electronics

MDPI

Article

STC-BERT (Satellite Traffic Classification-BERT): A Traffic Classification Model for Low-Earth-Orbit Satellite Internet Systems

Kexuan Liu, Yasheng Zhang * and Shan Lu

Satellite Communications and Broadcasting Department, 54th Research Institute of China Electronics Technology Group Corporation, Shijiazhuang 050011, China; 18131790215@163.com (K.L.); 17732872762@163.com (S.L.)
* Correspondence: zys@163.com

Abstract: The low-Earth-orbit satellite internet supports the transmission of multiple business types. With increasing business volume and advancements in encryption technology, the quality of service faces challenges. Traditional models lack flexibility in optimizing network performance and ensuring service quality, particularly showing poor performance in identifying encrypted traffic. Therefore, designing a model that can accurately identify multiple business scenarios as well as encrypted traffic with strong generalization capabilities is a challenging issue to resolve. In this paper, addressing the characteristics of diverse low-Earth-orbit satellite traffic and encryption, the authors propose STC-BERT (satellite traffic classification-BERT). During the pretraining phase, this model learns contextual relationships of large-scale unlabeled traffic data, while in the fine-tuning phase, it utilizes a semantic-enhancement algorithm to highlight the significance of key tokens. Post semantic enhancement, a satellite traffic feature fusion module is introduced to integrate tokens into specific low-dimensional scales and achieve final classification in fully connected layers. The experimental results demonstrate our approach's outstanding performance compared to other models: achieving 99.31% (0.2%↑) in the USTC-TFC task, 99.49% in the ISCX-VPN task, 98.44% (0.9%↑) in the Cross-Platform task, and 98.19% (0.8%↑) in the CSTNET-TLS1.3 task.

Keywords: low-Earth-orbit satellite; encrypted traffic; traffic classification; STC-BERT; semantic enhancement; feature fusion

check for updates

Citation: Liu, K.; Zhang, Y.; Lu, S. STC-BERT (Satellite Traffic Classification-BERT): A Traffic Classification Model for Low-Earth-Orbit Satellite Internet Systems. *Electronics* **2024**, *13*, 3933. https://doi.org/10.3390/electronics13193933

Academic Editor: Christos J. Bouras

Received: 26 August 2024
Revised: 26 September 2024
Accepted: 3 October 2024
Published: 4 October 2024

1. Introduction

The architecture of the low-Earth-orbit (LEO) satellite internet system [1] is depicted in Figure 1. Terminals provide users access to satellite internet services, while ground stations manage communication links between satellites and ground networks. The ground control center [2] ensures the satellites operate normally and maintain service quality. This system supports a wide range of services [3], including routine traffic generated by daily web browsing and social interactions; large-scale high-speed data transfers for enterprises conducting big data operations; and rapid transmission of medical imaging data for remote healthcare. Even in special scenarios like emergency responses, the LEO satellite system reliably provides critical traffic transmission services, ensuring timely delivery and processing of information.

To achieve more efficient network management and optimized resource allocation, traffic classification becomes crucial. Satellite traffic classification [4] is a critical technology for maintaining network security and ensuring service quality. It aims to identify various types of traffic from different applications and web pages. Traditional models like deep packet inspection (DPI) rely on application protocol identification [5] and content inspection to statistically analyze traffic behaviors over time, such as traffic flow, business distribution, and top visited websites. However, with the advancement of encryption technologies, malicious traffic [6] enhanced through VPN or Tor encryption evades network monitoring, posing challenges to maintaining service quality.

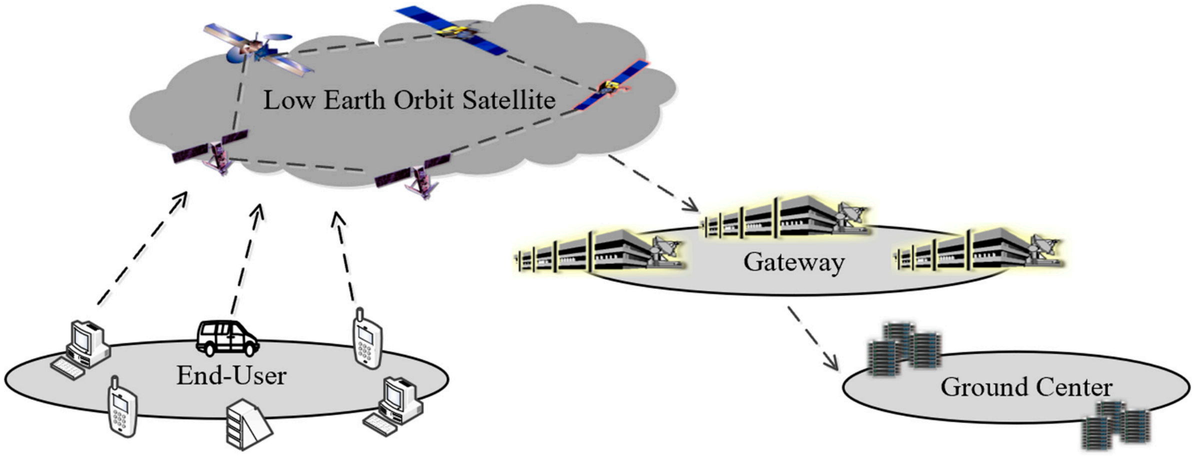


Figure 1. Architecture of low-Earth-orbit satellite internet.

In addressing the accurate identification of encrypted traffic, research in encryption traffic classification has evolved significantly over time. Early studies focused on clustering residual plaintext portions of encrypted traffic to build fingerprint databases [7], considering characteristics like plaintext length and protocols for classification. However, fingerprints can be easily altered in the network, losing their original features. Other studies attempted classification based on statistical features such as traffic size and transmission time [8,9], yet in the complex and diverse traffic information of satellite internet systems, designing universal features for each type of traffic remains challenging. With the development of deep learning, supervised learning approaches that automatically learn traffic feature representations from large labeled datasets [10] have shown promising results in traffic classification. However, these models require high-quality and high-quantity data; deviations and noise in data can severely impact model performance. Additionally, they are vulnerable to adversarial attacks where malicious actors manipulate model classifications through carefully crafted minor disturbances.

In recent years, Transformer-based models [11] have made significant advancements in fields such as natural language processing and computer vision. The self-attention mechanism in Transformers primarily serves to process different parts of input sequences, enabling the model to dynamically focus on various input elements during output generation. This is achieved by computing a weighted sum of queries, keys, and values, thereby enhancing the model's expressive capability and contextual understanding [12]. This robust capacity is a key reason for the effectiveness of Transformer-based models. For instance, large visual models like CLIP and DALL-E leverage Transformers to significantly address issues in multimodal understanding, knowledge transfer, and autonomous driving. ChatGPT resolves challenges related to domain knowledge, natural language understanding, and personalized interaction through its comprehension of text information. Similarly, the Transformer-based natural language model BERT processes input text to extract informational features and understand relationships between contexts and is commonly employed for tasks such as text completion and sentence similarity analysis.

