applied sciences

Article

TransECA-Net: A Transformer-Based Model for Encrypted Traffic Classification

Ziao Liu, Yuanyuan Xie, Yanyan Luo, Yuxin Wang and Xiangmin Ji *

School of Computer and Information, Fujian Agriculture and Forestry University, Fuzhou 350002, China;
1221193004@fafu.edu.cn (Z.L.); 5221139031@fafu.edu.cn (Y.X.); 5221139007@fafu.edu.cn (Y.L.);
wyx1632025@163.com (Y.W.)
* Correspondence: jixm168@126.com

Featured Application: Classification of encrypted traffic.

Abstract: Encrypted network traffic classification remains a critical component in network security monitoring. However, existing approaches face two fundamental limitations: (1) conventional methods rely on manual feature engineering and are inadequate in handling high-dimensional features; and (2) they lack the capability to capture dynamic temporal patterns. This paper introduces TransECA-Net, a novel hybrid deep learning architecture that addresses these limitations through two key innovations. First, we integrate ECA-Net modules with CNN architecture to enable automated feature extraction and efficient dimension reduction via channel selection. Second, we incorporate a Transformer encoder to model global temporal dependencies through multi-head self-attention, supplemented by residual connections for optimal gradient flow. Extensive experiments on the ISCX VPN-nonVPN dataset demonstrate the superiority of our approach. TransECA-Net achieved an average accuracy of 98.25% in classifying 12 types of encrypted traffic, outperforming classical baseline models such as 1D-CNN, CNN + LSTM, and TFE-GNN by 6.2–14.8%. Additionally, it demonstrated a 37.44–48.84% improvement in convergence speed during the training process. Our proposed framework presents a new paradigm for encrypted traffic feature disentanglement and representation learning. This paradigm enables cybersecurity systems to achieve fine-grained service identification of encrypted traffic (e.g., 98.9% accuracy in VPN traffic detection) and real-time responsiveness (48.8% faster than conventional methods), providing technical support for combating emerging cybercrimes such as monitoring illegal transactions on darknet networks and contributing significantly to adaptive network security monitoring systems.

Keywords: convolutional neural network; deep learning; encrypted traffic; traffic classification; transformer

check for updates

Academic Editor: Stefan Fischer

Received: 12 February 2025
Revised: 2 March 2025
Accepted: 4 March 2025
Published: 10 March 2025

Citation: Liu, Z.; Xie, Y.; Luo, Y.; Wang, Y.; Ji, X. TransECA-Net: A Transformer-Based Model for Encrypted Traffic Classification. *Appl. Sci.* **2025**, *15*, 2977. https://doi.org/10.3390/app15062977

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Network traffic classification [1] is a technology that enables intelligent traffic categorization by parsing network data flows. Its implementation process involves key stages such as packet capture, protocol parsing, and traffic feature extraction [2]. The core objectives of this technology lie in enhancing network behavior monitoring [3], traffic management, and security protection capabilities [4], playing a crucial role in scenarios such as intrusion detection [5], malware identification, and traffic analysis [6]. Recent studies have confirmed its effectiveness in encrypted traffic feature extraction [7] and Quality of Service (QoS)-aware resource allocation [8] within software-defined networking (SDN) environments, thereby significantly enhancing network-architecture optimization efficiency. The emergence of hybrid encryption systems—integrating the efficacy of symmetric encryption with the security provisions of public-key encryption—coupled with hashing functions and digital signature technologies has established a multi-layered security defense mechanism for contemporary network communications.

It is imperative to recognize that the widespread implementation of encryption technologies [9] has given rise to new security challenges, as malicious actors exploit privacy-enhancing tools for illicit transactions conducted on the dark web, including arms and drug trafficking, thereby complicating the task of traffic tracing considerably.

Traditional classification methodologies encounter three principal limitations: port/protocol-based methods have become obsolete in the face of dynamic port technologies; Deep Packet Inspection (DPI) [10] grapples with issues of privacy violation and high computational costs; and statistical feature classification is heavily reliant on extensive volumes of labeled data. While machine-learning approaches [11,12], such as Decision Trees and Random Forests, overcome the dependency on protocols, they remain constrained by the financial burden associated with the acquisition of labeled data. Although deep learning technologies [13] facilitate automatic feature extraction, they still encounter challenges relating to the consumption of substantial computational resources and limited accuracy in the classification of encrypted traffic. To address these challenges, this paper introduces the TransECA-Net model, which innovatively integrates CNN, Transformer, and ECANet [14] architectures. The model employs an ECANet-enhanced CNN module to facilitate automatic feature extraction and efficient dimensionality reduction, incorporates a Transformer to establish comprehensive global spatiotemporal relationships, and utilizes residual connections to promote optimal gradient flow. In the experimental section, we leverage the publicly available ISCX dataset to juxtapose TransECA-Net against four other models. The findings from the experiments indicate that TransECA-Net surpasses almost all other models, achieving superior performance across the datasets utilized (including a classification accuracy enhancement of 3.2% and a convergence acceleration of 37.4%).

- A novel hybrid architecture integrating local feature enhancement with global interaction is proposed.
- A tailored solution for classifying encrypted traffic is developed, effectively overcoming the data dependency and processing limitations characteristic of traditional methods.
- The model markedly enhances efficiency and generalization capabilities for the high-precision, real-time classification of non-VPN encrypted traffic.

2. Related Work

In the realm of network security, the classification of encrypted network traffic has consistently attracted significant attention. Traditional methods predominantly depend on manually crafted features and basic machine-learning algorithms for classification. In this section, we delineate several principal methodologies for network traffic classification.

2.1. Port-Based Methods

The early classification of network traffic predominantly hinged on port numbers and protocol analysis. In an era when network applications were limited and relatively static, port-based approaches demonstrated efficacy. In 2009, Yoon et al. [15] introduced a technique for classifying application traffic leveraging fixed IP-port information, which involved matching packet headers with the aggregated fixed IP-port data. Nevertheless, as network technologies have evolved, malicious actors have developed methods to obscure their true traffic types by modifying traffic ports, thereby circumventing detection mechanisms that rely on fixed port classifications. Furthermore, port-based classification methodologies primarily rely on the identification of port number characteristics, neglecting other significant attributes such as packet content and protocol headers. Consequently, the classification capacity of this approach is inherently constrained and fails to fully capitalize on the extensive information embedded within network traffic.

2.2. Protocol-Analysis-Based Methods

Velan [16] and Stevanovic et al. [17] employed Deep Packet Inspection (DPI) to scrutinize both packet headers and payloads, thereby analyzing the distinctive characteristics of specific protocols for classification purposes. This methodology is renowned for its high accuracy and its ability to identify non-standard applications that utilize standard ports. However, it is accompanied by considerable computational overhead and faces challenges in accommodating emerging protocols and encrypted traffic. Furthermore, rising concerns surrounding user privacy have precipitated the gradual decline in the utilization of this approach.

2.3. Feature-Based Methods

As network applications have diversified and become more complex, particularly with the rise of P2P and encrypted traffic, traditional port-based methods have gradually proven ineffective, resulting in the emergence of traffic-feature-based strategies. Huang et al. [18] introduced an internet classification method that utilizes statistical features, effectively overcoming the operational difficulties associated with unreliable port numbers and the complexities of payload interpretation. However, this approach necessitates a large volume of training data, and the processes of feature selection and extraction can be quite intricate.

2.4. Machine-Learning-Based Methods

During the 2010s, there was a notable emergence of machine-learning methodologies in the domain of network traffic classification, wherein researchers systematically implemented traditional machine-learning algorithms. Prominent algorithms in this context encompass Decision Trees, Random Forests, Support Vector Machines (SVMs) [11], and K-Nearest Neighbors (KNN) [19]. These methodologies exhibit a high degree of versatility and possess the capability to manage intricate and ever-evolving network traffic. Nonetheless, they necessitate an extensive volume of labeled data for effective training, and the precision of the resulting models is contingent upon the quality of the training datasets utilized.

