2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom) | 979-8-3503-8199-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/TrustCom60117.2023.00039

FlowBERT: An Encrypted Traffic Classification Model Based on Transformers Using Flow Sequence

Quanbo Pan[1,2], Yang Yu[3], Hanbing Yan[4,*], Maoli Wang[3] and Bingzhi Qi[5]

[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3]Qufu Normal University, Jining, China
[4]National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing, China
[5]Shandong Jianzhu University, Jinan, China
[*]Corresponding author, Email: yhb@cert.org.cn

***Abstract*—With the widespread application of network traffic encryption, traffic identification has become increasingly critical. As the types of encryption protocols continue to grow, identifying encrypted traffic with limited training samples has become more challenging. In recent years, pre-training models have been extensively applied in natural language processing due to their ability to utilize a large amount of unlabeled data effectively. However, when applied to encrypted traffic identification, these methods lack sufficient information extraction from encrypted network traffic, resulting in the loss of some essential features and negatively impacting the recognition performance of such approaches. Therefore, we proposed an encrypted traffic classification model based on a Transformer named FlowBERT. In FlowBERT, the semantic features of the traffic can be learned from two dimensions: payload and packet length sequence in large-scale, unlabeled encrypted traffic scenarios. Length sequences are encoded to extract traffic sequence features efficiently, enabling the model to learn the contextual semantic relationships within the sequences. Simultaneously, the pre-training process is improved by balancing data samples, enhancing the performance of the pre-training model. We validated the performance of this method on both classic encrypted traffic classification datasets and the novel network protocol DoH dataset. We concluded that our approach demonstrates robustness and superior recognition performance compared to similar methods.**

***Index Terms*—encrypted traffic classification, transformers, deep learning**

I. Introduction

Network traffic identification is the foundation for network security maintenance, daily network operation and maintenance, network traffic monitoring, and network design. Accurately identifying the types of network traffic is a crucial focus of current research. With the development of technology, an increasing amount of network traffic is transmitted using encryption. Although encrypted traffic technologies can protect user privacy, they also obscure traces of malicious attacks by malware and attackers. Traditional traffic analysis methods are insufficient for effectively identifying malicious encrypted traffic. Consequently, the identification and classification of encrypted traffic have attracted widespread attention from both academia and industry [1]. Encrypted traffic classification refers to categorizing and identifying data within a network, making it a critical area in network security and management. Traffic classification techniques play a pivotal role in effectively determining malicious traffic and providing valuable data for network management, assisting administrators in devising appropriate network security policies and measures [2], [3].

The method based on fingerprint features, the process based on feature engineering, and the method based on machine learning belong to the traditional encrypted traffic classification methods [4]–[6]. Fingerprint-based methods are not suitable for current encrypted traffic classification tasks. Many studies have employed machine learning techniques for encrypted traffic classification. However, traditional machine learning methods require manual feature extraction, and traffic classification tasks involve complex and numerous features, making it challenging to extract all relevant features manually.

In recent years, the pre-training model method has been widely applied in the field of NLP, showing strong advantages [7], [8]. This method can improve the model's classification accuracy and generalization ability by using many unlabeled data [9]. However, in the task of encrypted network traffic classification, such methods do not extract enough traffic information, only pay attention to the encryption load in traffic, and lose other valuable information, which affects the further improvement of the classification effect [10], [11].

This paper proposes an encrypted traffic classification model based on Transformers using a Flow Sequence called FlowBERT. It uses a multimodal fusion method to improve the pre-training model and obtains payload and packet length sequence features from the traffic. During the model training phase, payload and length sequence information are integrated into the model training process. Then, we fine-tune the model for different downstream classification tasks. This method has high robustness. The contributions of our paper can be

2324-9013/23/$31.00 ©2023 IEEE
DOI 10.1109/TrustCom60117.2023.00039

summarized below:

- We propose a BERT-based multimodal traffic recognition framework, which learns representations for a series of encrypted traffic classification tasks from multiple dimensions using large-scale unlabeled encrypted traffic; the presented results demonstrate promising performance, particularly excelling in novel network protocols.
- To efficiently extract traffic sequence features, we employ length-based sequence characteristics and effectively encode them in conjunction with payload information. This enables the model to learn contextual semantic relationships within the sequences during training.
- We also improve the pre-training process by balancing data samples to enhance the performance of the pre-training model.
- Validated on classic encrypted traffic classification datasets and new network protocol datasets, the framework demonstrates robustness and achieves an accuracy of 99.73% compared to similar methods.

The remainder of this paper is organized as follows. Section II summarizes the related works. Section III illustrates the detailed extraction framework of FlowBERT. The experimental results are provided and discussed in Section IV. Finally, we arrive at conclusions in Section V.

II. Related Works

In this section, we mainly introduce the current mainstream methods of encrypted traffic classification, including fingerprint-based methods, feature engineering-based methods, deep learning-based methods, and pre-training model-based methods.