Transformer-based models are adept at understanding relationships between contexts and efficiently and accurately completing diverse tasks across various scenarios. Consequently, we consider whether natural language models can be utilized to accomplish the complex task of traffic classification in low-Earth-orbit satellite internet systems, where resources are constrained. The choice of model must not only account for performance but also consider memory deployment and resource utilization, aiming for a lightweight solution. BERT, which exclusively employs the Transformer encoder, is structurally simpler and has fewer parameters than other natural language models (such as GPT), making it more suitable for deployment in low-Earth systems for traffic classification.

However, existing research on BERT has predominantly focused on natural language processing, such as the efficient performance enhancement of BERT-base on the SQuAD dataset and the excellent results of Tiny-BERT [13] on the GLUE benchmark. In the context of traffic classification within low-Earth-orbit satellite internet systems, BERT must process traffic data rather than natural language. Unlike natural language, traffic characters, while encoded with certain rules, possess a lower overall semantic level and lack the depth of information that humans can comprehend. Some studies [14] have employed BERT for traffic classification to address the representation of traffic inputs, yet they have not modified the model structure specifically for this task. Instead, they continue to apply natural language models to handle traffic inputs, which appears unreasonable and lacks interpretability.

In this paper, addressing the low-semantic nature of traffic inputs, the authors propose STC-BERT (satellite traffic classification-BERT) to perform traffic classification tasks in low-Earth-orbit satellite systems. In the fine-tuning phase, the authors introduce a semantic-enhancement algorithm that precisely extracts traffic features from important tokens embedded in various traffic inputs. Specifically, the model transforms each token into matrix allocation parameters for training, calculates token correlations within traffic clusters, and converts tokens into new vectors as inputs. Furthermore, the authors propose a satellite traffic feature fusion module, integrating multi-dimensional vectors derived from the semantic-enhancement algorithm into this module, to generate unified low-dimensional feature vectors. These low-dimensional feature vectors are then fed into fully connected layers for classification.

2. Research Status

2.1. Traditional Traffic Classification Models

Statistical Features: As shown in Figure 2, Some studies have conducted statistical analysis on encrypted traffic, focusing on packet length, transmission time, transmission rate, and other aspects to extract statistical metrics that reflect traffic characteristics. Common statistical features include packet length, inter-packet transmission time, transmission rate, and flow duration, which can all serve as features for encrypted traffic classification. For instance, the k-nearest neighbors (KNN) algorithm uses packet sizes to train classifiers, while support vector machine (SVM) algorithms utilize time interval features to identify and classify encrypted traffic. However, satellite traffic encompasses diverse types, making it challenging to design universal statistical features for them. Our approach understands intrinsic meaning by learning context relationships from input traffic, without relying on manually designed features.

Deep Learning Models: As shown in Figure 2, Supervised deep learning models are currently mainstream for traffic classification. Unlike statistical feature models, deep learning models automatically extract input features without depending on manually designed features. For example, SwinT-CNN [15] extracts temporal features from both local and global perspectives to achieve traffic classification. I2RNN [16] extracts sequence fingerprints from sessions with local robustness and adapts incrementally to emerging traffic types. Traffic Reconstruction [17] extracts part of the payload as key data, inserts identifiers, and classifies reconstructed data using convolutional neural networks. While these models show certain advantages over traditional machine learning in terms of performance, they rely heavily on the quality and quantity of training data and perform poorly with imbalanced data. Our model achieves excellent performance with minimal specific task data fine-tuning, without depending on large-scale labeled data.

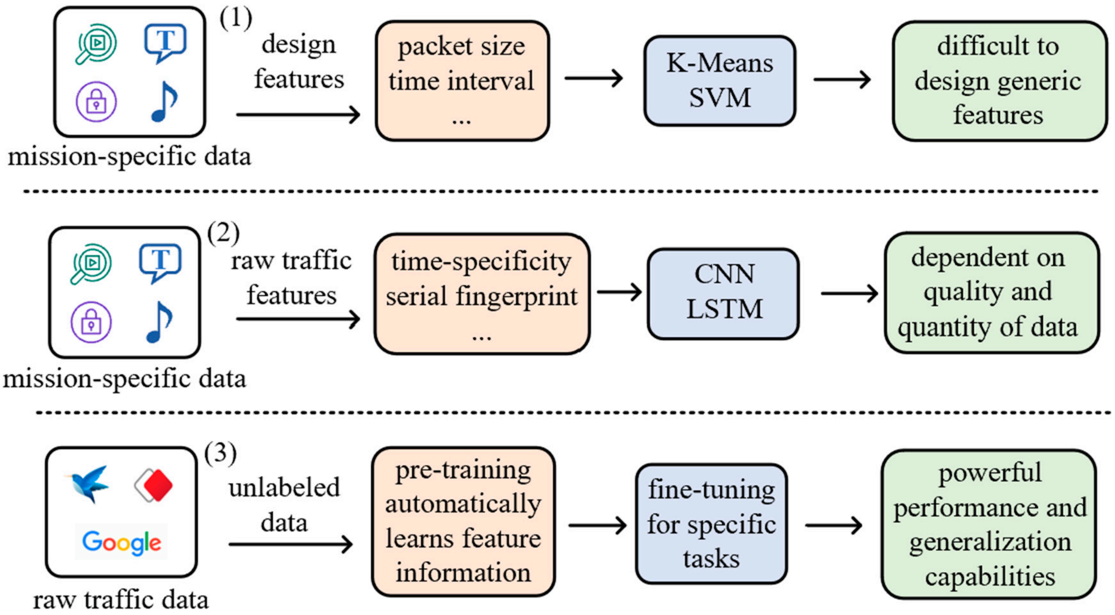

(1)
design features
packet size
time interval
...
K-Means
SVM
difficult to design generic features
mission-specific data
(2)
raw traffic features
time-specificity
serial fingerprint
...
CNN
LSTM
dependent on quality and quantity of data
mission-specific data
(3)
unlabeled data
pre-training automatically learns feature information
fine-tuning for specific tasks
powerful performance and generalization capabilities
Google
raw traffic data

Figure 2. Three traffic classification models: (**1**) Statistical feature models; (**2**) Deep learning models; (**3**) Pretrained models.

2.2. Pretrained Models

Pretrained models based on Transformers: PERT performs pretraining tasks like masked language modeling to predict original word positions, learning contextual relationships for classification tasks. ALBERT [18] significantly reduces model parameters through parameter sharing and introduces SOP to model coherence between sentences, achieving efficient classification. RoBERTa [19] uses dynamic masking during pretraining and removes NSP tasks to shorten training time while enhancing contextual understanding, enabling effective classification. DistilBERT [20] employs knowledge distillation [21] to notably reduce computational requirements and training duration while preserving most of the original model's performance, suitable for classification tasks in resource-constrained scenarios. ET-BERT [22] introduces new pretraining tasks tailored for traffic classification, demonstrating strong performance across tasks without structural improvements. Our approach focuses on semantic analysis of payload segments [23,24] and comprehensive feature extraction [25], utilizing semantic-enhancement algorithms and traffic feature fusion modules to better fit classification tasks.

3. STC-BERT

3.1. Model Architecture

In this section, the objective is to enable STC-BERT to learn various satellite traffic features and perform traffic classification tasks in different scenarios. Thus, we pretrain the model using a large-scale unlabeled traffic dataset and fine-tune the pretrained model for downstream classification tasks specific to particular scenarios. During the fine-tuning stage, the authors devised two modules to enhance the model's feature extraction capabilities. With labeled data for specific tasks, the model predicts their traffic categories.