2.5. Deep-Learning-Based Methods

Since the 2020s, deep learning technologies have matured further, particularly with the application of convolutional neural networks (CNNs) and recurrent neural networks (RNNs), significantly enhancing the accuracy and efficiency of traffic classification. Wang et al. [20] first proposed an end-to-end encrypted traffic classification method based on one-dimensional CNNs in 2017, which integrates feature extraction, feature selection, and classification into a unified framework. Lotfollahi [21] used a method combining Stacked Autoencoders (SAEs) and CNN, merging the feature extraction and classification stages into a single system. Wang subsequently introduced HAST-IDS, an intrusion detection system based on hierarchical spatiotemporal features [22], which employs CNNs to learn low-level spatial features of network traffic and Long Short-Term Memory networks (LSTMs) to capture high-level temporal features. In 2023, Hu et al. [23] proposed a network traffic classification method that introduces an attention mechanism into the CNN–LSTM framework, embedding SENet to weight and redistribute high-order spatial features, thereby extracting key spatial features from the network flow. Additionally, Zhang et al. [24] introduced a model named TFE-GNN (temporal fusion encoder using graph neural networks) in

2023, aimed at achieving the fine-grained classification of encrypted traffic through graph neural networks.

2.6. Comparison of GNNs and Transformers for Sequence-Aware Classification

In the field of encrypted traffic classification, graph neural network (GNN)-based methods typically construct packet nodes and temporal-relationship edges to capture traffic interaction features. However, these methods face three key challenges in practical applications: First, static graph structures struggle to effectively characterize the dynamic temporal characteristics and implicit correlations of network traffic, potentially introducing noise or erroneous assumptions. Second, the local perception nature of traditional neighborhood-aggregation mechanisms limits their ability to model global dependencies across sessions. Third, large-scale traffic graphs significantly increase the computational complexity of neighborhood aggregation, making real-time classification challenging.

By contrast, Transformer-based architectures demonstrate distinct advantages: Their inherent sequence-modeling capabilities naturally align with the temporal characteristics of traffic data, enabling the direct processing of raw packet sequences or statistical features without manual graph construction. The self-attention mechanism captures long-range behavioral patterns in encrypted traffic through the parallel computation of dependencies between arbitrary positions, while the multi-head attention mechanism adapts to heterogeneous feature spaces via multidimensional dynamic feature interactions. This synergistic mechanism of global context awareness and dynamic feature fusion provides a more robust solution for encrypted traffic classification.

2.7. Transformer Adaptation Strategies for Limited Data

Nevertheless, overfitting remains challenging when applying Transformer architectures to small-scale datasets. Recent studies propose multidimensional solutions: (1) pre-training and fine-tuning paradigm: leveraging BERT/GPT-like foundational models pre-trained on large-scale generic datasets, followed by target-domain fine-tuning to enhance generalization; (2) data augmentation techniques: expanding training data distributions through spatial operations like random cropping and rotation transformations; (3) regularization strategies: integrating dropout layers and weight decay parameters to control model complexity; (4) lightweight architectures and knowledge distillation: combining streamlined structures with model distillation techniques to reduce overfitting risks while maintaining computational efficiency; and (5) transfer learning: utilizing pre-trained parameters from related tasks for model initialization, followed by domain-adaptive fine-tuning for knowledge transfer. These methods collectively balance model capacity with data scale through synergistic interactions.

With the widespread adoption of encryption technologies, evolution of attack techniques, and strengthening of privacy regulations, traditional traffic-classification methods have gradually become obsolete. Early machine-learning approaches were constrained by data-quality limitations, prompting the academic community to shift toward deep learning. While end-to-end architectures significantly enhanced classification performance, GNN encountered bottlenecks due to insufficient dynamic feature representation and high computational complexity. The introduction of Transformer architectures has brought breakthroughs through sequence modeling and self-attention mechanisms, with their advantages demonstrated in three key aspects: raw traffic processing, long-range dependency capture, and dynamic feature fusion. To address overfitting in small datasets, strategies like pre-training, data augmentation, and lightweight architectural designs have emerged as effective solutions. The field currently exhibits three major trends: the transition from manual feature engineering to deep automated learning, the evolution from static analysis to spatiotemporal dynamic modeling, and the systematic progression toward intelligent systems with enhanced robustness.

3. Methodology

To mitigate the shortcomings of traditional network traffic classification models—particularly their excessive dependence on port numbers and protocols, user privacy concerns, challenges in accurately classifying encrypted traffic, and their limited adaptability to emerging traffic types—this article introduces an efficient and high-precision encrypted network traffic classification model termed TransECA-Net. The model comprises two principal components: the first component is the CNN module integrated with the ECANet framework, while the second component is the Transformer module. (As shown in Figure 1).

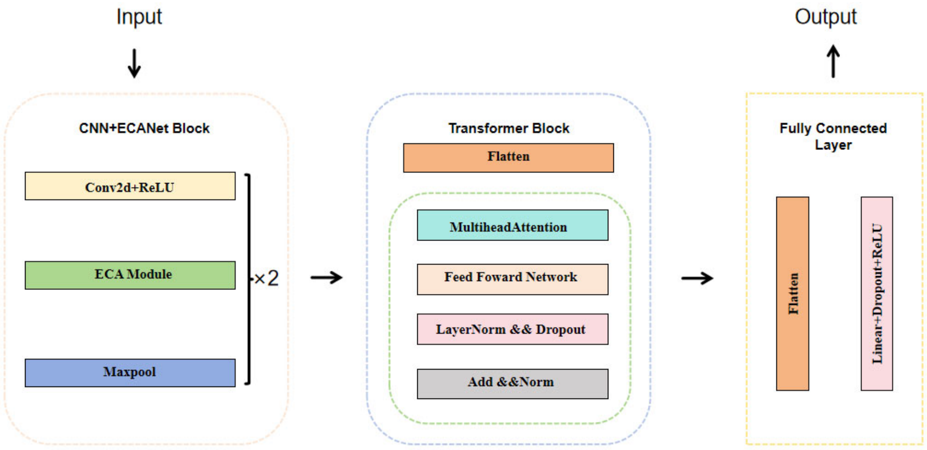

Figure 1. TransECA-Net structural flowchart.

The operational flow of the TransECA-Net model entails the sequential processing of input data through several components to achieve enhanced classification of encrypted network traffic. The process begins with feature extraction via a convolutional neural network layer. This is followed by channel attention adjustment through the ECANet module, which focuses on optimizing feature representation. The extracted features are then directed to the Transformer module, where dependencies between sequences are further captured. Finally, a fully connected layer is utilized to execute the classification task.

This model seamlessly integrates CNN, ECANet, and Transformer, capitalizing on the strengths of each component: the robust feature extraction capabilities of CNN, the advantages of the Transformer in managing sequential data, and the enhanced focus on significant features offered by ECANet. Collectively, these elements contribute to improved classification performance. Below is a detailed explanation of each component.

3.1. CNN–ECANet

Convolutional neural networks are extensively employed models in deep learning, particularly within the domains of computer vision and natural language processing. These networks are adept at processing images, videos, and various forms of structured data. A typical CNN architecture comprises convolutional layers, pooling layers, fully connected layers, and activation functions, all of which collaboratively facilitate the extraction of specific features from the data through convolution and pooling operations.

Given the characteristics of network traffic—such as packet length, source IP address, destination IP address, source port number, destination port number, and transport layer protocol type—all of which constitute sequential data, convolutional neural networks are particularly well suited for processing such information. Their exceptional scalability enables CNN to effectively manage time-series data, signal data, and various forms of sequential data. Consequently, this paper adopts the CNN structure as the foundational model for the proposed TransECA-Net framework.

Subsequently, we incorporate ECANet into the CNN architecture. ECANet serves as an efficient channel attention mechanism specifically designed to enhance the performance of convolutional neural networks while maintaining minimal computational overhead. By learning and weighting the significance of different channels, ECANet optimizes feature extraction. In the context of encrypted traffic analysis, it is vital to discern which features (i.e., channels) provide the richest information due to the inherent complexity and diversity of the data. ECANet autonomously identifies and prioritizes the more informative channels, thereby enhancing classification accuracy.

In terms of adaptability, patterns within encrypted traffic are likely to evolve continuously, influenced by variations in encryption algorithms and protocols. ECANet equips CNNs with the capacity to adjust dynamically to these changes by modulating attention toward different channels in response to fluctuations in traffic characteristics. Below is a detailed description of the ECANet structure:

Assume that X is the size of the input feature map, and its shape is N × C × H × W (where N is the batch size, C is the number of channels, H is the height, and W is the width). For each channel C of the input characteristic diagram X, calculate the global average value of the channel. Equation (1) describes the process of calculating the global average value of the channel,

$$z_c = AvgPool(X_c) = \frac{1}{H \times \mathrm{W}} \sum\nolimits_{i=1}^{H} \sum\nolimits_{j=1}^{W} \mathrm{X}_c^{(i,j)} \tag{1}$$

where is the average output of channel C, is the eigenvalue at channel C position (i, j), and H and W are the height and width of the characteristic graph, respectively.

