Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.061767

ARTICLE

DMF: A Deep Multimodal Fusion-Based Network Traffic Classification Model

Xiangbin Wang[1], Qingjun Yuan[1,*], Weina Niu[2], Qianwei Meng[1], Yongjuan Wang[1] and Chunxiang Gu[1]

[1]Henan Key Laboratory of Network Cryptography Technology, Information Engineering University, Zhengzhou, 450001, China
[2]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China

*Corresponding Author: Qingjun Yuan. Email: gcxyuan@outlook.com

Received: 02 December 2024; Accepted: 24 January 2025; Published: 16 April 2025

ABSTRACT: With the rise of encrypted traffic, traditional network analysis methods have become less effective, leading to a shift towards deep learning-based approaches. Among these, multimodal learning-based classification methods have gained attention due to their ability to leverage diverse feature sets from encrypted traffic, improving classification accuracy. However, existing research predominantly relies on late fusion techniques, which hinder the full utilization of deep features within the data. To address this limitation, we propose a novel multimodal encrypted traffic classification model that synchronizes modality fusion with multiscale feature extraction. Specifically, our approach performs real-time fusion of modalities at each stage of feature extraction, enhancing feature representation at each level and preserving inter-level correlations for more effective learning. This continuous fusion strategy improves the model's ability to detect subtle variations in encrypted traffic, while boosting its robustness and adaptability to evolving network conditions. Experimental results on two real-world encrypted traffic datasets demonstrate that our method achieves a classification accuracy of 98.23% and 97.63%, outperforming existing multimodal learning-based methods.

KEYWORDS: Deep fusion; intrusion detection; multimodal learning; network traffic classification

1 Introduction

With the widespread adoption of the Internet and rapid advancements in information technology, encrypted traffic has become increasingly prevalent in network communications [1]. It plays a crucial role in protecting user privacy and data security, while also supporting various network applications such as secure communication, online payments, and cloud storage [2,3]. However, the proliferation of encrypted traffic presents significant challenges for network traffic monitoring, analysis, and management, particularly in the realm of security, where traditional traffic classification methods struggle to effectively identify the content and behavior of encrypted traffic [4–8].

Recently the application of deep-learning technology to encrypted traffic classification has marked a significant advancement in network security [9,10]. By leveraging deep neural networks, researchers can analyze patterns and features of encrypted data transmissions to classify data streams more effectively. In this context, the core function of deep learning is autonomous learning and feature extraction from encrypted data, enabling differentiation between different types of network traffic. For instance, convolutional neural networks (CNNs) [11] are highly effective at processing and identifying spatial features within encrypted traffic, such as packet size patterns and time intervals. Conversely, recurrent neural networks (RNNs) and their variants, including long short-term memory networks (LSTMs) [12,13], excel at handling sequential data, rendering them well-suited for capturing dynamic temporal characteristics within encrypted traffic.

Conventional deep-learning methods often focus on the individual features of encrypted traffic, overlooking the complementarity among these features and their comprehensive utilization. This limitation can potentially undermine the classification effectiveness and generalization ability of the models. Therefore, in recent years, researchers have increasingly turned to Multimodal learning methods for network traffic classification, aiming to address these shortcoming [14–17].

Multimodal learning integrates information from different data sources, providing richer and more multidimensional insights compared to single-modality approaches, which rely on data from a single source [18]. In encrypted traffic classification, for instance, multimodal learning can integrate various types of information such as time-series data, packet sizes, transmission intervals, and protocol types. A significant advantage of this method is its ability to improve the learning capability of complex data structures while enhancing classification accuracy.

However, existing multimodal network traffic classification methods still face significant limitations. Many methods rely on simple late fusion, where different sub-models extract features from various modalities and then concatenate their outputs for classification. Although this approach offers some advantages by utilizing the diverse features of encrypted traffic, it remains inefficient in fully exploiting deep features. Additionally, these methods generally assume that features from different modalities can be simply fused, overlooking the potential heterogeneity between modalities. Modality heterogeneity refers to the structural differences and inconsistencies in features between different modalities, which often make simple feature concatenation methods ineffective at capturing the deep information of each modality. As a result, these methods tend to lack sufficient flexibility and struggle to adapt and optimize the model based on different traffic patterns.