A. Fingerprint-based methods

Fingerprint-based methods (or Deep Packet Inspection, DPI) classify applications by matching key strings in network traffic.

Hayes [12] proposed a website fingerprinting method for large-scale internet data. This approach initially utilizes a machine learning algorithm known as k-nearest neighbors (k-NN) to construct the feature space. Subsequently, a random forest classifier is employed on this feature space for website fingerprinting. Experimental results demonstrate the efficacy of this method in effectively identifying and categorizing encrypted website traffic within extensive internet data. FlowPrint [13] employs semi-supervised learning and feature extraction techniques to model and identify mobile application network traffic without decrypting it. Analyzing network traffic's time and size characteristics extracts unique fingerprints associated with the applications. Hynek et al. [14] adopted statistical features of packet length and inter-packet interval time, selecting three ensemble learning methods - AdaBoost, Bagging, and C4.5 - as algorithms.

B. Feature engineering-based methods

Although the payload information of the traffic is encrypted, practical information can still be obtained through its features. These features include time sequence, length sequence, message types, and statistical characteristics.

Schuster [15] proposed a method for remotely identifying encrypted video streams. This method first extracts features using the periodic burst patterns of video streams. Appscanner [4] classifies mobile applications based on packet length sequences and utilizes the random forest algorithm to classify 110 popular Android applications generated by bots. Chen et al. [16] proposed a method for encrypted network protocol stacks in a multi-protocol environment based on composite deep learning by taking full advantage of the Markov properties of multi-PDU length sequences.

C. Deep learning-based methods

Traditional feature extraction methods often fail to fully explore the data's potential information, restricting the classification models' performance and leading to limited adaptability to new encryption protocols and complex communication patterns.

To overcome these limitations, deep learning methods have gradually been introduced into the field of encrypted traffic classification [1]. Xinming Ren et al. proposed a Tree-Structured Recursive Neural Network (Tree-RNN) [17], which can automatically learn the nonlinear relationship between input and output data and use the tree structure to divide large categories into smaller ones. Ting-Li Huoh et al. introduced a Graph Neural Network (GNN) model [18] for classifying encrypted network traffic by considering the original bytes of data packets, metadata, and packet relationships simultaneously.

By leveraging deep learning techniques, these approaches have shown promising results in addressing the challenges of encrypted traffic classification, demonstrating improved adaptability to diverse protocols and communication patterns. Integrating deep learning methods in this domain can advance the field further and enhance the performance of traffic classification models.

Aceto [19] proposed a hybrid neural network called App-net for encrypted mobile traffic classification. App-net combines the advantages of convolutional neural networks and long short-term memory networks. Wenting Wei et al. [20] introduced the ABL-TC traffic classification method. Saadat Izadi et al. [21] created a traffic classification model that can accurately identify traffic types by using convolutional neural networks (CNN), ant lion meta-heuristic algorithms (ALO), and self-organizing maps (SOM).

D. Pre-training models

In recent years, Transformer models have been successfully applied to the field of encrypted traffic classification, drawing inspiration from their successful applications in natural language processing and image domains [7], [22], [23]. Compared to traditional Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), Transformer models have shown better performance in traffic classification tasks by leveraging a large amount of unlabeled data, improving classification accuracy, generalization capability, parallel computing efficiency, and modeling capacity. Building upon pre-trained Transformer models, Xinjie Lin et al. [24] proposed a method called ET-BERT for encrypted traffic classification. This method represents packets as contextualized vectors and processes them using pre-trained BERT models to extract meaningful features. These features are then input into a classifier for classification.

Peng Lin et al. [25] designed a novel multimodal deep learning framework called PEAN for encrypted traffic classification. He et al. [26] introduced a method called Payload Encoding Representation from Transformer (PERT), which uses advanced dynamic word embedding techniques for automatic traffic feature extraction. Zihan Wu et al. [27] introduced a Transformer-based robust intrusion detection system (RTIDS) to reconstruct feature representations, striking a balance between dimensionality reduction and feature preservation in imbalanced datasets. Zhao et al. [28] proposed an efficient classifier with an attention mechanism to extract features from flow sequences at a lower computational cost.

Pre-training models have demonstrated powerful feature learning and representation capabilities, enabling direct modeling and classification of raw encrypted traffic data without relying on handcrafted features. Therefore, pre-training models have significantly progressed in encrypted traffic classification tasks, adapting to complex encryption communication patterns and novel encryption algorithms while improving classification accuracy and robustness.

However, current research mainly focuses on traffic payload information rather than other flow sequence features. The ability of pre-training models to extract information from unlabeled data directly affects the classification performance of the models.

III. Models

This section aims to describe the proposed Transformer-based end-to-end multimodal encrypted traffic classification framework, FlowBERT, and validate its performance on publicly available datasets and novel network protocols. The architecture of our model comprises three main modules: data processing, pretraining, and fine-tuning. Figure 1 is the overview of FlowBERT.