Next, a 1D convolution is used to capture dependencies between channels and generate channel attention weights. Equation (2) illustrates the process of generating channel attention weights.

$$\mathrm{s} = \mathrm{Con}\nu(z) = \nu^{*}\sigma(W^{*}z) \tag{2}$$

Here, z is the channel vector obtained after global average pooling, W is the convolution weights, σ is the ReLU activation function, v is the output channel weight of the convolution, and the symbol * denotes the convolution operation.

The output of the convolution operation is processed through a Sigmoid activation function to normalize the channel weights to the range of 0 to 1. The Sigmoid function ensures that the output weights can be used as a normalization factor. Equation (3) shows this operation.

$$\beta = \sigma(s) = \frac{1}{1 + e^{-s}} \tag{3}$$

Here, β is the normalized channel weights, and s is the output of the convolution layer.

Finally, each channel of the input feature map is multiplied by its corresponding attention weight to enhance or suppress the features of specific channels, as shown in the following Equation (4):

$$Y_c = \beta_c \cdot X_c \tag{4}$$

where Y_c is the weighted output channel, β_c is the channel attention weight, and X_c is the original input channel. The flowchart of the ECANet structure is shown in Figure 2.

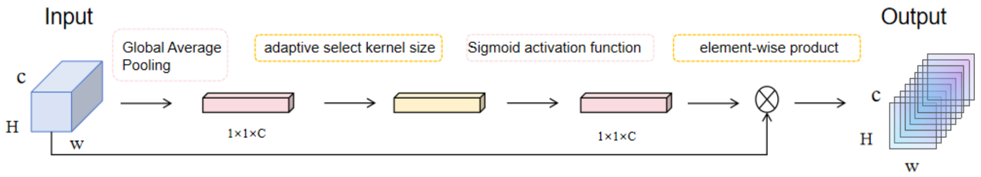

Figure 2. ECANet structural flowchart.

In the TransECA-Net model, the operational flow of the CNN layer encompasses several systematic steps:

1. 2D convolutional layer: Initially, a 2D convolutional layer processes the input feature map, configured with 1 input channel and 32 output channels. The layer uses a kernel size of (1, 25), a stride of 1, and "same" padding to ensure that the output dimensions are consistent with those of the input.
2. Activation function layer: Following the convolutional layer, an activation function layer applies the ReLU function. This introduction of non-linearity is crucial for enhancing the model's capacity to learn complex patterns.
3. ECAModule: Next, an ECAModule is implemented. This effective channel attention module adaptively modifies the weights of various channels through the application of 1D convolution followed by a Sigmoid activation function. This mechanism enables the model to focus on the most important features in the data.
4. Max pooling layer: Subsequently, a max pooling layer is employed to reduce the spatial dimensions of the feature maps. This layer is configured with a kernel size of (1, 3) and a stride of 3, leading to a more compact representation of the features while retaining the essential information.
5. Repetition: The previously mentioned configurations are repeated for a second convolutional layer, where the number of output channels is increased to 64. This enhancement allows the model to capture more intricate features and patterns in the input data.

Through this structured approach, the CNN layer is able to effectively extract and refine pertinent features from the input, contributing to the overall robustness of the TransECA-Net model.

In the domain of classifying encrypted network traffic, the incorporation of ECANet into the CNN architecture yields several distinct advantages. Unlike the straightforward one-dimensional convolutional neural network structure utilized by Wang [20], this paper enhances the CNN framework by integrating the ECANet module. This attention mechanism effectively boosts the network's performance while maintaining minimal computational overhead, particularly by emphasizing the significance of inter-channel relationships, thereby improving feature extraction and learning efficiency. In comparison to the SENet employed in Hu et al.'s research [23], ECANet presents a more efficient alternative that retains the beneficial aspects of channel attention while achieving superior computational efficiency and adaptability. This efficiency makes ECANet an appealing choice for addressing large-scale or complex image-recognition tasks, particularly in contexts where both efficiency and performance are critical considerations. Overall, the integration of ECANet within the CNN framework significantly enhances the model's ability to classify encrypted network traffic effectively.

3.2. Transformer

In the architectural design of TransECA-Net, the Transformer [25] module achieves advanced semantic modeling through multi-stage feature interaction. During the model design phase, we conducted systematic evaluations of Transformer variants based on the characteristics of encrypted traffic classification tasks. Although Perceiver reduces computational complexity through cross-attention mechanisms, its global feature interaction capability is constrained by latent-space mapping processes, which may compromise the fine-grained pattern capture of byte-level traffic sequences. While MobileViT excels in mobile vision tasks, its inherent spatial inductive bias introduces architectural mismatches with the temporal nature of network traffic. By contrast, the multi-head self-attention mechanism of the standard Transformer establishes full-connectivity dependencies between byte features at arbitrary positions, which is crucial for identifying discriminative yet dispersed protocol fingerprints in encrypted traffic.

The technical implementation involves the following workflow:

First, the H × W × C 3D feature maps extracted from the CNN backbone network (where H and W denote spatial dimensions and C represents channel count) undergo spatial dimension flattening to transform into N × (H × W) sequence representations (N being batch size), adapting to Transformer's sequential processing paradigm. This sequence is then fed into a multi-head self-attention mechanism with four parallel attention heads. Each sub-head employs learnable linear projections to map input features into 64-dimensional query (Q), key (K), and value (V) vector spaces. Global dependency matrices across spatial positions are established through scaled dot-product attention computation. The outputs from all attention heads are concatenated along the channel dimension and reconstructed via a learnable linear transformation matrix (4 heads × 64 dimensions) to form enhanced feature representations with global contextual information.

The attention output first undergoes residual addition with input features, followed by layer normalization. The normalized features are processed through a feed-forward network (FFN) comprising two fully connected layers, with ReLU activation for non-linear transformation in the intermediate layer. Both core components (MSA and FFN) integrate dropout regularization applied before residual connections to enhance model generalization.

This cascaded architecture design leverages multi-head attention to capture long-range spatial dependencies while coordinating with local FFN refinement. The residual learning strategy effectively mitigates gradient vanishing issues. Through these mechanisms, the Transformer module achieves global modeling and representation enhancement of input features, providing high-quality feature representations for subsequent classification tasks. Unlike previous models that relied on RNNs and LSTMs, the Transformer exclusively leverages the attention mechanism to handle temporal dependencies in data, without using any recurrent layers. In contrast to the CNN + LSTM network traffic classification method employed by Sarhangian [26] and Hu [23], replacing the LSTM with a Transformer module offers significant advantages, particularly in handling large-scale and high-dimensional time-series data. Although LSTMs are designed to manage long-range dependencies, they often struggle to retain information from earlier time steps when processing very long sequences. By contrast, the self-attention mechanism of the Transformer allows it to effectively handle long-range dependencies, as each element in the sequence can directly interact with all other elements without needing to propagate information iteratively through multiple time steps.

In terms of parallel computing capabilities, LSTMs are constrained by their recursive nature, which limits parallelization. However, the Transformer, which does not rely on the state of the previous time step, can process the entire input sequence simultaneously. Regarding feature extraction, while LSTMs can extract temporal features, they may require additional mechanisms (such as attention) to highlight important features. By contrast, the Transformer's self-attention mechanism inherently identifies and emphasizes key features within the sequence. This is especially crucial in encrypted traffic classification, where the features are not always immediately apparent. As for model complexity, while the Transformer is more complex than the LSTM—due to its multiple layers of self-attention and feed-forward networks—it is more efficient in processing complex data. (See Table 1 for details).

Table 1. Comparison of Transformer and LSTM performance.

Comparison Aspect	Transformer	LSTM
Processing capacity	Strong	Weak
Dependency capture	Strong	Weak
Computational efficiency	High	Mid
Feature learning	Automatically learn the features from the data	Dependence on manual feature extraction
Model complexity	Complex	Simple

Finally, in the TransECA-Net model, the operation of the fully connected layers proceeds as follows: The output of the Transformer is flattened again, and three linear layers (fully connected layers) are used to make decisions for the classification task. This process includes dropout layers to reduce overfitting and ReLU activations to introduce non-linearity. The final output layer is responsible for predicting the class labels, with its dimensionality matching the number of target categories.