To address these issues, we propose an encrypted traffic classification model based on deep multimodal fusion, named DMF. In contrast to existing multimodal network traffic classification methods, the DMF model progressively completes fusion during the modality feature learning process. The result of each fusion step influences subsequent feature learning and fusion, thereby achieving gradual deep fusion. Specifically, DMF extracts and integrates multi-scale features from network traffic data through deep learning models. This deep fusion approach allows the model to fully learn the relationships between modalities, reducing the impact of modality heterogeneity on learning, thus improving classification performance and generalization ability. It is important to note that extracting multi-scale features is crucial for achieving deep fusion of different feature modalities. However, unlike image, audio, or video data, which naturally possess multi-scale features, network traffic data lacks such features. Therefore, we use convolutional layers with different kernel sizes to generate intermediate feature maps, capturing the multi-scale features within network traffic data. This enables the model to capture information from both microscopic and macroscopic levels, ensuring strong model performance.

Specifically, we employed CNNs with varying receptive field sizes in convolutional and pooling layers to process different modalities of encrypted traffic. These CNNs generate intermediate feature maps at multiple scales to capture the multi-scale characteristics of encrypted traffic. We selected packet length sequences and raw byte sequences as two distinct modalities, representing the semantic features and temporal characteristics of the traffic, respectively.

To achieve a deep fusion of the extracted multi-scale features, we employed a Transformer model. The self-attention mechanism effectively integrates these features and leverages the information within them, enhancing the classification performance of the model. This deep fusion approach enables a more comprehensive utilization of diverse feature information within the encrypted traffic, leading to improved classification accuracy and better generalization capability of the model.

The main contributions of this study are as follows:

1. We designed a multimodal encrypted traffic classification model called DMF, which employs deep fusion for modal integration. This approach enables a more comprehensive use of deep features across modalities compared to conventional methods, enhancing the flexibility and adaptability of the model.
2. We developed a method for extracting multi-scale features from encrypted traffic data using 1D CNNs with convolutional layers of varying kernel sizes to extract intermediate feature maps at different scales. These multi-scale features enable the model to capture information at multiple levels during deep fusion, thereby enhancing feature representativeness and improving overall performance. levels during deep fusion, thereby enhancing feature representativeness and improving model performance.
3. We validated the performance of DMF using real-world datasets. The results demonstrated that the proposed model outperforms conventional multimodal encrypted traffic classification models that rely on standard fusion methods.

The subsequent sections are structured as follows: Section 2 reviews related research on encrypted traffic classification. Section 3 details the overall structure and design rationale of the proposed model. Section 4 describes the model's implementation in detail. Section 5 presents the experimental validation. Finally, Section 6 concludes the paper and Section 7 highlights the study's contributions.

2 Related Work

2.1 Deep Learning Methods

The deep learning-based network traffic classification method utilizes deep learning techniques to analyze and classify network traffic, to identify different types of traffic (such as HyperText Transfer Protocol (HTTP), File Transfer Protocol (FTP), Voice over Internet Protocol (VoIP), etc.) or detect anomalies. This type of method typically involves steps such as data collection, feature extraction, model training, classification, and evaluation optimization. At present, various deep learning models have been applied to encrypted traffic classification, such as CNNs, RNNs, Graph Neural Networks (GNNs), Transformers, etc. [19,20]. Using deep learning for encrypted traffic classification can effectively improve the accuracy and efficiency of classification, especially when dealing with complex and large-scale network data.

Deep learning based methods typically include methods that use features as well as end-to-end methods. The method of using features first extracts features from the total network traffic, and then inputs these features into a deep learning model for classification. Izadi et al. [21] proposed a traffic classification method based on deep learning and data fusion techniques. They used Deep Belief Networks (DBNs), CNNs, and multi-layer perceptrons (MLPs) to process the statistical features of traffic, and then fused the results of the three classifiers using Bayesian decision fusion, effectively improving the accuracy of classification. Aouedi et al. [22] improved the generalization accuracy of the model by combining multiple tree based classifiers using deep learning and non-linear hybrid ensemble methods. Bovenzi et al. [23] applied class incremental learning methods to network traffic classification to cope with rapid changes in network structure and improve the efficiency of model updates.