In the data processing module, we extract and clean the unlabeled data and then slice and encode it. In the pre-training process, we input many unlabeled data payload and sequence information into the word vector. At the same time, in the fine-tuning stage, we adapt to different downstream encrypted traffic classification tasks.

A. Data Processing

The data processing module is critical in preparing the input data for subsequent stages, ensuring that our model can effectively learn and represent the complex relationships in the encrypted traffic data. By following this meticulous data processing method, our framework can better handle the challenges posed by real-world network traffic, leading to improved performance and accuracy in encrypted traffic classification tasks. In the data processing module of our framework, we implement a comprehensive four-step procedure: traffic cleansing, slicing, feature extraction, and encoding.

1) Traffic Cleaning: It is common for network traffic data to contain various forms of noise, redundancy, and anomalies, such as duplicate packets and incomplete data. A meticulous traffic cleansing process is employed to enhance the accuracy and efficiency of subsequent analysis and processing. This stage effectively filters out irrelevant or erroneous data, resulting in a refined dataset.

2) Slicing: After cleansing, the traffic is sliced and segmented into distinct fragments based on the shared IP, port, and protocol attributes. This segmentation enables the grouping related data and facilitates more granular analysis and pattern recognition.

3) Feature Extraction: Within each segmented flow, we perform feature extraction to capture pertinent information. Precisely, we extract payload and length sequence information from the sliced flows. By concatenating payload and length data obtained from network packets, we form a comprehensive textual representation, serving as the input sequence for BERT.

4) Encoding: To effectively process the textual input using BERT, we employ BERT's Tokenizer to tokenize the concatenated string into a series of tokens, including both words and subwords. Following tokenization, we structure the token sequence to conform to BERT's input format, which entails appending a special [CLS] token at the beginning of the sequence and a [SEP] token at the end. Subsequently, the input sequence is fed into the pre-trained BERT model for encoding, generating embedding vectors that represent the input sequence. We term this specific encoding procedure as "Payload-Length Tokenization." After encoding pre-training, use the embedded vector representation as input features, perform full connection, and use the softmax function for classification. The process is as follows in Algorithm 1.

Compared to feature extraction methods based on network data packets, the Payload-Length Tokenization method offers many distinct advantages, establishing it as a robust and efficient approach for network traffic classification. The key advantages of this method are as follows:

(1) Strong Generalization Capability: The Payload-Length Tokenization method exhibits exceptional generalization capability, enabling it to handle diverse types of network traffic, including encrypted, compressed, and other variations. Unlike traditional methods that rely on specific assumptions about data packet payloads and lengths, this approach operates without such prerequisites, providing greater flexibility in handling a wide range of network traffic scenarios.

(2) Automatic Feature Representation Learning: Leveraging deep learning-based techniques, the Payload-Length Tokenization method facilitates automatic feature representation learning. This approach significantly improves feature

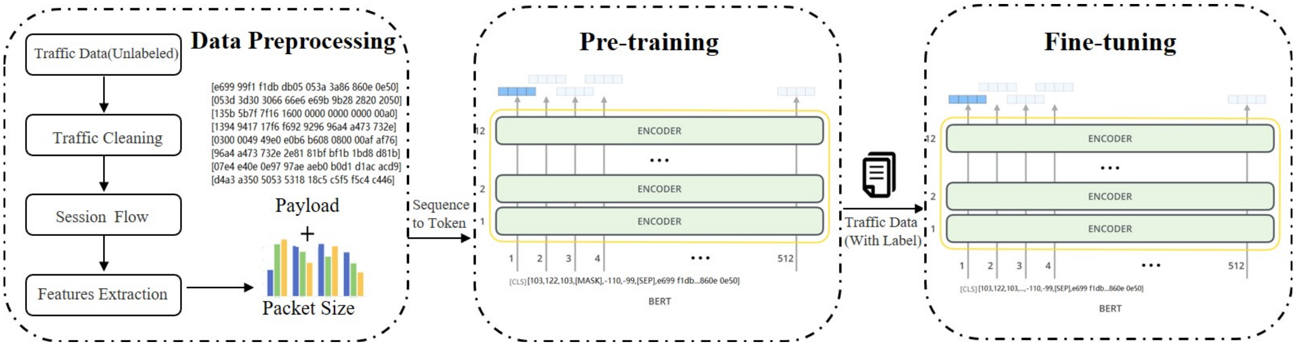
Data Preprocessing
Traffic Data(Unlabeled)
Traffic Cleaning
Session Flow
Features Extraction
[e699 99f1 f1db db05 053a 3a86 860e 0e50]
[053d 3d30 3066 66e6 e69b 9b28 2820 2050]
[135b 5b7f 7f16 1600 0000 0000 0000 00a0]
[1394 9417 17f6 f692 9296 96a4 a473 732e]
[0300 0049 49e0 e0b6 b608 0800 00af af76]
[96a4 a473 732e 2e81 81bf bf1b 1bd8 d81b]
[07e4 e40e 0e97 97ae aeb0 b0d1 d1ac acd9]
[d4a3 a350 5053 5318 18c5 c5f5 f5c4 c446]
Payload
+
Packet Size
Sequence to Token
Pre-training
12
2
1
ENCODER
ENCODER
ENCODER
1 2 3 4 ... 512
[CLS] [103,122,103,[MASK],-110,-99,[SEP],e699 f1db...860e 0e50]
BERT
Traffic Data (With Label)
Fine-tuning
12
2
1
ENCODER
ENCODER
ENCODER
1 2 3 4 ... 512
[CLS] [103,122,103,...,-110,-99,[SEP],e699 f1db...860e 0e50]
BERT