4. Experimental Result

In the experimental design and performance evaluation section, this study employs a three-phase progressive validation framework: first, establishing a benchmark experimental system based on standardized datasets; second, validating the effectiveness of the model components through ablation experiments; and finally, conducting horizontal comparative experiments to demonstrate methodological superiority. To ensure experimental reproducibility, this paper strictly adheres to international machine-learning research standards and implements interpretable experimental design principles: (1) selecting authoritative public datasets to avoid data bias; (2) ensuring feature engineering consistency through modular preprocessing pipelines; (3) balancing model capacity with computational efficiency via adaptive hyperparameter configurations; and (4) conducting comprehensive performance analysis using confusion matrix analysis, metric analysis, computational resource analysis, and ablation experiments. Notably, the experimental environment configuration fully considers industrial application requirements, implementing production-environment transferability verification on consumer-grade GPU hardware. This systematic experimental design not only guarantees the statistical significance of the research conclusions but also provides engineering practice references for deploying real-world network traffic analysis systems.

4.1. Dataset

This study utilizes the ISCX VPN-nonVPN dataset, which was created in 2016 by the Canadian Institute for Cybersecurity (CIC) at the University of New Brunswick. The key characteristics of this dataset are as follows:

1. Traffic types: The dataset includes various types of VPN-encrypted traffic as well as unencrypted regular internet traffic. It covers common network services and applications such as web browsing, email, file transfer, and VoIP (Table 2).
2. Realistic scenario simulation: All traffic data were generated in a controlled environment to simulate real-world network communication scenarios.

3. Detailed annotations: The dataset is thoroughly labeled, indicating whether the traffic was transmitted via a VPN and specifying which VPN protocol was used.

Table 2. Labeled ISCX VPN-nonVPN dataset.

Traffic	VPN-Traffic
Browsing	VPN-Browsing
Email	VPN-Email
Chat	VPN-Chat
Streaming	VPN-Streaming
File transfer	VPN-File transfer
VoIP	VPN-VoIP
TraP2P	VPN-TraP2P

Based on the ISCX VPN-nonVPN dataset, the preprocessing method used by Wang in 2017 [20] was implemented with the USTC-TL2016 tool. The processing involves four steps:

1. Traffic segmentation: Split raw PCAP traffic into session and flow units. To ensure uniform input dimensions for CNN, truncate each session/flow to retain only 784 bytes.
2. Traffic cleaning: Remove retransmitted packets, corrupted packets, and non-encrypted traffic.
3. Image generation: Convert traffic bytes into a one-dimensional numerical sequence, where each byte corresponds to an integer value between 0 and 255, forming time-series data similar to textual sequences.
4. IDX format conversion: Convert byte sequences into IDX3 files (input data) and IDX1 files (labels) to align with frameworks like PyTorch 3.9/TensorFlow (https://tensorflow.google.cn/?hl=zh-cn, accessed on 20 September 2023). Label PCAP files according to ISCX dataset descriptions and remove ambiguous categories (e.g., conflicting portions between "Browser" and "Streaming"), ultimately retaining 12 distinct classes.

The selection of this dataset is based on the following rationale. As an internationally recognized benchmark dataset widely adopted in the field of cybersecurity (references [15,22,23]), its evaluation results possess comparative value across studies. The 12 service scenarios constructed by the dataset comprehensively cover both VPN-encrypted traffic (six categories) and non-VPN regular traffic (six categories), accurately characterizing multi-dimensional network behavior features, including browsing, email, and P2P transmission (as shown in Table 2). The controlled environment generation mechanism, combined with manual cleaning processes, achieves the high-precision labeling of traffic types and encryption states by eliminating retransmitted or corrupted packets. In the preprocessing stage, the truncation of 784 bytes per flow allows the conversion of raw data into a standardized 28 × 28 image format, fully aligning with the input dimensional requirements of deep learning models such as CNNs. It is particularly noteworthy that the unique protocol encapsulation mechanism of VPN tunnels significantly increases the complexity of traffic pattern decoupling, which effectively validates the model's ability to parse encryption-obfuscated features.

4.2. Basic Experiment

This paper proposes an encrypted network traffic classification method based on CNN–ECANet + Transformer, using the publicly available ISCX VPN-nonVPN dataset [27] for experimental tasks. The dataset consists of raw PCAP files, which, after preprocessing, are divided into individual flows and sessions and then converted into IDX files in the form of 28 × 28 × 1 pixel images for training. During the training process, we randomly selected 1/10 of the total data as test data (i.e., the test set), while the remaining 9/10 of the data was used for training (i.e., the training set). Throughout the training phase, the model parameters were optimized through 10-fold cross-validation.

To evaluate the proposed method, the experiments were conducted on a device equipped with an AMD Ryzen 5 5600H processor with Radeon Graphics (3.30 GHz), 16 GB of RAM, and an NVIDIA GeForce 3050Ti GPU, running Microsoft Windows 11 (64-bit). (Detailed device specifications and the operating environment are shown in Table 3.) The experiments utilized the open-source machine learning library PyTorch 3.9. As a widely used deep learning framework, PyTorch offers a rich collection of pretrained models and components, which can accelerate the development process of encrypted network traffic classification models. Additionally, it provides a variety of debugging tools and visualization libraries that help users better understand and analyze model behavior, offering advantages in debugging and visualization.

Table 3. Computer configuration and running environment.

Configuration and Environment	
CPU	AMD Ryzen 5 5600H with Radeon Graphics 3.30 GHz
RAM	16.0 GB
GPU	NVIDIA GeForce RTX3050Ti
Operating system	Windows 11 64-bit
Integrated development environment	PyCharm
Deep learning framework	PyTorch

Finally, PyTorch natively supports GPU-accelerated computation, fully leveraging the parallel computing capabilities of the GPU to speed up model training and inference. In encrypted network traffic classification tasks, where large datasets and complex models often need to be processed, GPU acceleration can significantly enhance computational efficiency.

During the model training phase, this study employs cross-entropy loss, the AdamW optimizer [28], and data augmentation techniques [29]. For each training sample, the model generates a predicted probability distribution, representing the likelihood that the sample belongs to each category. The true label indicates the actual class of the sample. Cross-entropy loss is used to measure the difference between the model's predictions and the true labels, and the model's parameters are optimized by minimizing this difference.

The AdamW optimizer, introduced by Ilya Loshchilov and Frank Hutter in their 2017 paper "Decoupled Weight Decay Regularization" [30], is an improvement over the traditional Adam optimizer. In this experiment, we opted for the more efficient AdamW optimizer instead of the conventional SGD or Adam optimizers. AdamW combines the benefits of both RMSprop and Momentum by calculating the exponentially moving average of gradients and adjusting the learning rate for each parameter based on unbiased estimates of these averages. It also resolves the issue of weight decay being coupled with gradient updates, which is a known limitation of the Adam optimizer. Numerous deep learning tasks have shown that AdamW outperforms traditional SGD and Adam optimizers, providing more stable training and better generalization, particularly in tasks involving weight decay.

Data augmentation plays a key role in the training process by expanding the dataset. By applying various transformations (such as rotation, scaling, cropping, etc.) to the training data, the technique effectively increases the size of the dataset, which enhances the model's robustness and performance. In this study, network traffic data is transformed into a two-dimensional matrix (represented as 28 × 28 image-like data). The following data augmentation strategies were implemented: minor rotations or horizontal flips simulated variations in packet sequencing or protocol implementations; scaling operations adjusted matrix dimensions with zero-padding or truncation; cropping techniques emulated traffic length variations; and random matrix truncation mimicked incomplete traffic capture scenarios. These approaches partially addressed dataset imbalance through enhanced data diversity.

Hyperparameter selection is a central component of deep learning model design, requiring decisions to be grounded in a balanced integration of empirical heuristics, task-specific characteristics, hardware limitations, and experimental validation. The selection of hyperparameters in this study was based on a combination of literature review and theoretical analysis. By referencing multi-domain research findings, we ensured that the chosen parameters exhibited both theoretical validity and practical feasibility.

1. Learning rate configuration: As demonstrated in [31], the learning rate plays a decisive role in model convergence speed and performance. An initial learning rate of 0.001 was adopted, a value validated by multiple studies as an ideal starting point for deep learning tasks. Additionally, an adaptive learning rate scheduling strategy was implemented to enhance training efficiency through dynamic adjustment mechanisms.
2. Batch configuration: Considering GPU parallel-processing capabilities and training stability, the batch size was set to 128. As discussed in [32], this configuration achieves a balance between training speed and model performance while effectively mitigating overfitting risks.
3. Training epochs: By monitoring training loss and validation accuracy, the final training epoch count was determined to be 40. Research [33] highlights that the excessive prolongation of training epochs increases overfitting risks; thus, the optimal upper limit was established based on convergence curve analysis.
4. Dropout mechanism: Following the sensitivity analysis of network layers in [34], a layered dropout strategy was implemented: 0.5 for fully connected layers and 0.3 for the output layer. This approach employs differentiated regularization to suppress overfitting while enhancing model generalization capabilities.