Traffic characters differ significantly from natural language because, while traffic lacks semantic information understandable by humans, it is encoded according to specific patterns. To enable the model to better comprehend the inherent meanings of the payload section within traffic and achieve accurate traffic classification, the proposed STC-BERT is divided into the following parts, as illustrated in Figure 3: (1) The authors extract raw traffic data from pcap packets and convert these traffic data into tokens for training. (2) During the pretraining phase, the authors use a random masking model to pretrain the model, helping it understand the potential meanings of payloads in context. (3) In the fine-tuning phase, to better fit the classification tasks, the authors propose a semantic-enhancement algorithm and a traffic feature fusion module to help the model better understand contextual relationships. STC-BERT's primary network architecture consists of 12 layers of bidirectional Transformers. Each encoder layer in these layers is used to understand the potential meanings between input traffic characters. Each self-attention layer comprises 12 attention heads, with traffic token embeddings having a dimensionality of H = 768, and the maximum input is limited to 512.

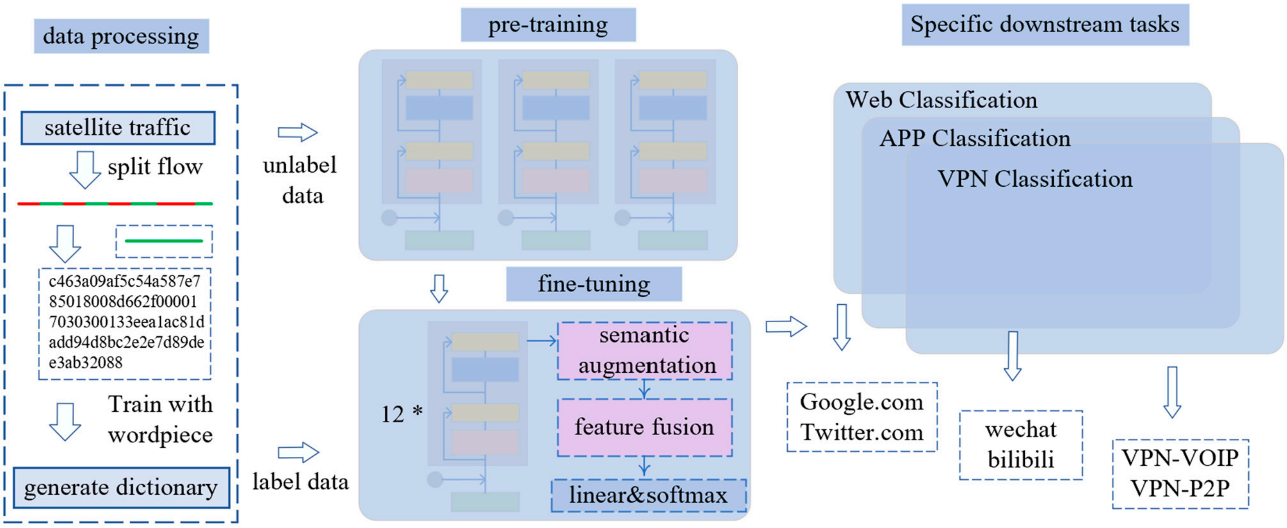


Figure 3. STC-BERT model architecture.

3.2. Data Processing and Traffic Cluster Encoding

3.2.1. Data Extraction

During satellite traffic transmission, traffic data typically include packet timestamps, source and destination information, protocol types, sizes, and other data. As shown in Figure 3, the authors define information composed of five tuples (source IP, port, destination IP, port, and transport layer protocol) and a payload as a cluster unit. This traffic cluster serves as a carrier for information exchange between two terminals. The authors removed data irrelevant to transmission content such as address resolution and dynamic host configuration protocol data. Additionally, they eliminated Ethernet headers, IP addresses, TCP headers, UDP headers, protocol ports, and other information, forcing the model to rely on other data within traffic packets for classification. This approach enables the model to learn relationships between payloads through large-scale unlabeled traffic cluster training rather than depending on specific strong-feature information.

We utilized a wordpiece training dictionary with a size of 65,536. Additionally, special tokens such as [CLS], [PAD], and [MASK] were added. The first token of each embedding is always [CLS], as stipulated by BERT training requirements. However, the authors did not use this token to represent the complete sequence for the classification task, which is discussed in Section 3.4; this section will discuss how all the tokens were trained to achieve classification tasks. [PAD] is a padding symbol that meets minimum length requirements, and [MASK] is used during pretraining, where the model predicts masked tokens to learn contextual associations within traffic.

3.2.2. Traffic Cluster Embedding

In the embedding layer, each token within a traffic cluster is represented by two types of embeddings: cluster embedding and position embedding. In this layer, we take the processed traffic cluster as input with an embedding dimension of H = 768. After passing through N Transformer encoders, we obtain the final token embeddings.

Cluster embedding: This transforms the traffic cluster into a low-dimensional, continuous, real-valued vector. Tokens with related features are closer together in the vector space. The dimensionality H = 768 of the cluster embedding significantly reduces the vector dimensions, thereby reducing computational complexity and storage space.

Position embedding: The information carried by traffic clusters is closely related to their relative positions. Introducing position embedding (dimension H = 768) helps the model capture the sequential relationships of tokens in the input sequence. This embedding is added to the cluster embedding, as depicted in Figure 4, to form the final token embeddings passed to the model.

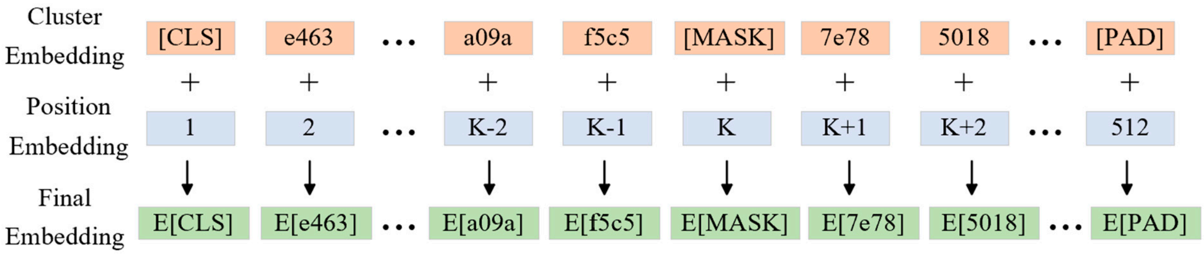


Figure 4. Traffic cluster encoding embeddings.

3.3. Pretraining

In the model's pretraining phase, we employed the Mask Cluster Model (MCM) for the pretraining task. In this task, 15% of tokens are masked. Among these masked tokens, 80% is replaced with [MASK], 10% is replaced with a random token, and the remaining 10% is unchanged. The model learns relationships between traffic clusters by predicting the masked tokens. The authors defined the input traffic cluster as C, where m tokens are randomly masked using negative log-likelihood as the training loss function, defined as follows:

$$L_{MCM} = -\sum_{n=1}^{m} log\left(P\left(MASK_n = token_n \middle| \overline{C}; \varphi\right)\right) \tag{1}$$

φ is a trainable parameter of STC-BERT, and the probability P is generated by Transformer encoders containing φ, representing the input after masking, where the nth token is masked.