Fig. 1. Overview of FlowBERT

Algorithm 1 Network Traffic Classification

```
Require: $P$: a collection of pcap files $p_i$
Ensure: the classification result of each pcap file in $P$
procedure TRAFFIC CLASSIFICATION($P$)
    for all $p \in P$ do
        // Step 1: Traffic Cleaning
        $cleaned\_traffic \leftarrow$ Clean the traffic data in $p$
        // Step 2: Traffic Slicing
        $sliced\_traffic \leftarrow$ slice $cleaned\_traffic$ into segments with the same IP, port, and protocol
        // Step 3: Feature Extraction
        $features \leftarrow$ extract payload information and length sequence from $sliced\_traffic$
        // Step 4: Encoding with BERT
        $text\_input \leftarrow$ concatenate payload and length as a string
        $tokens \leftarrow$ converting $text\_input$ into tokens
        $input\_ids \leftarrow$ convert $tokens$ to ids
        $input\_mask \leftarrow$ create mask of $input\_ids$
        $segment\_ids \leftarrow$ create the token type of $input\_ids$
        $Bert\ input \leftarrow$ [[CLS], $input\_ids$, [SEP], $segment\_ids$, $input\_mask$, [PAD]]
        $embedding\_vector \leftarrow$ embedding the $Bert\ input$
        // Step 5: Traffic Classification
        $W \leftarrow$ fully-connected layer($embedding\_vector$)
        $W' \leftarrow$ Softmax($W$)
        $y \leftarrow$ argmax($W'$)
        output the classification result of $p$ as $y$
    end for
end procedure
```

extraction efficiency by circumventing the laborious and time-consuming process of manually designing features. Moreover, the reliance on automated learning reduces errors stemming from human factors, leading to more reliable and accurate feature representations.

(3) Multimodal Feature Extraction: The Payload-Length Tokenization method adopts a multimodal approach by extracting payload and length sequences. This combined utilization of traffic features enhances the model's capacity to capture more intricate patterns and relationships within the data. Empirical evidence has demonstrated that this multimodal feature extraction strategy yields superior performance in network traffic classification tasks, outperforming alternative methods.

(4) Scalability: A significant advantage of the Payload-Length Tokenization method lies in its scalability. This adaptability allows seamless integration with other models and algorithms, further empowering researchers to enhance network traffic analysis's accuracy and efficiency. Combining this method with complementary approaches opens possibilities for comprehensive and sophisticated traffic analysis solutions.

Advantages of Combining Payload and Packet Length Sequences:

- Comprehensive Feature Description: Payload provides content-related information, while packet length sequences offer temporal insights. Combining both features provides a comprehensive feature description, enabling the classification model to simultaneously consider application-layer data and data transmission patterns, thereby improving classification accuracy.
- Enhanced Resilience Against Adversarial Tactics: Malicious actors may attempt to disguise either content or traffic patterns. The combination of Payload and packet length sequences enhances the model's ability to detect adversarial behaviors, as these features offer distinct informational perspectives that are challenging to evade simultaneously.
- Holistic Analysis: The approach of combining Payload and packet length sequences permits holistic analysis, such as detecting abnormal transmission patterns within specific application-layer traffic or identifying concealed malicious behavior within regular communications. This contributes to improved network security and threat detection capabilities.

Payload and packet length sequences provide different types of information, and their integration allows for the full exploitation of their respective strengths, thereby enhancing the performance and accuracy of encrypted traffic classification. In practical applications, selecting appropriate feature combinations and suitable classification models is crucial to ensure optimal classification results.

B. Pre-training FlowBERT

FlowBERT uses BERT as a pre-training model. BERT is a widely applied pre-training model in the natural language processing domain. It is trained on a large-scale unlabeled text corpus to learn the relationships and semantic information between words, and it has achieved excellent performance in various NLP tasks. For network traffic classification tasks, it is necessary to convert network traffic data into textual form and process it using BERT for classification purposes.