To evaluate the performance of the model's training results and facilitate comparison with other encrypted network traffic classification methods, this paper uses three metrics—accuracy, precision, and recall—to assess the model's performance.

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

$$\text{Precision} = \frac{TP}{TP + FP}$$

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

In this context, TP (true positive) represents the number of instances correctly classified as the current network category. TN (true negative) denotes the number of instances accurately classified as not belonging to the current network category. FP (false positive) refers to the number of other network categories mistakenly classified as the current network category, while FN (false negative) indicates the number of instances from the current network category that were incorrectly identified as belonging to other network traffic categories.

The experimental results are shown in Figure 3. The left graph displays the training loss value for each epoch, with the x-axis representing the number of epochs and the y-axis representing the loss value. As shown in the figure, the loss value stabilizes around 0.035 by the 10th epoch. The right graph illustrates the training accuracy and test accuracy for each epoch, where the x-axis represents the number of epochs and the y-axis indicates the accuracy. From the graph, it is evident that the curve becomes smooth after the 15th epoch, with the accuracy approaching 99%. Additionally, after extensive experimentation, it was found that the model generally achieves optimal performance after 40 training iterations. Although the complexity of the model increases after incorporating ECANet and Transformer, the powerful parallel-processing capability of the Transformer and the optimized feature-extraction ability of ECANet enable the TransECA-Net model to converge faster in encrypted network traffic classification tasks, reducing computational resource consumption with fewer training iterations.

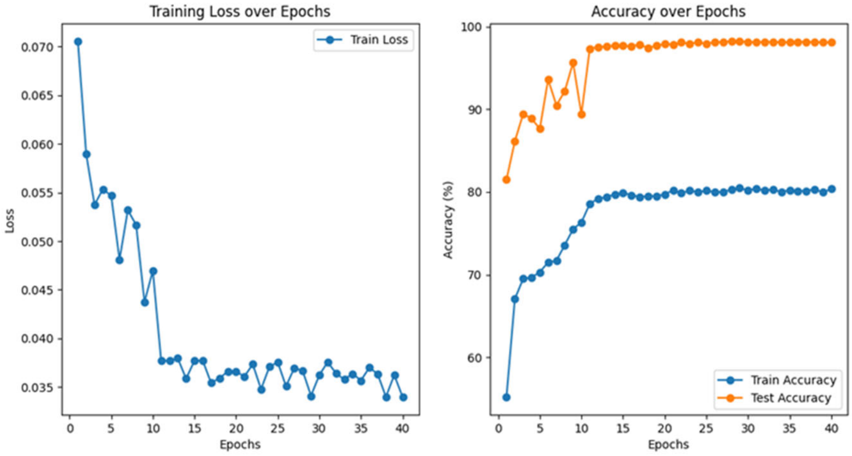

Figure 3. TransECA-Net model performance.

The calculated overall average accuracy rate of this model is 98.25%. Table 4 displays the accuracy and recall rates for 12 different types of network traffic. The data demonstrate that the TransECA-Net model achieved an average accuracy of 97.65% for 12-class traffic classification in the ISCX dataset, with an average precision of 98.15% and an average recall rate of 97.91%.

Table 4. The classification results on 12 types of traffic.

Category	Accuracy	Precision	Recall	Category	Accuracy	Precision	Recall
Chat	98.17%	98.67%	99.17%	VPN-Chat	99.83%	99.50%	99.83%
Email	99.00%	99.17%	99.00%	VPN-Email	98.33%	100%	98.33%
File	93.56%	89.56%	94.33%	VPN-File	98.49%	98.98%	99.49%
P2P	99.46%	100%	99.46%	VPN-P2P	96.74%	98.89%	96.74%
Streaming	99.75%	99.50%	99.75%	VPN-Streaming	99.19%	99.19%	100%
VoIP	92.20%	94.37%	89.33%	VPN-VoIP	99.50%	100%	99.50%

Figure 4 displays the normalized confusion matrix of TransECA-Net on the 12-class encrypted traffic from the ISCX dataset. The vertical axis represents the true classes, while the horizontal axis indicates the predicted classes. The diagonal elements show the correct classification rates for each category, whereas the off-diagonal elements reflect misclassification patterns. Analysis of Table 4 and Figure 4 reveals a bidirectional confusion between the Chat and VPN-Chat classes: 0.80% of the Chat samples were misclassified as VPN-Chat, while 0.50% of the VPN-Chat samples were, conversely, misclassified as Chat. This phenomenon may stem from similarities in message interaction frequency and payload distribution between non-encrypted communications and encrypted channels. The File class exhibits notable misclassification characteristics, with 6.50% of samples misclassified as P2P and 3.35% as Streaming, indicating feature-space overlap between the bursty traffic patterns of large file transfers, P2P chunked transmission, and streaming buffering mechanisms. The VoIP class demonstrates dual confusion patterns: 3.10% of samples were misclassified as P2P, potentially due to similar heartbeat mechanisms, while 7.60% were misclassified as VPN-VoIP, suggesting insufficient discriminative power in packet timing features for encrypted voice streams.

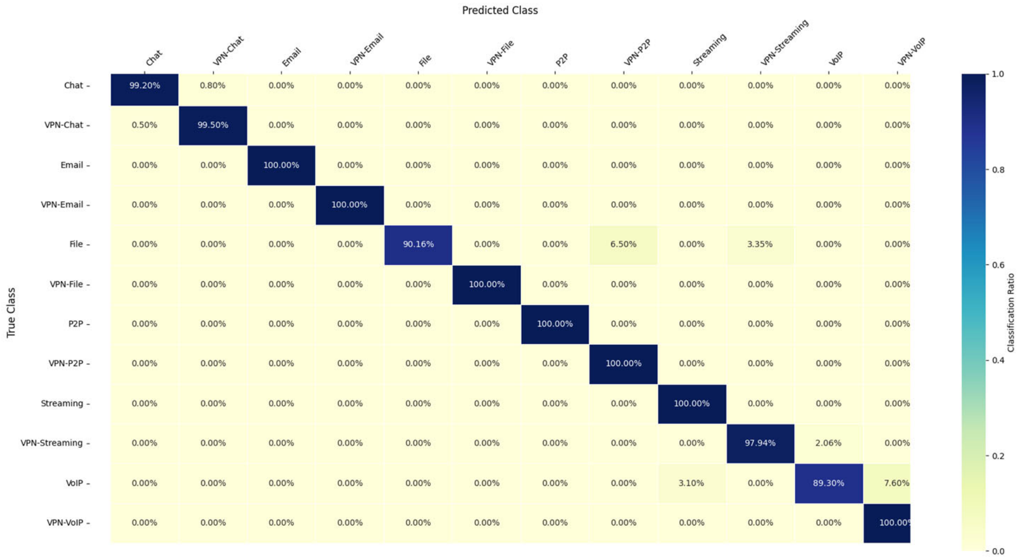

Figure 4. Normalized confusion matrix of 12 types of encrypted traffic.

Regarding performance metrics, the File class achieved the lowest classification accuracy of 93.56%, with a precision of 89.56% and recall of 94.33%. The significant disparity between precision and recall indicates a higher false-positive rate, consistent with its role as a primary source of misclassification in the confusion matrix. The VoIP class attained 92.20% accuracy, with precision and recall rates of 94.37% and 89.33%, respectively, revealing a trade-off between false positives and false negatives. This likely relates to the bidirectional encryption characteristics of voice traffic and its similarity to VPN-VoIP protocols.

Finally, to validate the generalization capability of the TransECA-Net model, we conducted tests using ten normal traffic types from the USTC-TFC2016 dataset. The experimental results demonstrate that the TransECA-Net model maintains robust classification performance in cross-dataset scenarios. Although the recognition accuracy for the Weibo and Outlook types showed a slight decrease of 0.6–2% compared to the average accuracy achieved on the ISCX dataset, the model attained exceptional classification accuracy exceeding 99% for other traffic types, particularly in the BitTorrent, FTP, and Skype categories. These findings effectively verify the strong generalization capability of TransECA-Net, ensuring its reliable classification performance across diverse network environments.

