification without violating user privacy.

In summary, the LAMBERT model has achieved significant performance improvement in encrypted traffic classification, but there are still many potential research directions and room for improvement. This will help to further improve the classification accuracy and generalization ability to meet the evolving needs of encrypted traffic classification.

7. Conclusions

In this paper, we have conducted a comprehensive and in-depth experimental study to further improve the accuracy and generalizability of encrypted traffic classification by improving the existing BERT fine-tuning model. Specifically, we utilize the BiGRU-Attention module, in which the BiGRU models and learns the context of the feature vector sequence from BERT and fully extracts the temporal correlation between different sequences to effectively capture the relationship between long-distance sequences. In addition, we introduce an attention mechanism to dynamically assign different weights to input features, thereby enhancing the influence of important features and helping to avoid losing key information when processing long sequences. The experimental results establish that our proposed LAMBERT model shows excellent performance in multiple classification scenarios, including VPN Services, VPN APP, malicious traffic classification, and TLS 1.3 applications. In particular, LAMBERT can effectively improve the classification performance when dealing with various challenges, such as an unbalanced distribution of samples, lack of pre-training or large-scale classification datasets. This provides strong evidence for the robustness and generality of our method. In the future, on the one hand, we hope to further improve the classification accuracy of encrypted traffic by improving the fine-tuning method based on LAMBERT. On the other hand, we shall explore the classification ability of LAMBERT in two extreme scenarios.