The end-to-end method directly uses raw network traffic data as input and classifies it through deep learning models without the need for manual feature extraction. This type of method converts network traffic data (such as raw data packets) into a form suitable for model input (for example, traffic data can be converted into time series, matrix, or graphical representations), and then uses deep learning models to directly learn features from the raw data. Lotfollahi et al. [9] proposed a scheme called Deep Packet, which includes two models: Stacked Autoencoders (SAEs) and CNNs. The model automatically extracts features from data and performs classification, reducing reliance on manual feature engineering and improving classification accuracy. Telikani et al. [24] implemented a cost sensitive learning method using raw bytes as model input on two deep learning classifiers, a stacked autoencoder and a CNN. This method effectively mitigates model overfitting and improves classification performance. In end-to-end methods, network traffic data can also be transformed into graph structures, and then GNNs can be used to complete classification tasks. Hu et al. [25] proposed the TCGNN model, which converts network packets into undirected graphs and uses GNN with three different aggregation strategies to learn graph representations, improving classification accuracy. Han et al. [26] encoded the original bytes of the packet header and payload, constructed the network flow into a traffic interaction graph (TIG), and then used a graph neural network with dual embedding layers for learning and classification.

Compared to traditional methods, deep learning based encrypted traffic classification methods can automatically extract features from raw data without relying on manually defined features or rules. In addition, deep learning models can classify traffic without decryption by analyzing the statistical characteristics and patterns of traffic, which can dynamically adapt to changes in network traffic. This makes deep learning based network traffic classification methods often more accurate and robust, suitable for handling traffic classification problems in complex network environments.

2.2 Multimodal Methods

Recent studies have adopted multimodal deep-learning methods to categorize encrypted traffic. By using multimodal learning, which combines data from various sources, these studies enhance the accuracy of classification by overcoming the constraints associated with using single data types. Wang et al. [27] developed a framework named AppNet for classifying encrypted traffic using a multimodal approach. They applied a 1D CNN to derive features from the initial 1014 bytes of the first packet and an LSTM to analyze the time series of packet lengths. These diverse features were combined for the classification task. In a similar vein, Aceto et al. [15] introduced a multimodal deep-learning framework called MIMETIC, which leverages the first 576 bytes of payload and four protocol features. A 1D CNN processed the payload, and a GRU, which is a more streamlined version of RNNs, managed the protocol features. Building on this, they later developed DISTILLER [16], a comprehensive model that performs multiple traffic classification tasks simultaneously using a multimodal, multitasking approach.

Previous methods typically focused on analyzing either the payload of the first packet or byte sequences extracted from multiple packets, resulting in an incomplete view of the entire network flow. Additionally, many current approaches use relatively simple models for byte embeddings, which, while supporting parallel computation, struggle to capture the temporal relationships between bytes or packets effectively. To address these limitations, Lin et al. [17] introduced a multimodal encrypted traffic classification method that leverages a Transformer encoder and a bidirectional LSTM. Their model, PEAN, improves classification by incorporating both raw byte sequences and packet-length information.

These multimodal approaches primarily use feature-level fusion, which can be categorized into two types:

- Early fusion: Features from different sources are concatenated before being input into the model. This method enables early interaction between different features during training but may overlook complex interactions and dependencies among features.
- Late fusion: In this method, the results from each modality are combined at the output stage. Although this method maintains the independence of each modality, it may miss complementary information at the feature level, thus failing to fully exploit the correlations among different modalities.

These fusion methods typically have limited capabilities in handling diverse types of data, particularly nonlinear and complex data relationships. They often rely on fixed data processing flows and feature extraction methods that lack the adaptability to automatically adjust and optimize feature representations based on the inherent data structures. Table 1 presents the modalities and fusion methods used in existing traffic classification methods based on multimodal learning. It can be observed that although the modalities used in these methods differ, they all adopt a late-fusion approach.