4.3. Ablation Study and Component Analysis

To verify the independent contributions of each module of TransECA-Net, this section designs ablation experiments to compare the following variants:

1. Baseline CNN: Retains only the original CNN structure (without ECANet or Transformer).
2. CNN + ECANet: Integrates the ECANet module based on CNN.
3. Transformer-only: Uses only the standard Transformer architecture (without CNN or ECANet).
4. TransECA-Net (full model): A hybrid architecture combining CNN–ECANet and Transformer.

The experiments use the same ISCX dataset and hyperparameter configuration (batch size = 128, epoch = 40).

The results in Table 5 demonstrate that after introducing the ECANet module, the classification accuracy of the baseline CNN model increased by 3.00%, and the recall rate improved by 2.84%. This method effectively validates the feasibility of dynamically adjusting feature channel weights while adding only 0.08M parameters and 0.03GFLOPs of computational cost. On the other hand, the pure Transformer architecture significantly increased the parameters by 45.6% and computation by 244%, achieving 95.30% accuracy and 94.07% recall under these conditions. While it demonstrates notable global feature modeling capabilities, its computational efficiency remains limited. Finally, the complete TransECA-Net model, through the joint optimization of CNN–ECANet and Transformer, achieved state-of-the-art performance in the benchmark (98.64% accuracy and 98.13% recall) with 6.89 M parameters and 13.45 GFLOPs computational cost. Compared to single architectures, this model shows a 4.75% accuracy improvement over the baseline CNN and a 3.34% gain compared to the pure Transformer, fully validating the architectural advantages of fusing local detailed features with global contextual information.

Table 5. Comparison of experimental results of component ablation.

Model	Precision	Recall	Number of Parameters	FLOPs
Baseline CNN	93.89%	93.37%	1.82 M	0.36 G
CNN + ECANet	96.88%	96.21%	1.90 M	0.39 G
Transformer-only	95.30%	94.07%	2.65 M	1.24 G
TransECA-Net (full model)	98.64%	98.13%	6.89 M	13.45 G

4.4. Comparative Experiment

To further validate the performance of the TransECA-Net model, this paper compares it with several existing network traffic classification methods on the ISCX dataset. Wang et al. [20] proposed an end-to-end network traffic classification approach based on 1D-CNN. Lotfollahi et al. [21] introduced a CNN + SAE-based method, where the use of SAE improved the performance of traffic classification tasks. Hu [23] proposed a network traffic classification method combining LSTM and CNN, integrating LSTM into the CNN architecture and adding the SENet attention mechanism, which significantly enhanced the classification performance for VPN traffic. Zhang et al. [24] proposed a temporal fusion encoder using graph neural networks (TFE-GNN) model for fine-grained encrypted traffic classification, which demonstrates superior performance in classification accuracy through graph-structured learning of packet interaction patterns.

Under the same conditions of using the ISCX international public dataset, the TransECA Net model was compared with these existing literature classification methods, and the comparison results are shown in Table 6.

Table 6. The comparison between TransECA-Net and other methods.

Method	VPN			Non-VPN		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
1D-CNN [20]	94.48%	94.48%	93.57%	83.94%	83.89%	84.00%
SAE + 1D-CNN [21]	93.00%	97.80%	96.30%	87.62%	86.70%	88.80%
CNN + LSTM [23]	93.69%	94.57%	94.33%	90.91%	91.69%	92.00%
TFE-GNN [24]	95.91%	95.26%	95.36%	90.40%	93.16%	91.90%
OUR	98.90%	98.20%	97.93%	98.39%	97.43%	98.02%

As shown in Table 5, the 1D-CNN and its improved version, SAE + 1D-CNN, demonstrate excellent performance in VPN traffic classification, with accuracy, precision, and recall all exceeding 90%. However, on the non-VPN dataset, both methods experience a significant drop in performance, with the 1D-CNN achieving an accuracy of only 83.94% and SAE + 1D-CNN reaching just 87.62%. Although the CNN–LSTM hybrid model reported in the literature performs exceptionally well in both VPN and non-VPN traffic classification (with VPN accuracy exceeding 97% and non-VPN accuracy above 95%), in our local experimental setup, the optimal classification accuracy for this model was only 93.69% for VPN traffic and 90.91% for non-VPN traffic. By contrast, the TFE-GNN method, which integrates byte-level traffic graphs with graph neural networks, demonstrates significant advantages in fine-grained feature extraction, complex structure modeling, and multi-dimensional feature fusion, achieving a VPN traffic classification accuracy of 95.91%. Notably, the first four methods generally exhibit inadequate performance in non-VPN traffic classification, with average metric declines ranging from 5% to 8%. The proposed TransECA-Net model excels in both traffic classification tasks, with average accuracy, precision, and recall exceeding 98%. The experimental results indicate that this model achieves significant performance improvements in encrypted traffic classification, showcasing clear advantages over existing methods.

To facilitate a more comprehensive comparison of the temporal efficiency associated with various models during the training process, this study meticulously assesses both the training duration and convergence period for each model. The training duration is defined as the time necessary for the model to be trained on the training dataset, and it is influenced by several factors, including the number of iterations, model complexity, and the specific hardware utilized in the experiments. Conversely, convergence time pertains to the duration required for the model to attain a state of relatively stable performance, which is likewise affected by variables such as model complexity.

As illustrated in Table 7, the time expenditure of diverse models operating on the ISCX dataset under consistent hardware conditions indicates that the proposed methodology significantly mitigates convergence time. Specifically, it necessitates only 51.16% of the duration required by the 1D-CNN, 53.78% of that for SAE + 1D-CNN, 62.56% of that for the CNN + LSTM model, and 46.87% of that for the TFE-GNN model. Consequently, it can be inferred that the TransECA-Net architecture accomplishes the classification task with reduced temporal expenditure during the training phase, thereby achieving enhanced classification efficiency in comparison to other methodologies.

Table 7. Comparison of operational efficiency of different models.

Model	Training Time (s)	Convergence Time (s)
1D-CNN [20]	2104.89	720.54
SAE + 1D-CNN [21]	2215.78	685.36
CNN + LSTM [23]	1967.76	589.17
TFE-GNN [24]	1768.21	780.05
OUR	1001.20	368.62

To comprehensively evaluate the performance of TransECA-Net, we conducted comparative experiments with baseline models including 1D-CNN, SAE + 1D-CNN, CNN–LSTM, and TFE-GNN on encrypted traffic datasets. To ensure the rigor of the performance comparison, each model underwent 30 independent repeated experiments, with the training and test sets partitioned at an 80%:20% ratio using random seeds for each trial. Paired t-tests were performed between TransECA-Net and all baseline models on the same test set:

Paired comparisons of accuracy rates between TransECA-Net and each baseline model on the identical test set were performed, with p-values calculated through statistical analysis.

Null hypothesis (H_0): No significant difference in performance between the two models (μTransECA-Net = μBaseline).

Alternative hypothesis (H_1): TransECA-Net performs significantly better (μTransECA-Net > μBaseline).

Significance level: $\alpha = 0.05$ (one-tailed test).

Compute Cohen's d to quantify the magnitude of differences:

$$d = \frac{\mu TransECAnet - \mu Baseline}{\sigma pooled}$$

where

$$\sigma pooled = \sqrt{\frac{(n_1 - 1)\sigma_1^2 + (n_2 - 1)\sigma_2^2}{n_1 + n_2 - 2}}$$

The results demonstrate statistically significant improvements ($p < 0.001$, single-tailed) for TransECA-Net over 1D-CNN (Cohen's d = 2.1), SAE + 1D-CNN (d = 1.8), CNN–LSTM (d = 1.6), and TFE-GNN (d = 1.2). The large effect sizes (d > 0.8) further confirm that the performance gaps are not only statistically significant but also practically meaningful.

In deep learning tasks, it is generally desirable for the loss value to gradually decrease during training, as the loss is a metric that measures the difference between the model's predictions and the true labels. As the model learns better feature representations and adjusts its parameters, the loss should steadily decline, indicating that the model's predictions on the training data are improving. Figures 5–8 respectively show the loss curves during training for the three aforementioned models and the TransECA-Net model used in this study. The x-axis represents each training batch, while the y-axis indicates the loss value. It can be observed that the loss values for the 1D-CNN and SAE + CNN models remain relatively high in the later stages of training and exhibit significant fluctuations. Although the CNN + LSTM model achieves a low loss after convergence, it still experiences large fluctuations toward the later stages of training, even at batch 10,000, with a maximum difference of up to 0.9. By contrast, the TransECA-Net model demonstrates a smaller and more stable loss value after convergence compared to the 1D-CNN and SAE + CNN models. The reduced fluctuations and more stable curve suggest that the TransECA-Net model is more stable and has better generalization performance.