For traffic, unlike textual contexts where there are semantic associations between sentences, the authors removed the Next Sentence Prediction (NSP) task from BERT, as indicated by RoBERTa [19], which did not improve model performance and could introduce noise.

Pretraining dataset: 40 GB of unlabeled raw traffic data was selected from the National University of Defense Technology traffic dataset [26,27]. The model learns feature representations of various traffic types through pretraining tasks, which are conducted using unsupervised learning on a large volume of traffic samples. The selected dataset should encompass a substantial and diverse array of raw traffic data. This dataset covers over a hundred applications with ample data per application, avoiding issues of sample imbalance. It is diverse, involving hundreds of different device models and users, various network environments, and application execution paths. The dataset includes rich data features such as packet payloads and loss information.

3.4. Fine-Tuning

After fine-tuning, the model can better perform downstream classification tasks because the data used for pretraining and traffic themselves are agnostic to specific categories, enabling classification of any type of traffic. The models used for fine-tuning and pretraining are structurally similar. We input data specific to the task into the pretrained STC-BERT for training, thereby fine-tuning the model parameters. The semantic-enhancement algorithm involves integrating the information of the entire text into the [CLS] token in the BERT model and using [CLS] as input to a multi-classifier for prediction. However, compressing multi-dimensional information into [CLS] makes it difficult to represent all features of the entire input, especially for low-semantic inputs like traffic clusters, where each token may carry unique meanings.

3.4.1. The Semantic-Enhancement Algorithm

For a traffic cluster $s = \langle a_1, a_2, \ldots, a_n \rangle$, the embedding layer generates a matrix $\mathrm{X} \in \mathbb{R}^{|\mathrm{n}| \times \mathrm{H}}$ where each token corresponds to a H=768-dimensional embedding. The authors measure the autocorrelation of a traffic cluster $S = XX^T$, representing the degree of correlation between the nth token and the mth token in traffic S. They assess the importance of tokens in a traffic cluster from two perspectives:

The maximum value in each row of the correlation matrix is taken as the matching score for that token, which measures the token's association with other tokens.

$$C = Max_{a \in S_m} S_{n,m} \tag{2}$$

To more accurately determine the correlation between tokens, the approach involves calculating the difference between the top two values in each row of the correlation matrix. When multiple important tokens have high matching scores, this method identifies the most semantically significant token among them.

$$M = Max_{a \in S_m} S_{n,m} - 2nd_{a \in S_m} S_{n,m} \tag{3}$$

Adding these two scores yields the semantic score for each token in the sentence.

$$Total\ Score = C + M = 2Max_{a \in S_m} S_{n,m} - 2nd_{a \in S_m} S_{n,m} \tag{4}$$

While scoring measures the correlation between tokens in a traffic flow, uncommon tokens may carry more semantic significance than common ones. For instance, tokens representing protocols often score higher due to their specific roles. However, high-frequency tokens are typically more semantically important, such as payload content following a protocol. Conversely, some low-frequency tokens are semantically unimportant, such as spaces used to separate text content.

To address the issue of mismatched token matching scores and frequencies, we directly train token weights from the text. As illustrated in Figure 5, in the BERT model, the features contained in the [CLS] vector after training represent the analogy of the traffic cluster. BERT fuses the features of each traffic cluster into the [CLS] vector, which is then fed into a fully connected layer classifier for classification. However, due to the semantic dispersion of traffic clusters, they lack the continuity seen in text. Therefore, the authors posit that inputting the entire traffic cluster into the classifier will yield more accurate results than relying solely on the [CLS] vector. Specifically, the authors assign a trainable weight to each token within the traffic cluster, where n represents the final layer embedding of the traffic. This allows the weights to be updated alongside the model parameters through gradient descent, thereby enhancing the model's performance on classification tasks. Tokens that exhibit a strong analogy recognition for traffic will have larger weights, playing a more significant role. The experimental results indicate that models trained using this method are more precise than those relying solely on the [CLS] token.

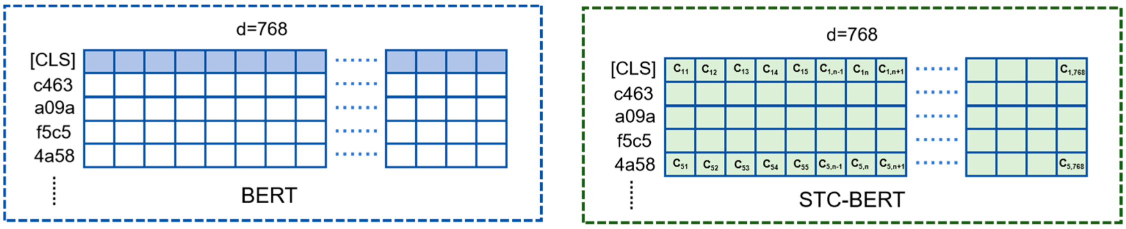

d=768
[CLS]
c463
a09a
f5c5
4a58
BERT
d=768
[CLS]
c463
a09a
f5c5
4a58
C11 C12 C13 C14 C15 C1,n-1 C1n C1,n+1 C1,768
C51 C52 C53 C54 C55 C5,n-1 C5,n C5,n+1 C5,768
STC-BERT

Figure 5. Semantic-enhancement algorithm principle diagram.

3.4.2. Feature Fusion Module

After semantic-enhancement processing, a high-dimensional matrix containing all feature information is fed into a fully connected layer for classification. The authors propose a satellite traffic feature fusion module to capture features at multiple scales and unify inputs to a fixed scale, facilitating classification tasks for the fully connected layer. They introduce convolutional kernels of different sizes kernel_sizes = (3,5,7) with two kernels for each size.

These convolutional kernels perform operations on the input matrix, sliding along the feature dimension of the traffic cluster from top to bottom to extract local feature information. Smaller kernels capture patterns from shorter inputs, while larger kernels can gather dependencies over longer ranges. Through the fusion layer, information from a traffic cluster is compressed into a low-dimensional vector, which is then input into the fully connected layer for classification, as depicted in Figure 6.