Table 1: Network traffic classification methods based on multimodal learning

Model	Modality 1	Modality 2	Modal fusion
AppNet [14]	Raw byte	Packet length sequence	Late fusion
MIMETIC [15]	Payload	Protocol feature	Late fusion
DISTILLER [16]	Payload	Protocol feature	Late fusion
PEAN [17]	Raw byte	Packet length sequence	Late fusion
DMF (Ours)	Raw byte	Packet length sequence	Deep fusion

An analysis of the aforementioned studies highlights the following issues:

- **Inadequacy of traditional classification methods:** Traditional methods for classifying encrypted traffic are increasingly inadequate in today's complex network environments. As network and encryption technologies evolve, existing methods face growing challenges, necessitating more effective alternatives to manage the continuously changing and expanding nature of network traffic.
- **Limitations of feature-based approaches:** Current methods heavily rely on feature extraction and utilization, often focusing on specific features or combinations within encrypted traffic. This approach can limit classification performance by failing to fully exploit the complementarity among different features. Hence, intricate interactions and dependencies may be overlooked, leading to decreased classification accuracy.
- **Issues with modality fusion in multimodal learning:** Existing multimodal learning approaches often employ simplistic and constrained fusion methods, typically involving basic concatenation of information from different modalities or simple aggregation at the output stage. These methods may not fully leverage the potential of multimodal learning, which aims to enhance classification performance by integrating and understanding complex relationships among different modalities.

Research should address these challenges by developing innovative feature extraction methods, more sophisticated modal fusion techniques, and advanced classification algorithms. These improvements will be crucial for effectively addressing the complexities of encrypted traffic classification in modern network environments.

3 Overview of the Proposed Model

The proposed DMF model is a multimodal network traffic classification model based on deep fusion. In contrast to typical late fusion methods, deep fusion integrates data from different modalities at an earlier stage, at the feature level. This approach enables the model to better comprehend and integrate information from various sources at a deeper level. In deep fusion methods, the extraction of multi-scale features is a frequently used and effective technique. However, encrypted traffic inherently lacks natural multi-scale features, which presents the following challenges when applying deep fusion to encrypted traffic:

A. Extraction of multi-scale features

In the computer vision and image processing fields, extracting multi-scale features is essential due to the varying scales at which objects in images appear. However, encrypted traffic data do not naturally exhibit such multi-scale forms. Therefore, we employed convolutional and pooling layers with varying receptive field sizes to extract information at different scales, enabling us to generate multi-scale feature representations across different modalities.

B. Deep fusion of multi-scale features

Through multi-scale feature extraction, different modalities of encrypted traffic data are decomposed into feature representations at different scales. Our goal was to integrate intermediate feature maps of the same scale from different modalities while preserving the correlations among features across scales. To achieve this, we employed a deep fusion approach for modality integration. In contrast to previous methods that directly fuse extracted multimodal features after complete extraction, our method integrates them during the multi-scale feature extraction process, enabling the result of one fusion to influence subsequent feature extraction. Additionally, after multi-scale feature fusion, we applied late-fusion techniques to the outputs of the submodels to achieve optimal integration.

C. Utilization of features after deep fusion for classification

We employed a deep fusion approach for modality integration, where features of different scales were fused during the extraction process. However, for the final classification, these features must undergo a final fusion before being passed to the classification layer. Therefore, building upon multi-scale feature fusion, we applied late-fusion to the outputs of the submodels to achieve optimal integration. The ultimately fused features were then input into the classification layer to effectively perform the encrypted traffic classification tasks.

Based on the preceding description, our approach primarily focuses on the flow perspective, extracting several modalities and utilizing convolutional and pooling layers with varying receptive fields to obtain modal feature representations at different scales. During this process, modality fusion is performed concurrently with feature extraction. The overall structure of the model is illustrated in Fig. 1. The core components of the model are multi-scale features extraction and the deep fusion of these features.