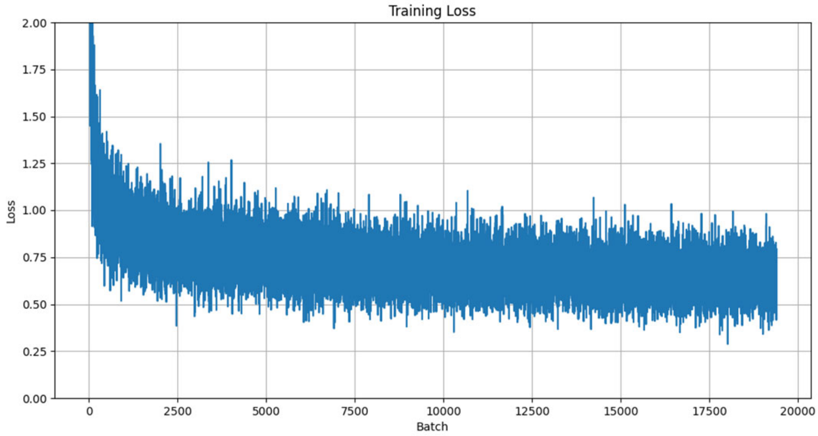

Figure 5. Linear graph of training loss values for 1D-CNN model.

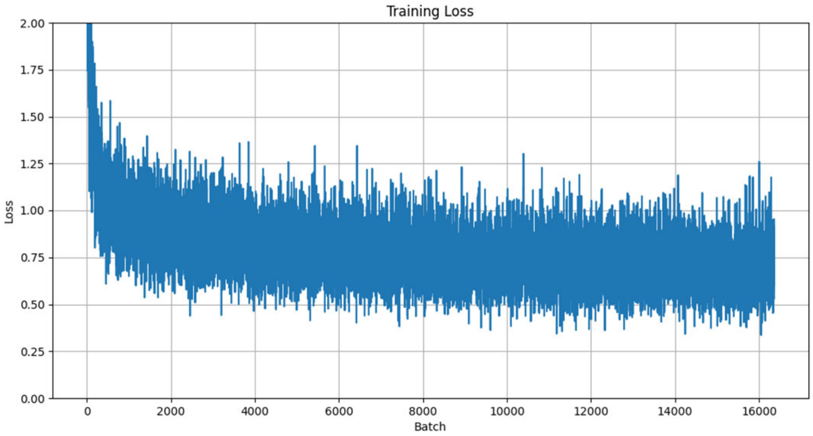

Figure 6. Linear graph of training loss values for SAE + 1D-CNN model.

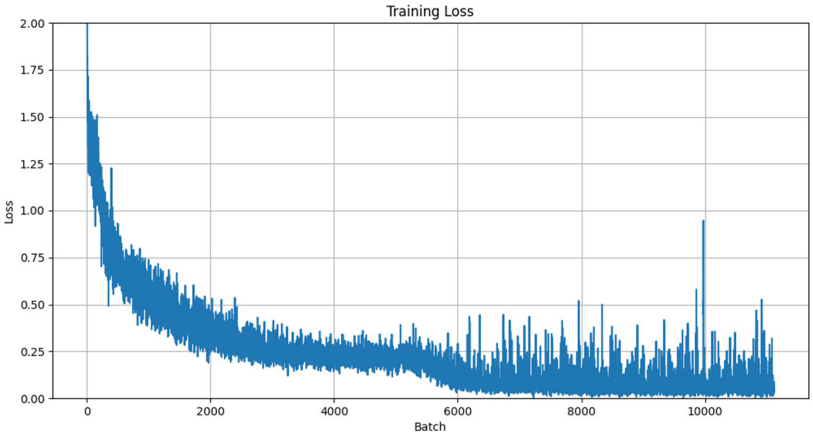

Figure 7. Linear graph of training loss values for CNN + LSTM model.

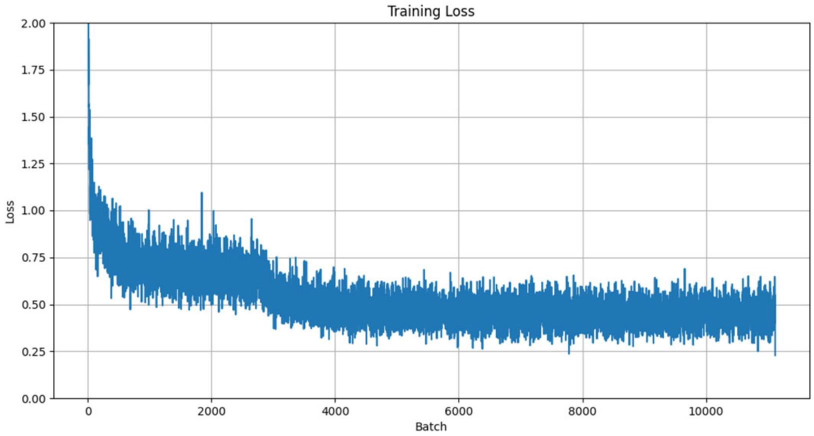

Figure 8. Linear graph of training loss values for TransECA-Net model.

5. Discussion

The TransECA-Net model proposed in this study demonstrates high performance in encrypted traffic classification tasks, but its practical deployment faces several challenges. Currently, the model has limitations in adapting to zero-day traffic patterns, struggling to effectively identify traffic types not encountered during training, which may compromise classification stability in dynamic network environments. Additionally, the model's sensitivity to adversarial attacks cannot be overlooked—attackers can disrupt traffic features through carefully crafted perturbations, leading to degraded classification performance and posing potential threats to network security reliability.

Future research should prioritize enhancing the model's dynamic awareness of unknown traffic patterns to achieve real-time detection and adaptive updates for zero-day traffic. Simultaneously, efforts should focus on improving the model's robustness at the algorithmic level by exploring adversarial training, feature perturbation defense mechanisms, or model fusion strategies to counter increasingly sophisticated adversarial attacks. Future work could also investigate integrating multiple data sources for encrypted traffic classification to further boost accuracy and robustness.

Moreover, building a real-time network monitoring system that integrates TransECA-Net into cybersecurity infrastructure is essential for rapid response to emerging threats. Concurrently, research should evaluate the model's applicability across diverse network environments, such as IoT or 5G networks, to ensure effectiveness in varied application scenarios. Through these improvements, TransECA-Net is expected to achieve more secure and sustainable encrypted traffic analysis capabilities in open threat environments.

6. Conclusions

This study proposes TransECA-Net, a novel hybrid architecture integrating CNN, ECANet, and Transformer modules that is designed to address critical challenges in encrypted traffic classification. While numerical metrics (e.g., 98.25% average accuracy and 37.44–48.84% faster convergence than baseline models) demonstrate its superiority, the core contributions lie in its architectural innovation and practical deployment value. The model's performance advantages over existing methods (1D-CNN, CNN + LSTM, TFE-GNN) stem from synergistic component design: The ECANet module dynamically prioritizes traffic patterns through adaptive channel feature recalibration, reducing computational redundancy by 37.44–48.84% compared to traditional attention mechanisms, effectively alleviating efficiency bottlenecks in manual feature engineering for dynamic encrypted traffic. Simultaneously, the Transformer module captures global temporal dependencies through self-attention, overcoming the limited receptive field of LSTM-based methods. By integrating local feature enhancement with global sequence modeling, the model successfully deciphers the hidden characteristics of encrypted protocol fingerprints—a capability crucial for addressing modern cybersecurity challenges.

Beyond accelerated training convergence, the architecture optimizes real-time inference efficiency. The Transformer's parallelizable self-attention eliminates RNN's sequential computation constraints, significantly boosting processing efficiency for high-dimensional traffic sequences. Combined with ECANet's lightweight channel-selection strategy, this creates low-computation inference pipelines suitable for edge devices and real-time monitoring systems. Subsequent research should quantify cross-platform inference latency to validate deployment effectiveness. Notably, this work's contributions transcend quantitative metrics: (1) transitioning from static graph representation to dynamic sequence modeling eliminates the computational costs of manual graph construction, enhancing adaptability to protocol evolution; (2) persistent technical barriers in non-VPN traffic classification are overcome (achieving 98.39% accuracy), surpassing traditional methods' limitations in detecting weak protocol features and high-entropy data; and (3) a modular design enables expansion to cross-protocol recognition and zero-shot learning scenarios, laying foundations for adaptive security systems.