In FlowBERT, we use BERT architecture to encode payload and packet length sequences. Specifically, we first convert network traffic data into text sequences, each character representing a byte. Next, we feed the text sequences into BERT, where each character is embedded into a high-dimensional vector. BERT encodes the input through a multi-head self-attention mechanism, and the output of each layer is a vector containing contextual information from the input sequence.

We employed a self-supervised learning approach, Masked Language Modeling (MLM) [24] to adapt to the network traffic classification task. During the pre-training process, we randomly selected a certain proportion of payload and length sequences in the input sequences and represented them using a special token "[MASK]." Subsequently, we tasked the model with predicting these tokens to obtain more representative encoding representations. During training, we utilize the negative log-likelihood to minimize the discrepancy between the model predictions and the true labels. Therefore, the loss function can be represented as follows:

$$l_{MBM}(\theta) = -\sum_{i=1}^{k} \log(P(MASK_i|\bar{X};\theta)) \qquad (1)$$

Where θ represents the set of trainable parameters that the model learns, the probability denoted as P, is modeled by the Transformer encoder with the parameter set θ. To achieve this, the input data X undergoes a masking process to generate the representation $\bar{X}$, wherein certain tokens in the token sequence are concealed using the notation MASKi, indicating the masked token at the ith position.

The essence of the Multi-Head Self-Attention Mechanism is to map a query (Q) onto a series of key-value pairs (K-V) through an attention function. In Self-Attention, Q, K, and V are obtained from the same value x through three different linear transformation matrices W^Q, W^K, and W^V. Self-Attention is a variant of the attention mechanism with less dependency on external information, making it more suitable for capturing the internal correlations of data or features. The formula for the Multi-Head Self-Attention Mechanism is as follows:

$$\text{Attention}(Q,K,V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (2)$$

After the pre-training process of the model is completed, we utilize the output vectors from the pre-trained model as the input for the FlowBERT encoder and fine-tune it on the traffic dataset. During the fine-tuning process, we update BERT's parameters via supervised training on the dataset to enhance FlowBERT's performance in network traffic classification tasks.

C. Fine-tuning FlowBERT

The fine-tuning process of FlowBERT follows a similar approach to that of BERT. Upon completing the pretraining of FlowBERT, the model undergoes fine-tuning, wherein task-specific data packets or flow representations are fed into the pre-training FlowBERT model. All model parameters are adjusted to adapt to the specific traffic classification task during fine-tuning. In this phase, the last layer of our model outputs semantic representations of payload and length sequences as a vector, denoted as $h_{\text{payload+length}}$. This vector is then connected to a fully connected layer, and its output represents the probability distribution for each traffic category. The prediction results, denoted as γ, are obtained through a Softmax function applied to the output of the fully connected layer:

$$\gamma = \text{Softmax}(W \cdot (h_{\text{payload+length}}) + b) \qquad (3)$$

W, b represents the parameters corresponding to the fully connected layer. Due to the similar structure of fine-tuning and pre-training, the FlowBERT model exhibits good generalizability and adaptability.

Since the fine-tuning structure closely resembles pretraining, the FlowBERT model exhibits robust generalizability and adaptability across various traffic classification tasks. By leveraging the pre-trained knowledge obtained during the initial training phase, the fine-tuning process fine-tunes the model to adapt to specific downstream tasks, making FlowBERT a highly versatile and efficient tool for handling diverse traffic patterns and encryption scenarios. Moreover, combining the last layer's payload and length sequence vectors allows the model to capture essential semantic information, enhancing its discriminative power and classification accuracy.

The capability of FlowBERT to effectively integrate pre-trained and task-specific representations facilitates quick adaptation to new encryption protocols, novel communication patterns, and emerging threats, contributing to its strong performance and applicability in practical encrypted traffic classification scenarios.

IV. Experiment and Discussion

A. Dataset

For fine-tuning, we chose two publicly available network traffic classification datasets, ISCX-Tor-NonTor-2017 and DoHBrw-2020, as well as our own captured HTTPS-DoH dataset, see Table I. Both public datasets are widely used benchmark datasets containing various types of network traffic data. To simulate real network traffic data, we developed a program in Python that emulates using a browser to request websites through DoH servers. We utilized multiple public DoH domain servers and requested websites from the Alexa top 500 list, resulting in a dataset divided into two categories: HTTPS and DoH. We use these three datasets to test the performance of FlowBERT and compare it with other methods.

TABLE I
DATASET DETAILS

Dataset	Category	Sample
ISCX-Tor-NonTor-2017	Tor	6000
	Non-Tor	6000
CIRA-CIC-DoHBrw-2020	dns2tcp	3645
	dnscat2	3600
	iodine	3643
DoH-HTTPS	DoH	440
	HTTPS	500

B. Evaluation Index

The classifier demonstrates accurate classifications in two scenarios: true positives (TP) decisions, where the traffic is correctly classified as the corresponding service, and true negatives (TN) decisions, where traffic of other services is correctly identified. On the other hand, the classifier may produce erroneous outputs, including false positives (FP) and false negatives (FN) decisions.