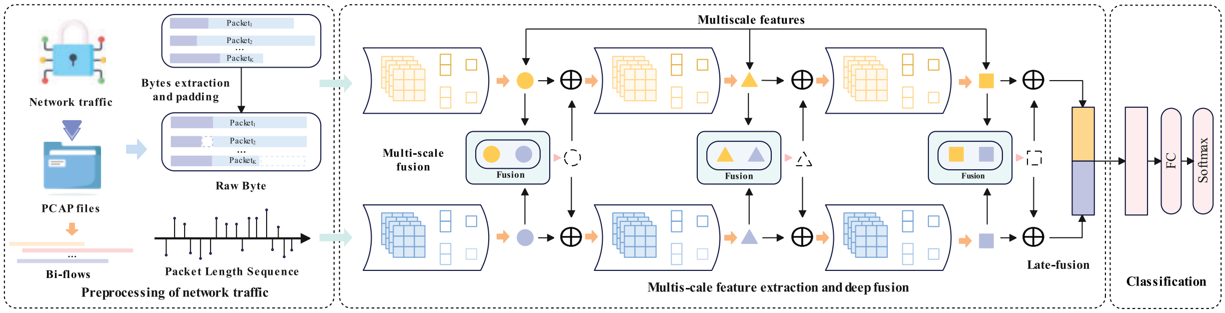

Figure 1: The framework of DMF. The network traffic preprocessing module extracts two modalities, namely the original byte and packet length sequence, and sends them to the two branches of the multi-scale feature extraction and deep fusion module. The circles, triangles, and squares in this module represent the features of different scales extracted by the multi-scale features, while the rectangles represent the output results of the branches. The pink rectangle in the classification module represents the final representation obtained

The overall framework of DMF consists of three main components: traffic preprocessing, multi-scale feature extraction and deep fusion, and the classification module. The preprocessing step primarily focuses on extracting the required modalities while eliminating unnecessary traffic data. The multi-scale feature extraction and deep fusion part is the core of DMF, designed to extract features at different scales for each modality and perform fusion. It employs a progressive deep fusion approach by combining the features after initial fusion. Finally, the classification module uses the fused representations to complete the network traffic classification task.

4 Methodology

This section provides a detailed introduction to the specific methods used in this study, including data preprocessing, multi-scale feature extraction, deep fusion methods, and classification modules.

4.1 Preprocessing of Network Traffic

4.1.1 Bidirectional Flow Extraction and Filtering

In the preprocessing phase, network traffic was organized into bidirectional flows, which serve as the basic unit for classification. A flow is typically defined by five key attributes: source IP, source port, destination IP, destination port, and protocol. Since flows are bidirectional, the source and destination IPs can be considered interchangeable, allowing the system to group traffic in both directions into a single flow. Certain types of traffic were filtered out due to their limited relevance for identifying network behavior. These include:

1. Failed TCP Handshakes: These packets are incomplete and do not contain useful application data, so they are excluded.
2. DNS Queries: DNS traffic often involves different IP addresses and ports, which do not represent typical business transactions. As it has minimal value for analysis, it was removed.
3. LLMNR Protocol: This protocol is used for name resolution in local networks when DNS servers are unavailable, functioning similarly to DNS. It was also excluded for similar reasons.

These types of traffic are often considered background noise and do not provide meaningful insights for the analysis of network behavior [28,29].

Additionally, to enhance the generalizability of the model, we removed explicit protocol identifiers, such as IP addresses, port numbers, Server Name Indication (SNI), and certificates. This step was taken to ensure that the model does not rely on easily recognizable identifiers, which could otherwise simplify classification but may not accurately reflect the underlying network behavior.

4.1.2 Basic Modality Feature Extraction

We extract two types of features from the bidirectional flow: the raw byte sequence and the packet length sequence. Due to the variability in packet size, header length (including the IP header and potentially higher-layer protocol headers), and payload length within the same flow, directly using raw bytes often fails to fully capture the flow's information. For instance, larger packets may occupy the entire input space, while longer payloads might overshadow shorter headers.

To address this, we standardize the packet size by allocating fixed lengths for headers and payloads. Specifically, we select the first$N$ packets from the flow, fix the header length to L_h bytes and the payload length to L_p bytes. Packets exceeding these lengths are truncated, and shorter packets are padded. This results in a sequence of $L_b = N \times (L_h + L_p)$ bytes, represented as $[b_1, b_2, \ldots, b_{L_b}]$.

Although this method preserves a more complete representation of the bidirectional flow's valid information, it disrupts the original packet structure. To complement this, we use the original packet length sequence as the second modality. We extract the packet lengths of the first M packets, truncating flows longer than M packets and padding flows shorter than M packets.

By combining byte-level and packet-length information, the model's input is enriched. This allows the model to capture more meaningful features. It also helps the model better understand the overall structure of the bidirectional flow. As a result, classification performance is ultimately enhanced.