In summary, TransECA-Net redefines encrypted traffic analysis paradigms through balanced model efficiency, classification accuracy, and architectural scalability. Its innovations not only advance technical boundaries but also provide systematic solutions for dynamic encryption evolution and real-time security demands, offering enduring value in adaptive cybersecurity defense.

Author Contributions: Conceptualization, Z.L.; Methodology, Z.L. and Y.X.; Software, Z.L.; Validation, Y.X. and Y.L.; Formal analysis, Y.L.; Writing—original draft, Z.L. and Y.W.; Supervision, X.J.; Project administration, X.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China: No. 61572304; the Key Program of the National Natural Science Foundation of China: No. 61332019; the Shanghai Sailing Plan of "Science and Technology Innovation Action Plan": No. 21YF1415100; National Natural Science Foundation of China: No. 61272096; Fujian Provincial Natural Science Foundation Project: No. 2021J01129.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

References

1. Nascita, A.; Aceto, G.; Ciuonzo, D.; Montieri, A. A Survey on Explainable Artificial Intelligence for Internet Traffic Classification and Prediction, and Intrusion Detection. *IEEE Commun. Surv. Tutor.* **2024**. [CrossRef]
2. Gang, L.; Zhang, Z. Deep encrypted traffic detection: An anomaly detection framework for encryption traffic based on parallel automatic feature extraction. *Comput. Intell. Neurosci.* **2023**, *2023*, 3316642.
3. De Keersmaeker, F.; Cao, Y.; Kabasele, G.; Sadre, R. A survey of public IoT datasets for network security research. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 1808–1840. [CrossRef]
4. D'Alconzo, A.; Drago, I.; Morichetta, A.; Mellia, M.; Casas, P. A survey on big data for network traffic monitoring and analysis. *IEEE Trans. Netw. Serv. Manag.* **2019**, *16*, 800–813. [CrossRef]
5. Saied, M.; Shawkat, G.; Magda, M. Review of artificial intelligence for enhancing intrusion detection in the internet of things. *Eng. Appl. Artif. Intell.* **2024**, *127*, 107231. [CrossRef]
6. Papadogiannaki, E.; Sotiris, I. A survey on encrypted network traffic analysis applications, techniques, and countermeasures. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–35. [CrossRef]
7. Atadoga, A.; Farayola, O.A.; Ayinla, B.S.; Amoo, O.O.; Abrahams, T.O.; Osasona, F. A comparative review of data encryption methods in the USA and Europe. *Comput. Sci. IT Res. J.* **2024**, *5*, 447–460. [CrossRef]
8. Wang, Z.; Vrizlynn, L.L. Thing. Feature mining for encrypted malicious traffic detection with deep learning and other machine learning algorithms. *Comput. Secur.* **2023**, *128*, 103143. [CrossRef]
9. Karakus, M.; Arjan, D. Quality of service (QoS) in software defined networking (SDN): A survey. *J. Netw. Comput. Appl.* **2017**, *80*, 200–218. [CrossRef]
10. Çelebi, M.; Alper, Ö.; Uraz, Y. A comprehensive survey on deep packet inspection for advanced network traffic analysis: Issues and challenges. *Niğde Ömer Halisdemir Üniv. Mühendislik Bilim. Derg.* **2023**, *12*, 1–29. [CrossRef]
11. Dong, S. Multi class SVM algorithm with active learning for network traffic classification. *Expert Syst. Appl.* **2021**, *176*, 114885. [CrossRef]
12. Najm, I.A.; Saeed, A.H.; Ahmad, B.A.; Ahmed, S.R.; Sekhar, R.; Shah, P.; Veena, B.S. Enhanced Network Traffic Classification with Machine Learning Algorithms. In Proceedings of the Cognitive Models and Artificial Intelligence Conference, Istanbul, Turkiye, 25–26 May 2024.
13. Rezaei, S.; Xin, L. Deep learning for encrypted traffic classification: An overview. *IEEE Commun. Mag.* **2019**, *57*, 76–81. [CrossRef]
14. Wang, Q.; Wu, B.; Zhu, P.F.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
15. Yoon, S.-H.; Park, J.; Park, J.-S.; Oh, Y.-S.; Kim, M.-S. Internet Application Traffic Classification Using Fixed IP-Port. In *Management Enabling the Future Internet for Changing Business and New Computing Services, Proceedings of the 12th Asia-Pacific Network Operations and Management Symposium, APNOMS, Jeju, South Korea, 23–25 September 2009*; Springer: Berlin/Heidelberg, Germany, 2009.
16. Velan, P.; Čermák, M.; Čeleda, P.; Drašar, M. A survey of methods for encrypted traffic classification and analysis. *Int. J. Netw. Manag.* **2015**, *25*, 355–374. [CrossRef]
17. Stevanovic, M.; Pedersen, J.M. An analysis of network traffic classification for botnet detection. In Proceedings of the IEEE 2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), London, UK, 8–9 June 2015.

18. Huang, S.; Chen, K.; Liu, C.; Liang, A.; Guan, H. A statistical-feature-based approach to internet traffic classification using machine learning. In Proceedings of the IEEE 2009 International Conference on Ultra Modern Telecommunications & Workshops, St. Petersburg, Russia, 12–14 October 2009.
19. Han, Y.; Kushal, V.; Erdal, O. Robust traffic sign recognition with feature extraction and k-NN classification methods. In Proceedings of the 2015 IEEE International Conference on Electro/Information Technology (EIT), Dekalb, IL, USA, 21–23 May 2015.
20. Wang, W.; Zhu, M.; Jinlin, W.; Zeng, X.; Yang, Z. End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In Proceedings of the 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), Beijing, China, 22–24 July 2017.
21. Lotfollahi, M.; Zade, R.S.H.; Siavoshani, M.J.; Saberian, M. Deep packet: A novel approach for encrypted traffic classification using deep learning. *Soft Comput.* **2020**, *24*, 1999–2012. [CrossRef]
22. Wang, W.; Sheng, Y.; Wang, J.; Zeng, X.; Ye, X.; Huang, Y.; Zhu, M. HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. *IEEE Access* **2017**, *6*, 1792–1806. [CrossRef]
23. Hu, F.; Zhang, S.; Lin, X.; Wu, L.; Liao, N.; Song, Y. Network traffic classification model based on attention mechanism and spatiotemporal features. *EURASIP J. Inf. Secur.* **2023**, *2023*, 6. [CrossRef]
24. Zhang, H.; Yu, L.; Xiao, X.; Li, Q.; Marcaldo, F.; Luo, X.; Liu, Q. TFE-GNN: A temporal fusion encoder using graph neural networks for fine-grained encrypted traffic classification. In Proceedings of the ACM Web Conference 2023, New York, NY, USA, 30 April–4 May 2023.
25. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
26. Sarhangian, F.; Rasha, K.; Muhammad, J. Efficient traffic classification using hybrid deep learning. In Proceedings of the 2021 IEEE International Systems Conference (SysCon), Vancouver, BC, Canada, 15 April–15 May 2021.
27. Azab, A.; Khasawneh, M.; Alrabaee, S.; Choo, K.-K.R.; Sarsour, M. Network traffic classification: Techniques, datasets, and challenges. *Digit. Commun. Netw.* **2024**, *10*, 676–692. [CrossRef]
28. Loshchilov, I.; Frank, H. Fixing Weight Decay Regularization in Adam. 2018. Available online: https://openreview.net/forum?id=rk6qdGgCZ (accessed on 30 September 2024).
29. Khan, A.A.; Chaudhari, O.; Chandra, R. A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation. *Expert Syst. Appl.* **2024**, *244*, 122778. [CrossRef]
30. Xie, Z.; Xu, Z.; Zhang, J.; Sato, I.; Sugiyama, M. On the Overlooked Pitfalls of Weight Decay and How to Mitigate Them: A Gradient-Norm Perspective. In Proceedings of the Advances in Neural Information Processing Systems, San Diego, CA, USA, 2–7 December 2024; Volume 36.
31. Smith, L.N. Cyclical Learning Rates for Training Neural Networks. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 27–29 March 2017.
32. Keskar, N.S.; Mudigere, D.; Nocedal, J.; Smelyanskiy, M.; Tak, P.; Tang, P. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv* **2016**, arXiv:1609.04836.
33. Goodfellow, A.C.I. *Deep Learning*; Goodfellow, I., Bengio, Y., Courville, A., Eds.; MIT Press: Cambridge, MA, USA, 2016.
34. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhudinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.

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.