Author Contributions: Conceptualization, T.L., X.M. and N.H.; Methodology, T.L. and X.M.; Formal analysis, T.L.; Data curation, T.L. and K.Z.G.; Investigation, L.L.; Project administration, N.H.; Resources, Y.Z. and N.H.; Supervision, X.L. and Y.Z.; Validation, L.L.; Writing—original draft, T.L. and X.M.; Writing—review and editing, X.M., X.L. and N.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in The National Key Research and Development Program of China (2021YFB2012402), The Major Key Project of PCL (Grant No. PCL2023A07-4).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Mohassel, P.; Zhang, Y. Secureml: A system for scalable privacy-preserving machine learning. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 19–38.
2. Available online: https://transparencyreport.google.com/https/overview?hl=zh_CN (accessed on 1 January 2024).
3. Ma, Y.; Chai, X.; Gan, Z.; Zhang, Y. Privacy-preserving TPE-based JPEG image retrieval in cloud-assisted internet of things. *IEEE Internet Things J.* **2023**, *11*, 4842–4856. [\[CrossRef\]](#)
4. Ning, J.; Poh, G.S.; Loh, J.C.; Chia, J.; Chang, E.C. PrivDPI: Privacy-preserving encrypted traffic inspection with reusable obfuscated rules. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, London, UK, 11–15 November 2019; pp. 1657–1670.
5. Fauvel, K.; Chen, F.; Rossi, D. A lightweight, efficient and explainable-by-design convolutional neural network for internet traffic classification. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Long Beach, CA, USA, 6–10 August 2023; pp. 4013–4023.
6. Shi, H.; Li, H.; Zhang, D.; Cheng, C.; Cao, X. An efficient feature generation approach based on deep learning and feature selection techniques for traffic classification. *Comput. Netw.* **2018**, *132*, 81–98. [\[CrossRef\]](#)
7. Bujlow, T.; Carela-Español, V.; Barlet-Ros, P. Independent comparison of popular DPI tools for traffic classification. *Comput. Netw.* **2015**, *76*, 75–89. [\[CrossRef\]](#)
8. Rezaei, S.; Liu, X. Deep learning for encrypted traffic classification: An overview. *IEEE Commun. Mag.* **2019**, *57*, 76–81. [\[CrossRef\]](#)
9. Suo, H.; Liu, Z.; Wan, J.; Zhou, K. Security and privacy in mobile cloud computing. In Proceedings of the 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC), Sardinia, Italy, 1–5 July 2013; pp. 655–659. [\[CrossRef\]](#)
10. Xie, R.; Wang, Y.; Cao, J.; Dong, E.; Xu, M.; Sun, K.; Li, Q.; Shen, L.; Zhang, M. Rosetta: Enabling robust tls encrypted traffic classification in diverse network environments with tcp-aware traffic augmentation. In Proceedings of the ACM Turing Award Celebration Conference, Wuhan, China, 28–30 July 2023; pp. 131–132.
11. Velan, P.; Čermák, M.; Čeleda, P.; Drašar, M. A survey of methods for encrypted traffic classification and analysis. *Int. J. Netw. Manag.* **2015**, *25*, 355–374. [\[CrossRef\]](#)
12. He, W.; Li, S.; Wang, W.; Wei, M.; Qiu, B. Cryptoeyes: Privacy preserving classification over encrypted images. In Proceedings of the IEEE INFOCOM 2021-IEEE Conference on Computer Communications, Vancouver, BC, Canada, 10–13 May 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–10.
13. Crotti, M.; Dusi, M.; Gringoli, F.; Salgarelli, L. Traffic classification through simple statistical fingerprinting. *ACM SIGCOMM Comput. Commun. Rev.* **2007**, *37*, 5–16. [\[CrossRef\]](#)
14. Shapira, T.; Shavitt, Y. Flowpic: Encrypted internet traffic classification is as easy as image recognition. In Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Paris, France, 29 April–2 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 680–687.
15. Yang, Y.; Kang, C.; Gou, G.; Li, Z.; Xiong, G. TLS/SSL encrypted traffic classification with autoencoder and convolutional neural network. In Proceedings of the 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Exeter, UK, 28–30 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 362–369.
16. Taylor, V.F.; Spolaor, R.; Conti, M.; Martinovic, I. Robust smartphone app identification via encrypted network traffic analysis. *IEEE Trans. Inf. Forensics Secur.* **2017**, *13*, 63–78. [\[CrossRef\]](#)
17. Lin, X.; Xiong, G.; Gou, G.; Li, Z.; Shi, J.; Yu, J. Et-bert: A contextualized datagram representation with pre-training transformers for encrypted traffic classification. In Proceedings of the ACM Web Conference 2022, Lyon, France, 25–29 April 2022; pp. 633–642.
18. Aceto, G.; Ciunzio, D.; Montieri, A.; Pescapè, A. MIMETIC: Mobile encrypted traffic classification using multimodal deep learning. *Comput. Netw.* **2019**, *165*, 106944. [\[CrossRef\]](#)
19. Liu, C.; He, L.; Xiong, G.; Cao, Z.; Li, Z. Fs-net: A flow sequence network for encrypted traffic classification. In Proceedings of the IEEE INFOCOM 2019-IEEE Conference On Computer Communications, Paris, France, 29 April–2 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1171–1179.
20. Ren, X.; Gu, H.; Wei, W. Tree-RNN: Tree structural recurrent neural network for network traffic classification. *Expert Syst. Appl.* **2021**, *167*, 114363. [\[CrossRef\]](#)
21. Lotfollahi, M.; Jafari Siavoshani, M.; Shiral Hossein Zade, R.; Saberian, M. Deep packet: A novel approach for encrypted traffic classification using deep learning. *Soft Comput.* **2020**, *24*, 1999–2012. [\[CrossRef\]](#)
22. Wang, W.; Zhu, M.; Wang, J.; Zeng, X.; Yang, Z. 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), Beijing, China, 22–24 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 43–48.

23. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
24. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
25. He, H.Y.; Yang, Z.G.; Chen, X.N. PERT: Payload encoding representation from transformer for encrypted traffic classification. In Proceedings of the 2020 ITU Kaleidoscope: Industry-Driven Digital Transformation (ITU K), Ha Noi, Vietnam, 7–11 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–8.
26. Hu, X.; Gu, C.; Chen, Y.; Wei, F. CBD: A deep-learning-based scheme for encrypted traffic classification with a general pre-training method. *Sensors* **2021**, *21*, 8231. [[CrossRef](#)] [[PubMed](#)]
27. Van Ede, T.; Bortolameotti, R.; Continella, A.; Ren, J.; Dubois, D.J.; Lindorfer, M.; Choffnes, D.; Van Steen, M.; Peter, A. Flowprint: Semi-supervised mobile-app fingerprinting on encrypted network traffic. In Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 23–26 February 2020; Volume 27.
28. Al-Naami, K.; Chandra, S.; Mustafa, A.; Khan, L.; Lin, Z.; Hamlen, K.; Thuraisingham, B. Adaptive encrypted traffic fingerprinting with bi-directional dependence. In Proceedings of the 32nd Annual Conference on Computer Security Applications, Los Angeles, CA, USA, 5–9 December 2016; pp. 177–188.
29. Sirinam, P.; Imani, M.; Juarez, M.; Wright, M. Deep fingerprinting: Undermining website fingerprinting defenses with deep learning. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018; pp. 1928–1943.
30. Shi, Z.; Luktarhan, N.; Song, Y.; Tian, G. BFCN: A novel classification method of encrypted traffic based on BERT and CNN. *Electronics* **2023**, *12*, 516. [[CrossRef](#)]
31. Ma, X.; Liu, T.; Hu, N.; Liu, X. Bi-ETC: A Bidirectional Encrypted Traffic Classification Model Based on BERT and BiLSTM. In Proceedings of the 2023 8th International Conference on Data Science in Cyberspace (DSC), Hefei, China, 18–20 August 2023; IEEE: Piscataway, NJ, USA 2023; pp. 197–204.
32. Sengupta, S.; Ganguly, N.; De, P.; Chakraborty, S. Exploiting diversity in android tls implementations for mobile app traffic classification. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 1657–1668.
33. Canard, S.; Diop, A.; Kheir, N.; Painsavoine, M.; Sabt, M. BlindIDS: Market-compliant and privacy-friendly intrusion detection system over encrypted traffic. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, Abu Dhabi, United Arab Emirates, 2–6 April 2017; pp. 561–574.
34. Weber, J.; Senior, A.E. Catalytic mechanism of F1-ATPase. *Biochim. Biophys. Acta (BBA)—Bioenerg.* **1997**, *1319*, 19–58. [[CrossRef](#)]
35. Peng, Z.; Huang, W.; Gu, S.; Xie, L.; Wang, Y.; Jiao, J.; Ye, Q. Conformer: Local features coupling global representations for visual recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 367–376.
36. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
37. Niu, D.; Yu, M.; Sun, L.; Gao, T.; Wang, K. Short-term multi-energy load forecasting for integrated energy systems based on CNN-BiGRU optimized by attention mechanism. *Appl. Energy* **2022**, *313*, 118801. [[CrossRef](#)]
38. Rahman, A.; Srikumar, V.; Smith, A.D. Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. *Appl. Energy* **2018**, *212*, 372–385. [[CrossRef](#)]
39. Zhang, Y.; Ai, Q.; Lin, L.; Yuan, S.; Li, Z. A very short-term load forecasting method based on deep LSTM RNN at zone level. *Power Syst. Technol.* **2019**, *43*, 1884–1892.
40. Cinar, Y.G.; Mirisaei, H.; Goswami, P.; Gaussier, E.; Ait-Bachir, A. Period-aware content attention RNNs for time series forecasting with missing values. *Neurocomputing* **2018**, *312*, 177–186. [[CrossRef](#)]
41. Zhang, Z.; Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 31.
42. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 30.
43. Zhang, Z.; Zhang, Z. Artificial neural network. In *Multivariate Time Series Analysis in Climate and Environmental Research*; Springer: Cham, Switzerland, 2018; pp. 1–35.
44. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
45. Jin, X.; Lan, C.; Zeng, W.; Chen, Z.; Zhang, L. Style normalization and restitution for generalizable person re-identification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 3143–3152.
46. Gil, G.D.; Lashkari, A.H.; Mamun, M.; Ghorbani, A.A. Characterization of encrypted and VPN traffic using time-related features. In Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP 2016), Rome, Italy, 19–21 February 2016; SciTePress: Setúbal, Portugal, 2016; pp. 407–414.
47. Wang, W.; Zhu, M.; Zeng, X.; Ye, X.; Sheng, Y. Malware traffic classification using convolutional neural network for representation learning. In Proceedings of the 2017 International Conference on Information Networking (ICOIN), Da Nang, Vietnam, 11–13 January 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 712–717.
48. Available online: <https://stratosphereips.org/category/dataset.html> (accessed on 15 June 2019).

-
49. Available online: <https://www.ixiacom.com/products/breakingpoint> (accessed on 7 May 2022).
50. Available online: <https://www.alexa.com/topsites> (accessed on 10 September 2021).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.