4.2 Multi-Scale Feature Extraction

Encrypted traffic does not possess natural multi-scale features such as image data. Therefore, special methods are needed for extraction. To address this issue, we use convolutional layers with different receptive fields to extract multi-scale features. The specific model structure is shown in Fig. 2. The model first uses MLP to map two modalities to the same dimension, and then sends them to two parallel 1D CNN branches to extract multi-scale features and fuse them. The outputs of the two branches are fused to obtain the final representation for classification tasks.

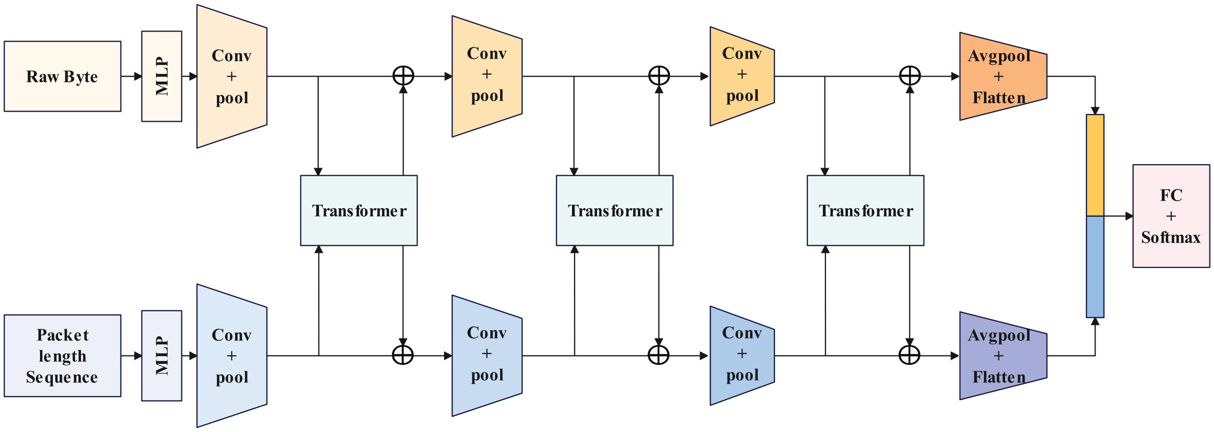

Figure 2: The multi-scale feature extraction and deep fusion model. This model structure uses two CNN structures, represented in blue and orange. Trapezoids are used to represent convolutional and pooling layers, with different colors, depths, and shapes indicating different sizes of convolution kernels used to extract multi-scale features of modalities

As mentioned earlier, raw bytes and packet length sequences are used as two modalities. We use two parallel CNNs to process these modalities. Both networks are 1D CNNs, specifically designed for processing sequential data such as text, audio signals, and network traffic data. By applying 1D convolution operations, these CNNs extracted various features from the sequential data, rendering them well-suited for encrypted traffic analysis. Before inputting the modality into the CNN, we used a shallow multi-layer perceptron (MLP) to unify the dimensions of the two modalities for subsequent multi-scale fusion.

We extracted both local and global features from the modalities using 1D CNN convolution kernels of different sizes to obtain multi-scale features. Specifically, larger convolution kernels help the model capture long-term dependencies in the traffic data [30,31]. For example, 7 × 1 or 9 × 1 convolution kernels can detect traffic patterns that span longer time series; these global features are essential for understanding the overall behavior of the encrypted traffic. Conversely, smaller convolution kernels, such as 3 × 1 or 5 × 1, enable the 1D CNN to capture fine-grained patterns within the traffic data. These smaller kernels focus on time-sensitive features, with the local features they extract reflecting the short-term dependencies of traffic, which are crucial for identifying subtle patterns in encrypted traffic.

4.3 Modal Fusion Based on Deep Fusion

After obtaining the multi-scale features, we performed modal fusion using a deep fusion approach that leveraged the self- attention mechanism of the Transformer. The model combines the intermediate feature maps of two CNNs, which are used to process raw bytes and packet length sequence, respectively. The Transformer processes input as a sequence of discrete tokens, where each token is represented by a feature vector [32]. The self-attention mechanism of Transformer can dynamically adjust the fusion weights between different features to better adapt to different input data. This can effectively integrate features from different modalities and improve the performance of the model.