Given the variations in data distribution for different service categories in the encrypted traffic dataset, some categories may be well-balanced, while others could be imbalanced. The following metrics are employed to ensure a comprehensive performance evaluation for each classification task: Accuracy, Precision, Recall, and F1-score.

C. Quantitative Evaluation

We used four models as baselines: AppNet [19], Beauty [15], Cumul [12], and ET-BERT [24], and our proposed method FlowBERT achieved the best performance on two public datasets. The quantitative metrics are shown in Table II. Our approach achieved an accuracy of 99.83%, which is 0.36% higher than the best baseline (Beauty, 99.47%). The precision value reached 99.94%, surpassing the best ET-BERT baseline by 0.13%, and the recall rate was 99.68%, surpassing the best baseline (AppNet, 99.43%) by 1.47%. The F1-score was 99.80%, which is 0.80% higher than the best ET-BERT baseline. These results demonstrate that our method outperforms traditional public datasets.

We selected the DoHBrw-2020 dataset to validate the robustness of our model, ensuring excellent classification performance in novel network protocols. The experimental results (TableIII.) indeed support this claim. The accuracy reached 99.73%, surpassing the best baseline (Beauty, 99.63%) by 0.10%. The precision rate was 99.67%, surpassing the best baseline (AppNet, 99.47%) by 0.2%, and the recall rate reached 99.78%, surpassing the best baseline (ET-BERT, 99.01%) by 0.77%. The F1-score reached 99.72%, exhibiting a 0.61% improvement compared to the best baseline (AppNet, 99.11%).

TABLE II
COMPARISON RESULTS ON ISCX-TOR-NONTOR-2017.

Model	Accuracy	Precision	Recall	F1-score
AppNet	98.96	69.57	99.43	77.83
Beauty	99.47	49.74	50.00	49.87
Cumul	98.63	59.96	89.35	65.65
ET-BERT	99.02	100	98.03	99.00
FlowBERT	**99.83**	**99.94**	**99.68**	**99.80**

TABLE III
COMPARISON RESULTS ON DOHBRW-2020.

Model	Accuracy	Precision	Recall	F1-score
AppNet	99.58	99.47	98.75	99.11
Beauty	99.63	99.23	99.00	99.10
Cumul	98.63	96.27	97.94	97.09
ET-BERT	99.35	99.10	99.01	99.06
FlowBERT	**99.73**	**99.67**	**99.78**	**99.72**

In the comparative experiments using the collected traffic data (as shown in Table IV), the FlowBERT model continued to outperform other models. Its accuracy was 98.72%, surpassing the best baseline (ET-BERT, 98.40) by 0.32%. The precision rate was 98.70%, slightly higher than the best baseline (ET-BERT, 98.69) by 0.01%, and the recall rate was 98.70%, surpassing the best baseline (AppNet) by 0.52%. The F1-score reached 98.70%, exceeding its best baseline (ET-BERT, 98.36) by 0.34%. These results indicate that our method maintains excellent performance and robustness in real traffic environments. The outcomes demonstrate FlowBERT's effectiveness and adaptability for traffic classification tasks, making it a promising and practical solution in the field of network security.

TABLE IV
COMPARISON RESULTS ON OUR DATASET OF DOH-HTTPS.

Model	Accuracy	Precision	Recall	F1-score
AppNet	98.33	98.50	98.18	98.31
Beauty	94.01	93.83	94.03	93.95
Cumul	90.09	90.05	90.11	90.07
ET-BERT	98.40	98.69	98.05	98.36
FlowBERT	**98.72**	**98.70**	**98.70**	**98.70**

D. Ablation Experiments

To validate the advantages of our approach, we conducted ablation experiments, and the results are presented in Table V. We compared the classification results using only payload or length sequences as features. The experimental findings demonstrate that our method outperformed other approaches, achieving an accuracy of 98.72%, followed by payload-only (98.40%). The precision rate reached 98.70%, which slightly improved upon the payload-only method (98.69%) by 0.01%. Moreover, our method exhibited a higher recall rate of 98.70%, showing a notable improvement compared to using only payload as features (98.05%), with a gain of 0.65%. The F1-score was 98.70%, surpassing the payload-only approach (98.36%) by 2.24%, compared to using only length sequences as features (96.51%).

The robust performance achieved when using only length sequences as features highlight their relevance in capturing essential traffic data characteristics. This suggests that incorporating the length sequences complements the payload-based features, reinforcing the model's capacity to discern traffic patterns more accurately.

These ablation experiments serve as crucial evidence of the efficacy of our proposed approach. The results demonstrate that integrating payload and length sequences in FlowBERT effectively enhances the model's performance, surpassing the individual use of either payload or length sequences. The observed improvements in accuracy, precision, recall, and F1-score substantiate the merit of considering payload and length information for comprehensive traffic feature extraction.