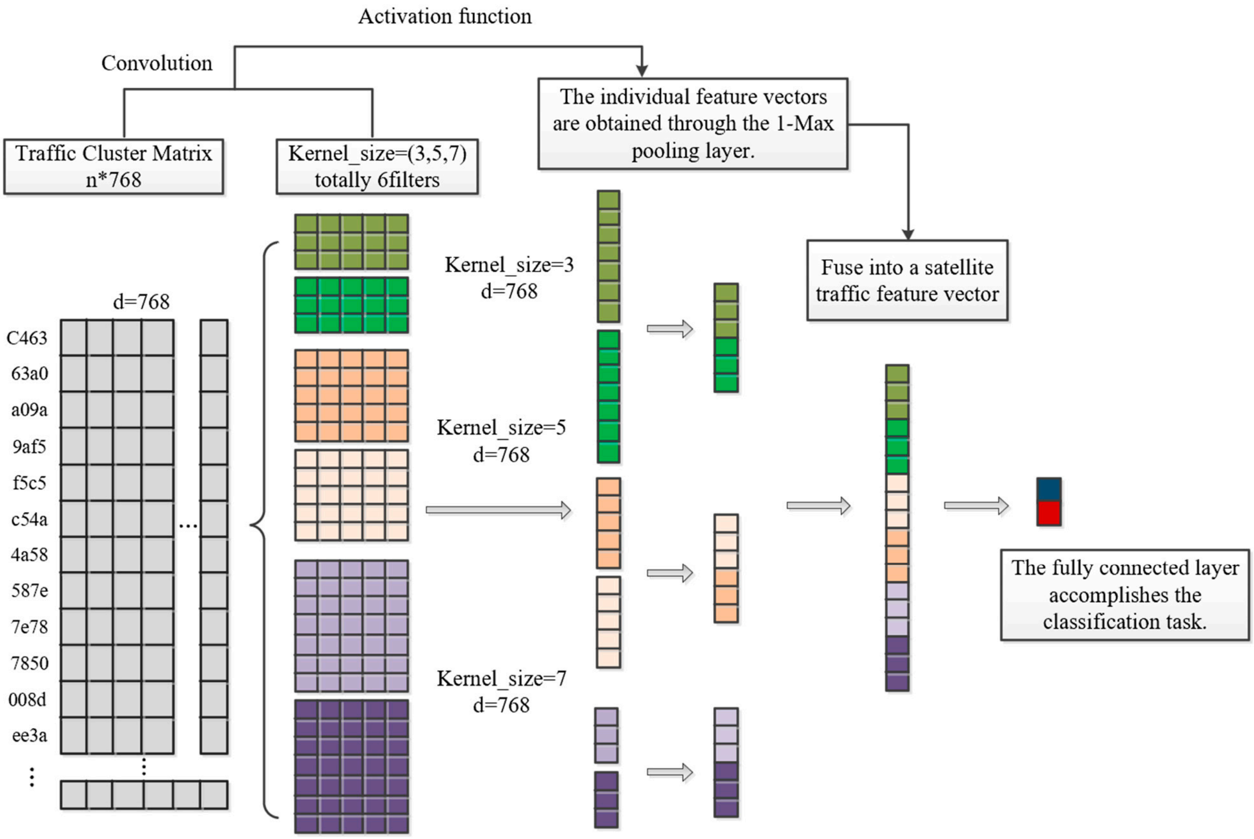

Activation function
Convolution
Traffic Cluster Matrix n*768
Kernel_size=(3,5,7) totally 6filters
The individual feature vectors are obtained through the 1-Max pooling layer.
Fuse into a satellite traffic feature vector
d=768
C463
63a0
a09a
9af5
f5c5
c54a
4a58
587e
7e78
7850
008d
ee3a
Kernel_size=3 d=768
Kernel_size=5 d=768
Kernel_size=7 d=768
The fully connected layer accomplishes the classification task.

Figure 6. Structural diagram of traffic feature fusion module.

4. Experimental Verification

4.1. Evaluation Metrics and Experimental Setup

In this section, the authors conducted four traffic classification tasks and compared STC-BERT with several other models to demonstrate its superior performance. They used heatmaps to show that the semantic-enhancement module achieves more accurate classification by considering all tokens compared to BERT's use of only the [CLS] token. Through ablation experiments, we demonstrated that our proposed semantic-enhancement algorithm and feature fusion module significantly improve model performance on classification tasks.

The authors evaluated the model using four metrics: accuracy (5), precision (6), recall (7), and F1 score (8). Here, TP represents true positives (samples correctly predicted as positive by the model), TN represents true negatives (samples correctly predicted as negative), FP represents false positives (samples incorrectly predicted as positive), and FN represents false negatives (samples incorrectly predicted as negative).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{8}$$

Accuracy primarily measures the overall ability of the model to classify correctly. However, when dealing with imbalanced data, it is crucial not only to consider accuracy but also to focus on the model's ability to classify a specific type of traffic (such as a particular type of malicious traffic). The F1 score evaluates the model's ability to correctly identify positives (e.g., malicious traffic) among all actual positives and the precision of the model in positive predictions. It combines both precision (the proportion of true positives among all positive predictions) and recall (the proportion of true positives among all actual positives), providing a balanced measure that is particularly useful in scenarios with imbalanced classes.

Interstellar traffic does not fundamentally differ in content from terrestrial networks. Therefore, specific datasets were chosen for testing tasks, divided into training, validation, and test sets in an 8:1:1 ratio. The input dimension was 768, with a pretrained batch size of 32, totaling 500,000 steps, using a learning rate of 2×10^{-5}, and a warmup ratio of 0.1. Fine-tuning utilized the AdamW optimizer [28] with a batch size of 32, learning rate of 2×10^{-5}, and 10 epochs. All experiments were conducted using the PyTorch 2.2.2 framework and trained on two NVIDIA RTX 3090 GPUs.

4.2. Comparison with Existing Models

This section compares STC-BERT with a few other models, using deep packet inspection (DPI) technology [29] to analyze payload content, including PERT, which employs permutation-based language modeling for pretraining tasks, and ET-BERT [22], specifically designed for encrypted traffic pretraining tasks.

DPI is a network traffic analysis technology that identifies, classifies, and manages network traffic by inspecting the contents and metadata of data packets. Compared to traditional port- or protocol-based traffic classification methods, DPI offers more precise traffic identification. This method analyzes not only the header information of packets (such as source and destination IP addresses and port numbers) but also delves into the payload of the packets to examine their contents for identifying applications and protocols. DPI technology is an indispensable tool in modern network management and security strategies.

PERT is a Transformer-based pretrained model, with a pretraining task that differs from BERT's masked language modeling. It models the sequence of packets to capture temporal features within the traffic, helping to identify specific applications and protocols. This approach effectively extracts key features from traffic, such as packet size, inter-arrival time, and sequence numbers, which assist in distinguishing between different types of traffic.

ET-BERT was the first to propose using BERT for encrypted network traffic classification, introducing two pretraining tasks aimed at traffic classification: the same-origin burst mask and same-origin burst prediction. By converting pcap-format traffic information into text inputs for training, ET-BERT improves classification accuracy for various types of encrypted traffic and enhances network security.

The pretrained models demonstrate the advantages of Transformer-based pretrained models in understanding input context through comparisons with DPI. Both ET-BERT and STC-BERT utilize random masking for pretraining, showing through comparisons with the PERT model that masked language modeling can better assist the model in understanding latent meanings among traffic features. STC-BERT, when compared with ET-BERT, demonstrates that the semantic-enhancement algorithm and satellite traffic feature fusion module can better help the model learn the relationships among traffic data, achieving classification tasks more efficiently and accurately.

Task [1]: In low-Earth-orbit satellite internet, the propagation of network attacks, spam, and malware can lead to data theft, network system disruption, and serious harm. The USTC-TFC dataset [30], provided by the University of Science and Technology of China, contains multiple types of malicious traffic. Due to the presence of payload-related malicious traffic in plaintext within this dataset, DPI significantly improves the model's classification accuracy by analyzing plaintext content. However, as shown in Table 1, Transformer-based pretrained models achieve superior performance on this task without relying on plaintext information. STC-BERT achieves an accuracy of 99.31%.

Table 1. Performance metrics of four models on USTC-TFC task.