Formally, we represent the input sequence as $X \in \mathbb{R}^{(n \times d)}$, where n denotes the sequence length, which is also the number of tokens. Each token is represented by a feature vector of dimension d. In attention mechanism, it is necessary to pay attention to how each position in the input sequence affects the representation of the current position. To achieve this, the Transformer calculates queries, keys, and values (denoted as Q, K, and V) as follows:

$$Q = XW^{Q}, K = XW^{K}, V = XW^{V} \tag{1}$$

where $W^{Q} \in \mathbb{R}^{(d \times d_Q)}$, $W^{K} \in \mathbb{R}^{(d \times d_K)}$ and $W^{V} \in \mathbb{R}^{(d \times d_V)}$ are weight matrices. The dot product of Query and Key needs to be calculated to obtain the attention score matrix.

$$A = \frac{QK^{T}}{\sqrt{D_k}} \tag{2}$$

Afterwards, the score matrix is passed through the softmax function to obtain the attention weight matrix.

$$M_{Att} = softmax(A) \tag{3}$$

The attention weight matrix Attention is used to weighted sum the value matrix V to obtain output matrix of Transformer.

$$T^{out} = M_{Att} \cdot V \tag{4}$$

Transformer contains multiple attention heads. Each attention head independently processes input data during computation to capture different features. The outputs of all the heads are concatenated and a linear transformation is performed:

$$M_{multihead} = Concat(T_1^{out}, T_2^{out}, \ldots, T_h^{out})W^{out} \tag{5}$$

where W^{out} is the linear transformation matrix of output.

Then, residual connection and normalization are performed, and the output is obtained through a feedforward network (FFN).

$$F_{trans} = \mathrm{FFN}(LayerNorm(X + M_{multihead})) \tag{6}$$

The above process will continue to be executed during the operation of the two CNN models, used to fuse intermediate feature maps at various scales until the representations of the two modalities are obtained. We also fused the outputs of the two submodels using a classic late-fusion method. The feature maps output by the submodels were concatenated to form the final representation, which was subsequently used for classification tasks:

$$F_{final} = Rep(mod_1) \oplus Rep(mod_2) \tag{7}$$

where $Rep(mod_1)$ and $Rep(mod_2)$ represent the representations of two modalities after deep fusion, respectively.

At this stage, the Transformer was used to fuse the intermediate feature maps, which were then combined with the late-fusion outputs of the submodels, resulting in a deep fusion of the modalities. This approach enabled the model to capture the deep features of encrypted traffic while offering the flexibility to adapt to various types and formats of data. The Transformer processes intermediate feature maps from two modalities by reshaping them into feature maps of the same dimension. These reshaped feature maps are then summed element-wise with the existing feature maps and returned to their respective modality branches. To reduce the computational cost of handling high-dimensional feature maps, average pooling is used to downsample them to a fixed dimension before feeding them into the Transformer. The Transformer's output is subsequently upsampled to the original feature map's dimensions using interpolation and then summed element-wise with the intermediate feature maps in each modality branch.

Both the raw byte sequence and the packet length sequence of network traffic are considered sequential data. The 1D CNN model is effective at capturing local patterns in sequential data. Through the sliding operation of the convolutional kernel, the 1D CNN can effectively extract local dependencies in byte sequences and packet length sequences. For example, in the payload, certain combinations of specific bytes may appear frequently, while in the packet length, changes in packet sizes may indicate specific traffic behaviors.

However, CNNs typically extract features within local regions, and their receptive field is limited, meaning they are not very good at capturing long-range dependencies. Additionally, byte sequences and packet length sequences are inherently different; they are independent in terms of data representation and statistical properties. Simple late-fusion approach may lead to information loss or distortion. Therefore, we use Transformer to fuse intermediate feature maps, thereby constructing a progressive deep fusion approach. The self-attention mechanism of Transformer can globally model the relationships between any two positions in the input sequence. As a result, when using Transformer to fuse the intermediate feature maps extracted by CNN, it can bridge relationships across different layers and modalities, thus enhancing the representational power of the features. This is especially useful in cross-modal feature fusion, as it allows for better capture of the complex long-range dependencies that may exist between the payload bytes and packet lengths.