TABLE V
ABLATION RESULTS OF FLOWBERT

Model	Accuracy	Precision	Recall	F1-score
Payload	98.40	98.69	98.05	98.36
Length	96.51	95.34	97.61	96.46
Payload+Length	**98.72**	**98.70**	**98.70**	**98.70**

E. Flow Length Parameters Selection

We also examined the impact of truncating different lengths of flows on the results and the size of the resulting data files, as shown in Figure 2. From the figure, it can be observed that truncating flows to different lengths indeed influences the experimental outcomes. Specifically, when the flow length is set to N =200, the accuracy is only 99.30%. As the flow length increases to N =300, the accuracy improves to 99.80%, and when it reaches N =500, the accuracy stabilizes at 99.83%. However, it is worth noting that at a truncation length of N =400, there is a slight decrease in accuracy, reaching 99.72%. We attribute this decline to the longer length sequences encoding of some traffic data.

Furthermore, with the increase in the truncated flow length, the size of the generated data files also gradually grows. For instance, the data file size is 124,228 KB when the flow length is N =200, 124,784 KB when the flow length is N =500, and 125,927 KB when the flow length is N =2000. The table shows that the optimal balance between accuracy and data file size is achieved when the flow length is set to N =500. The accuracy is maximized at this length, and the data file size remains relatively optimal compared to other truncation lengths.

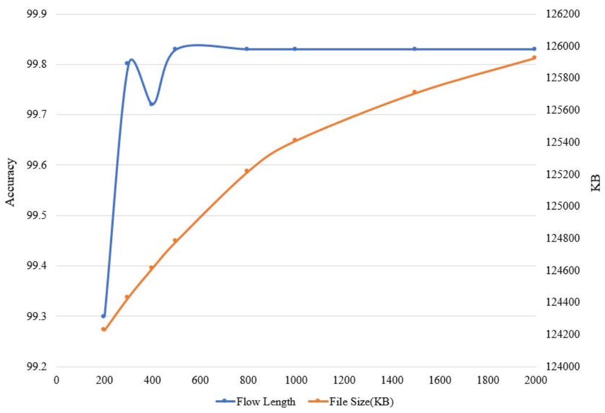

Fig. 2. The Relationship between the Length of the Flow Sequence and the Size of File.

F. Pre-training Phase Data Balancing

The Transformer model utilizes unlabeled data during pretraining, effectively utilizing existing network traffic. However, our experiments revealed that the choice of training dataset during pretraining also significantly influences downstream classification tasks. As depicted in Figure 3, we constructed five groups of pretraining datasets, each consisting of five categories: Tor, Non-Tor (NorTor), DNS over HTTPS (DoH), Malicious DoH, and Other traffic data. These datasets were composed with varying proportions to generate different pre-trained models used for classification. Our specific downstream tasks focused on classifying DoH traffic and Malicious DoH traffic.

In particular, when the pretraining dataset contained an equal proportion of benign DoH and malicious DoH samples, it achieved the optimal binary classification performance of 99.20%. To explore the impact of data distribution, we conducted experiments with imbalanced datasets, using 500 times more benign DoH samples than malicious DoH samples and vice versa. The results indicated a decrease in classification rates, reaching 98.93% and 98.84%, respectively. To further validate our findings, we introduced additional types of traffic data, including Tor and Non-Tor traffic, which also led to a decline in classification performance, reaching 98.84%. Subsequently, we incorporated more types of traffic data, resulting in a further decrease in classification performance, reaching 98.75%. These experimental outcomes demonstrate that the model's classification performance improves when the pretraining dataset is more closely aligned with the downstream classification task. Maintaining a balanced distribution of samples in the pretraining dataset also enhances the model's performance in the downstream classification tasks.

V. CONCLUSION

In this research, we introduce FlowBERT, a novel Transformer-based model designed for traffic classification utilizing sequential flows. Our approach stands out by effectively extracting payload and length features from traffic data

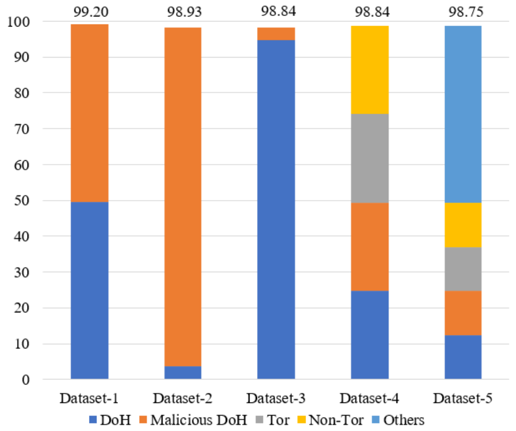

Fig. 3. Performance of Different Types of Pre-training Datasets

and encoding them multimodally during pretraining. To gain deeper insights into the model's functionality, we conducted ablation experiments, systematically analyzing the impact of different modalities on the classification model. Additionally, we examined the influence of data balance during the pre-training stage to further comprehend its implications on model performance.