Dataset	**USTC-TFC**			
Models	**AC**	**PR**	**RC**	**F1**
DPI	0.9640	0.9650	0.9631	0.9641
PERT	0.9909	0.9911	0.9910	0.9911
ET-BERT	0.9915	0.9915	0.9916	0.9916
STC-BERT	0.9931	0.9928	0.9878	0.9903

Task [2]: The ISCX-VPN-non-VPN dataset [31] is used for researching and developing techniques in network traffic analysis and classification. It includes 15 types of VPN encrypted traffic and 15 types of non-VPN traffic, categorized into 12 classes with corresponding labels. This dataset covers six types of regular encrypted traffic and six types of non-VPN encrypted traffic. Effectively classifying VPN traffic allows for a better understanding of network traffic distribution and composition, thereby optimizing network performance and enhancing security. When users establish a VPN service, an encrypted channel is formed between the user and the VPN server. The VPN server replaces the user's IP with its own server IP to send requests and transfers response data back to the user. Due to the complexity of encryption types, traditional models relying on IP and MAC features perform poorly in classification. As shown in Table 2, STC-BERT achieves classification by understanding the information between payloads, improving accuracy by approximately 16% compared to PERT. While there exists a slight difference in accuracy compared to ET-BERT, STC-BERT demonstrates superior precision and recall rates over ET-BERT, effectively fulfilling the task of VPN traffic classification.

Table 2. Performance metrics of four models on ISCX-VPN task.

Dataset	**ISCX-VPN**			
Models	**AC**	**PR**	**RC**	**F1**
DPI	0.9758	0.9785	0.9745	0.9765
PERT	0.8229	0.7092	0.7173	0.6992
ET-BERT	0.9962	0.9936	0.9938	0.9937
STC-BERT	0.9949	0.9937	0.9951	0.9944

Task [3]: The Cross-Platform dataset [32] consists of data generated by 215 Android and 196 iOS applications' users. These apps were all sourced from the top 100 rankings on the App Store and Google Play in the United States, China, and India. This task aims to classify data under encryption protocols. The dataset contains complete payload information, allowing models trained on this task to learn characteristic patterns of traffic transmission structures. This enhances the representation of features and structures in encrypted traffic data. Due to the introduction of semantic-enhancement algorithms and feature fusion modules, the model excels in semantic understanding compared to other models. As shown in Table 3, STC-BERT outperforms ET-BERT in all performance aspects, achieving an accuracy of 98.44%.

Table 3. Performance metrics of four models on Cross-Platform task.

Dataset	**Cross-Platform**			
Models	**AC**	**PR**	**RC**	**F1**
DPI	0.8805	0.8004	0.7567	0.8138
PERT	0.9772	0.8628	0.8591	0.8550
ET-BERT	0.9728	0.9439	0.9119	0.9206
STC-BERT	0.9844	0.9725	0.9402	0.9561

Task [4]: The CSTNET-TLS 1.3 dataset [22] comprises data collected from 120 applications under the CSTNET network from March to July 2021. This dataset is the first focused on the TLS 1.3 protocol and includes applications sourced from the Alexa Top 5000 list that deploy TLS 1.3. Each session flow is labeled using Server Name Indication (SNI). TLS 1.3 represents new encryption technology that enhances transmission security, thereby increasing the difficulty of identification. However, as shown in Table 4, STC-BERT learns implicit feature representations of the TLS 1.3 protocol through traffic payload analysis. This approach achieves more precise classification on this task, with an accuracy of 98.19%.

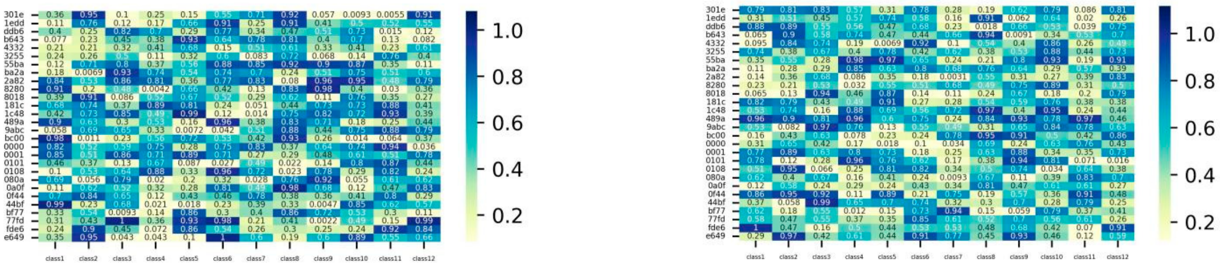
Table 4. Performance metrics of four models on CSTNET-TLS 1.3 task.

Dataset	**CSTNET-TLS 1.3**			
Models	**AC**	**PR**	**RC**	**F1**
DPI	0.8019	0.4315	0.2689	0.4022
PERT	0.8915	0.8846	0.8719	0.8741
ET-BERT	0.9737	0.9742	0.9742	0.9741
STC-BERT	0.9819	0.9770	0.9711	0.9740

In Task [2], the authors selected a cluster of traffic from category 8, out of 12 categories, and observed semantic scores after semantic enhancement. According to Figure 7, scores for tokens related to specific transmission content in the headers and middle parts of the traffic increased significantly. This indicates that these tokens play a crucial role in feature extraction, which is why STC-BERT achieves significant effectiveness in various classification tasks.

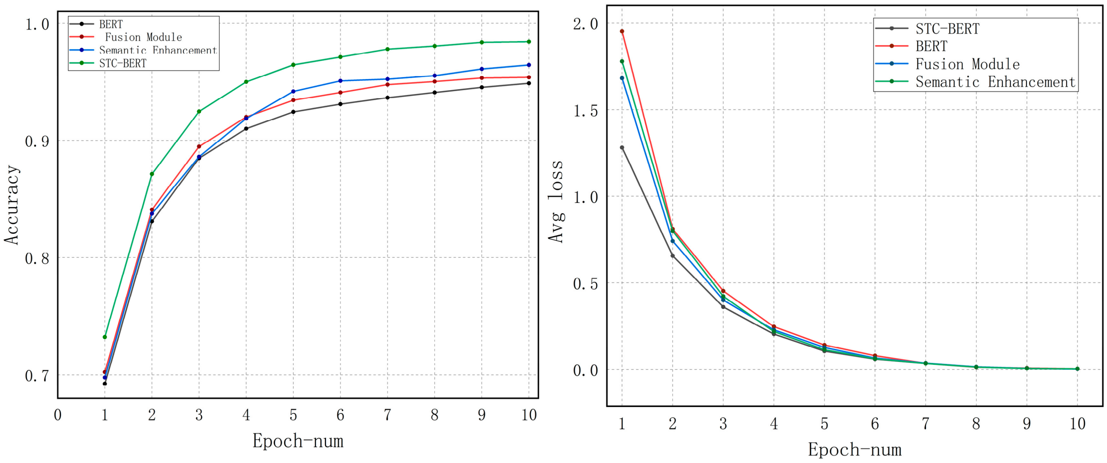
4.3. Ablation Experiments

In this section, the authors conducted ablation experiments to validate the impact of semantic-enhancement algorithms and feature fusion modules on model performance. Using Task [4] as an example, comparisons were made between STC-BERT, BERT, BERT-Semantic Enhancement, and BERT-Feature Fusion in terms of classification accuracy and training loss, as depicted in Figure 8.



(**a**) (**b**)

Figure 7. (**a**) shows the score of each token when STC-BERT recognizes a category 8 flow while processing a VPN task, (**b**) is the score for each token when recognizing a category 8 flow when processing the VPN task after adding the semantic-enhancement algorithm by STC-BERT.



(**a**) (**b**)

Figure 8. (**a**) is the variation in accuracy with the number of iterations for BERT and BERT with the addition of the semantic-enhancement algorithm and feature fusion module, respectively, and for STC-BERT training. (**b**) is the variation in loss with the number of iterations for BERT and BERT with the addition of the semantic-enhancement algorithm and feature fusion module, respectively, and STC-BERT training.

The experimental results showed that when applying the semantic-enhancement algorithm to BERT, the accuracy plateaued after 10 iterations, resulting in a 3.1% increase compared to BERT. Similarly, when using the feature fusion module with BERT, after 10 iterations, the accuracy stabilized with a 2.15% improvement over BERT. STC-BERT, which integrates both the semantic-enhancement algorithm and the feature fusion module, achieved a 5.6% improvement in accuracy compared to BERT, with faster reduction in training loss. Additionally, the F1 score increased by 6.32% after ten iterations, indicating that the semantic-enhancement algorithm and feature fusion module contribute distinct yet complementary traffic features. The semantic-enhancement algorithm emphasizes the importance of individual tokens by reallocating training weight parameters, helping the model learn contextual associations within payload contexts and enhancing its ability to capture features of traffic clusters, thereby achieving better performance in specific tasks. Meanwhile, the feature fusion module extracts information at different scales, improving the model's handling of relationships before and after entire traffic clusters and enhancing classification accuracy.

In Figure 8, we observe that STC-BERT achieves significantly higher accuracy compared to BERT and BERT with the addition of the semantic-enhancement algorithm and satellite traffic feature fusion module. Additionally, the loss decreases more rapidly during training. This clearly indicates that using the token representations of the entire traffic flow for classification is more accurate than relying solely on the [CLS] vector. It also highlights the advantages of multi-scale convolution in helping the model deeply extract traffic features.

5. Discussion

In this section, we discuss some limitations of STC-BERT as well as related research. System Applicability: With the development of low-Earth-orbit satellite internet communication technologies, the diversification of service content leads to increased variability and complexity of traffic. This poses challenges to our fixed data processing workflows. For instance, in some real-world systems, protocols and payloads may be encapsulated together, and the emergence of new encryption methods makes it difficult to label encrypted data using traditional methods. Training Complexity: In low-Earth-orbit satellite internet systems, computational resources are often limited. Currently, training is conducted by deploying servers at ground stations, which incurs high time costs. It is crucial to explore ways to reduce training complexity or compress model size while maintaining performance. Update and Iteration Issues: Given the high training time costs, when new traffic content appears, it is important to consider whether the model can continue training on the existing foundation without the need for retraining from scratch. Security: BERT-based models rely on the Transformer architecture, which has high requirements for training samples, necessitating the most original and pure data. If the training samples contain vocabulary deliberately set by attackers, the model may learn incorrect information and fall into traps, while the semantic-enhancement algorithm may misinterpret this as significant information. This is a concern that requires attention, and future research could focus on strategies to prevent such situations from occurring.

6. Conclusions

This article presents a traffic classification model called STC-BERT, designed for low-Earth-orbit satellite internet systems. This model is based on the Transformer architecture and learns the implicit relationships within traffic clusters through pretraining. During the fine-tuning phase, we introduce a semantic-enhancement algorithm tailored for low-semantic traffic inputs. Unlike BERT, STC-BERT does not classify based on the category represented by the [CLS] vector; instead, it uses the final layer embeddings of each traffic cluster as input to a fully connected layer for classification. We propose a satellite traffic feature fusion module that deeply extracts important traffic features from various dimensions. Both the semantic-enhancement algorithm and the satellite traffic feature fusion module are designed to address the characteristics of diverse and encrypted traffic in low-Earth-orbit satellite internet systems. STC-BERT achieved an accuracy of 99.31% on the USTC-TFC dataset, which includes malicious traffic, helping the satellite internet system identify anomalous traffic or potential security threats, such as DDoS attacks, and take timely protective measures to enhance network security. On the ISC-VPN dataset, which contains VPN encrypted traffic, the accuracy reached 99.49%, and on the Cross-Platform dataset, the accuracy was 98.44%. This allows low-Earth-orbit satellite internet systems to provide a more refined quality of service, setting priorities for different types of traffic, thereby reducing latency and packet loss, ultimately enhancing user experience. On the CSTNET-TLS 1.3 dataset, which includes a large volume of the latest TLS protocols, the accuracy reached 98.19%. This capability supports various application scenarios for low-Earth-orbit satellite internet, such as agricultural monitoring, smart cities, and telemedicine, ensuring effective operation across these applications. Given that low-Earth-orbit satellite networks are significantly affected by ground conditions and satellite positioning, traffic classification can greatly assist the network in dynamically adjusting to the ever-changing network environment, ensuring connection stability. Additionally, STC-BERT achieved F1 scores of 99.03%, 99.44%, 95.61%, and 97.40% across the four aforementioned datasets, demonstrating its strong capability in handling imbalanced sample data, which is particularly helpful in identifying malicious traffic. When only a small amount of specific data is present in the training samples, STC-BERT can accurately identify such specific traffic during inference, which is crucial for network defense in low-Earth-orbit satellite internet systems. The model effectively protects the network environment, which we believe benefits from the semantic-enhancement algorithm, as certain tokens representing strong features played a significant role in helping the model perform downstream tasks more effectively.

Overall, STC-BERT can handle complex traffic classification scenarios within low-Earth-orbit satellite internet systems effectively. However, we have not explored more lightweight models. Currently, STC-BERT can only be trained on servers, with the inference model deployed at ground stations for traffic identification. If inference is to be conducted on satellites, developing a more lightweight model will be a challenging task. The authors have not investigated how well STC-BERT generalizes, particularly whether the inference model can accurately identify a new type of malicious traffic [33] that it has not been trained on. This is an important issue for network security and could serve as a focal point for future research. The authors believe that even if new types of traffic cannot be accurately identified, at least appropriate alerts should be generated to help network administrators recognize the traffic.

Author Contributions: Conceptualization, K.L., Y.Z., and S.L.; Methodology, K.L.; Software, K.L.; Validation, K.L.; Formal analysis, K.L., Y.Z., and S.L.; Investigation, K.L. and S.L.; Data curation, K.L.; Writing—original draft, K.L.; Writing—review and editing, K.L., Y.Z., and S.L.; Visualization, K.L.; Supervision, Y.Z. and S.L.; Project administration, K.L. and S.L.; Funding acquisition, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Key Research and Development Program of China (2022YFB2902505).

Data Availability Statement: Data are contained within this article.

Acknowledgments: I (Kexuan Liu) would like to thank Yasheng Zhang and Shan Lu for their support and guidance during my research over the past six months. I also appreciate the valuable feedback from the editors and reviewers on this article. I wish everyone all the best.