Through comprehensive evaluations of publicly available datasets and real-world traffic scenarios, as well as comparative analyses with several state-of-the-art encryption traffic classification algorithms, our experimental results conclusively demonstrate the superior performance and robustness of our proposed FlowBERT model. The method exhibits a remarkable ability to handle various traffic patterns with heightened accuracy and stability, underscoring its potential to enhance the field of traffic classification and network security.

References

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

[2] N. Al Khater and R. E. Overill, "Network traffic classification techniques and challenges," in *2015 Tenth international conference on digital information management (ICDIM)*. IEEE, 2015, pp. 43–48.

[3] J. Zhao, X. Jing, Z. Yan, and W. Pedrycz, "Network traffic classification for data fusion: A survey," *Information Fusion*, vol. 72, pp. 22–47, 2021.

[4] 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.

[5] Y.-n. Dong, J.-j. Zhao, and J. Jin, "Novel feature selection and classification of internet video traffic based on a hierarchical scheme," *Computer Networks*, vol. 119, pp. 102–111, 2017.

[6] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE communications surveys & tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[7] 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.

[8] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[9] Y. Yue, X. Chen, Z. Han, X. Zeng, and Y. Zhu, "Contrastive learning enhanced intrusion detection," *IEEE Transactions on Network and Service Management*, 2022.

[10] T. Shapira and Y. Shavitt, "Flowpic: A generic representation for encrypted traffic classification and applications identification," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1218–1232, 2021.

[11] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 689–696.

[12] A. Panchenko, F. Lanze, J. Pennekamp, T. Engel, A. Zinnen, M. Henze, and K. Wehrle, "Website fingerprinting at internet scale." in *NDSS*, 2016.

[13] T. Van Ede, R. Bortolameotti, A. Continella, J. Ren, D. J. Dubois, M. Lindorfer, D. Choffnes, M. van Steen, and A. Peter, "Flowprint: Semi-supervised mobile-app fingerprinting on encrypted network traffic," in *Network and Distributed System Security Symposium (NDSS)*, vol. 27, 2020.

[14] K. Hynek and T. Cejka, "Privacy illusion: Beware of unpadded doh," in *2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, 2020, pp. 0621–0628.

[15] R. Schuster, V. Shmatikov, and E. Tromer, "Beauty and the burst: Remote identification of encrypted video streams." in *USENIX Security Symposium*, 2017, pp. 1357–1374.

[16] Z. Chen, G. Cheng, Z. Xu, S. Guo, Y. Zhou, and Y. Zhao, "Length matters: Scalable fast encrypted internet traffic service classification based on multiple protocol data unit length sequence with composite deep learning," *Digital Communications and Networks*, vol. 8, no. 3, pp. 289–302, 2022.

[17] X. Ren, H. Gu, and W. Wei, "Tree-rnn: Tree structural recurrent neural network for network traffic classification," *Expert Systems with Applications*, vol. 167, p. 114363, 2021.

[18] T.-L. Huoh, Y. Luo, P. Li, and T. Zhang, "Flow-based encrypted network traffic classification with graph neural networks," *IEEE Transactions on Network and Service Management*, 2022.

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

[20] W. Wei, H. Gu, W. Deng, Z. Xiao, and X. Ren, "Abl-tc: A lightweight design for network traffic classification empowered by deep learning," *Neurocomputing*, vol. 489, pp. 333–344, 2022.

[21] S. Izadi, M. Ahmadi, and R. Nikbazm, "Network traffic classification using convolutional neural network and ant-lion optimization," *Computers and Electrical Engineering*, vol. 101, p. 108024, 2022.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[23] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu *et al.*, "A survey on vision transformer," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 1, pp. 87–110, 2022.

[24] 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," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 633–642.

[25] P. Lin, K. Ye, Y. Hu, Y. Lin, and C.-Z. Xu, "A novel multimodal deep learning framework for encrypted traffic classification," *IEEE/ACM Transactions on Networking*, 2022.

[26] H. Y. He, Z. G. Yang, and X. N. Chen, "Pert: Payload encoding representation from transformer for encrypted traffic classification," in *2020 ITU Kaleidoscope: Industry-Driven Digital Transformation (ITU K)*. IEEE, 2020, pp. 1–8.

[27] Z. Wu, H. Zhang, P. Wang, and Z. Sun, "Rtids: A robust transformer-based approach for intrusion detection system," *IEEE Access*, vol. 10, pp. 64 375–64 387, 2022.

[28] R. Zhao, X. Deng, Z. Yan, J. Ma, Z. Xue, and Y. Wang, "Mt-flowformer: A semi-supervised flow transformer for encrypted traffic classification," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 2576–2584.