Conflicts of Interest: The company was not involved in the study design, collection, analysis, interpretation of data, the writing of this article or the decision to submit it for publication. The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Ying, T.; Wenting, J.; Zhao, G.; Wei, Z.; Zihe, G. Current Situation and Development Prospect of Satellite Internet. *Int. Space* **2024**, *5*, 57–63.
2. Zorzi, M.; Zanella, A.; Testolin, A.; Grazia, M.D.F.D.; Zorzi, M. Cognition-Based Networks: A New Perspective on Network Optimization Using Learning and Distributed Intelligence. *IEEE Access* **2015**, *3*, 1512–1530. [CrossRef]
3. Centenaro, M.; Costa, C.E.; Granelli, F.; Sacchi, C.; Vangelista, L. A Survey on Technologies, Standards and Open Challenges in Satellite IoT. *IEEE Commun. Surv. Tutorials* **2021**, *23*, 1693–1720. [CrossRef]
4. Bu, Z.; Zhou, B.; Cheng, P.; Zhang, K.; Ling, Z.-H. Encrypted Network Traffic Classification Using Deep and Parallel Network-in-Network Models. *IEEE Access* **2020**, *8*, 132950–132959. [CrossRef]
5. Zhou, P. Research and Design of P2P Traffic Detection System Based on DPI and DFI. Master's Thesis, Guangxi University, Nanning, China, 2015.
6. Zang, X.; Wang, T.; Zhang, X.; Gong, J.; Gao, P.; Zhang, G. Encrypted malicious traffic detection based on natural language processing and deep learning. *Comput. Netw.* **2024**, *250*, 110598. [CrossRef]
7. Yu, L.; Tao, J.; Xu, Y.; Sun, W.; Wang, Z. TLS fingerprint for encrypted malicious traffic detection with attributed graph kernel. *Comput. Netw.* **2024**, *247*, 110475. [CrossRef]
8. Yan, H.; He, L.; Song, X.; Yao, W.; Li, C.; Zhou, Q. Bidirectional Statistical Feature Extraction Based on Time Window for Tor Flow Classification. *Symmetry* **2022**, *14*, 2002. [CrossRef]
9. Zhao, S.; Chen, S.; Wei, Z. Statistical Feature-Based Personal Information Detection in Mobile Network Traffic. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 5085200. [CrossRef]
10. Wang, L.; Che, L.; Lam, K.-Y.; Liu, W.; Li, F. Mobile traffic prediction with attention-based hybrid deep learning. *Phys. Commun.* **2024**, *66*, 102420. [CrossRef]

11. Ke, A.; Luo, J.; Cai, B. UNet-like network fused swin transformer and CNN for semantic image synthesis. *Sci. Rep.* **2024**, *14*, 16761. [CrossRef] [PubMed]
12. Min, L.; Fan, Z.; Dou, F.; Sun, J.; Luo, C.; Lv, Q. Adaption BERT for Medical Information Processing with ChatGPT and Contrastive Learning. *Electronics* **2024**, *13*, 2431. [CrossRef]
13. Jiao, X.; Yin, Y.; Shang, L.; Jiang, X.; Chen, X.; Li, L.; Wang, F.; Liu, Q. TinyBERT: Distilling BERT for Natural Language Understanding. *arXiv* **2019**, arXiv:1909.10351. [CrossRef]
14. 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; pp. 1–8. [CrossRef]
15. Wang, Y.; Gao, Y.; Li, X.; Yuan, J. Encrypted Traffic Classification Model Based on SwinT-CNN. In Proceedings of the 2023 4th International Conference on Computer Engineering and Application (ICCEA), Hangzhou, China, 7–9 April 2023; pp. 138–142. [CrossRef]
16. Song, Z.; Zhao, Z.; Zhang, F.; Xiong, G.; Cheng, G.; Zhao, X.; Guo, S.; Chen, B. I^2 RNN: An Incremental and Interpretable Recurrent Neural Network for Encrypted Traffic Classification. *IEEE Trans. Dependable Secur. Comput.* **2023**. [CrossRef]
17. Ma, Q.; Huang, W.; Jin, Y.; Mao, J. Encrypted Traffic Classification Based on Traffic Reconstruction. In Proceedings of the 2021 4th International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, China, 28–31 May 2021; pp. 572–576. [CrossRef]
18. Zhang, X.; Ma, Y. An ALBERT-based TextCNN-Hatt hybrid model enhanced with topic knowledge for sentiment analysis of sudden-onset disasters. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106136. [CrossRef]
19. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692. [CrossRef]
20. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108. [CrossRef]
21. Tian, X.; Zhang, Z.; Lin, S.; Qu, Y.; Xie, Y.; Ma, L. Farewell to Mutual Information: Variational Distillation for Cross-Modal Person Re-Identification. *arXiv* **2021**, arXiv:2104.02862. [CrossRef]
22. 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, Lyon, France, 25–29 April 2022; pp. 633–642.
23. Wang, H.; Yu, D. Going Beyond Sentence Embeddings: A Token-Level Matching Algorithm for Calculating Semantic Textual Similarity. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Toronto, Canada, 9–14 July 2023; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 563–570.
24. Li, B.; Zhou, H.; He, J.; Wang, M.; Yang, Y.; Li, L. On the Sentence Embeddings from Pre-trained Language Models. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 9119–9130.
25. Kim, Y. Convolutional Neural Networks for Sentence Classification. *arXiv* **2014**, arXiv:1408.5882. [CrossRef]
26. Zhao, S.; Chen, S.; Wang, F.; Wei, Z.; Zhong, J.; Liang, J. A Large-Scale Mobile Traffic Dataset for Mobile Application Identification. *Comput. J.* **2023**, *67*, 1501–1513. [CrossRef]
27. Zhao, S.; Zhong, J.; Chen, S.; Liang, J. Comprehensive Mobile Traffic Characterization Based on a Large-Scale Mobile Traffic Dataset. In *Network and System Security*; NSS 2022. Lecture Notes in Computer Science, vol 13787; Springer: Cham, Switzerland, 2022. [CrossRef]
28. Zhou, P.; Xie, X.; Lin, Z.; Yan, S. Towards Understanding Convergence and Generalization of AdamW. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 6486–6493. [CrossRef]
29. Lotfollahi, M.; Siavoshani, M.J.; Zade, R.S.H.; Saberian, M. Deep packet: A novel approach for encrypted traffic classification using deep learning. *Soft Comput.* **2019**, *24*, 1999–2012. [CrossRef]
30. 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. [CrossRef]
31. Draper-Gil, G.; Lashkari, A.H.; Mamun, M.S.I.; Ghorbani, A.A. Characterization of Encrypted and VPN Traffic Using Time-Related Features. In Proceedings of the International Conference on Information Systems Security and Privacy (ICISSP), Rome, Italy, 19–21 February 2016. [CrossRef]
32. 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, San Diego, CA, USA, 23–26 February 2020. [CrossRef]
33. Liu, X.; Liu, Z.; Zhang, Y.; Zhang, W.; Lv, D.; Zhou, Q. TCN enhanced novel malicious traffic detection for IoT devices. *Connect. Sci.* **2022**, *34*, 1322–1341.

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.