4.4 Classification Module

By fusing multi-scale features of different modalities, we obtained the final representation of encrypted traffic data and inputted it into the classification module. The loss function in multimodal learning usually includes two parts: modal internal loss and modal fusion loss. Modal internal loss refers to the loss function of a sub model itself, used to optimize its own performance, while modal fusion loss is used to evaluate and optimize the joint representation of multiple modalities. Here, we denote the loss of raw byte modality as $Loss_{rb}$ and the loss of packet length sequence modality as $Loss_{pls}$. We use the cross entropy loss function for both of them, which is calculated using the following formula:

$$Loss_{rb} = -\sum_{i=1}^{n} y_i^1 log[p(x_i^1)] \tag{8}$$

$$Loss_{pls} = -\sum_{i=1}^{n} y_i^2 log[p(x_i^2)] \tag{9}$$

where y_i^1 and y_j^2 are the one hot encoding of the true label, and $p(x_i^1)$ and $p(x_j^2)$ are the predicted probabilities of the i-th class output by the two submodels, respectively.

Afterwards, we calculate the modal fusion loss, and a contrastive loss function is used:

$$Loss_{fusion} = \frac{1}{2N}\sum_{i=1}^{N}(y_i)\cdot D_i^2 + (1-y_i)\cdot max(0, m-D_i^2) \tag{10}$$

where D_i is the distance between modalities, y_i is the label, and m is the boundary threshold.

As shown in Fig. 2, the classification module adopts a fully connected layer combined with Softmax, and the total loss function is

$$Loss_{total} = \alpha Loss_{rb} + \beta Loss_{pls} + \gamma Loss_{fusion} \tag{11}$$

where α, β and γ are weight coefficients used to balance the contributions of different parts.

5 Experiment

This section mainly explains the network traffic dataset used in the study, elaborates on the specific details of implementation, and discusses the evaluation metrics and baseline methods. In addition, it also outlines experiments conducted using the DMF model for encrypted traffic classification. These experiments include quantitative evaluations, ablation studies to determine the impact of various model components, and an analysis of computational complexity. The purpose of these experiments is to evaluate the effectiveness and efficiency of the DMF model in classifying network traffic.

5.1 Experimental Setup

(1) We utilized two publicly available datasets Malicious_TLS [33] and USTC-TFC2016 [34]. The Malicious_TLS dataset comprises encrypted TLS traffic from 22 active malicious code families and benign sources, collected from real networks between 2018 and 2021. Malicious traffic typically has a high degree of diversity and dynamism, which further increases the difficulty of classification. The USTC-TFC2016 dataset contains 10 categories of benign traffic and 10 categories of malicious traffic. We used all 20 categories of traffic data.

(2) Environment settings: The proposed model is implemented using PyTorch 2.5.1 and Python 3.10.14, with all experiments conducted on a Ubuntu 22.04 server equipped with CPU of Intel(R) Xeon(R) Gold 5218R @ 2.10 GHz, GPU of NVIDIA TESLA V100 32 GB.

(3) Evaluation metrics: The performance of the model was evaluated using four key metrics: accuracy, recall, false positive rate (FPR), and precision. The formulas for these metrics are as follows:

$$Accuracy = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \tag{12}$$

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{TN}} \tag{13}$$

$$FPR = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}} \tag{14}$$

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{15}$$

where TP, FP, TN, and FN represent true positives, false positives, true negatives, and false negative, respectively.

(4) Baseline Baseline: To assess the performance strengths and weaknesses of the proposed DMF model, we chose seven encrypted traffic classification models for comparison. Among these, four are based on single-modal learning and two on multimodal learning.

1D CNN: Developed by Wang et al. [27], this model is an end-to-end classification approach that first converts byte sequences into grayscale image values. It then employs a one-dimensional convolutional neural network (1D CNN) to classify network traffic, leveraging the spatial hierarchy in data for effective feature extraction and classification.

XGBoost: Presented by Wang et al. [35], XGBoost stands as a powerful, decision tree-based ensemble model that uses the gradient boosting framework. It's renowned for its performance and speed in training, making it a popular choice across various machine learning tasks for predictive modeling.

ACID: This model, introduced by Diallo et al. [36], focuses on clustering encrypted network traffic. The ACID model is specially designed to handle the complexities of encrypted data streams, providing robust classification capabilities in secure communication environments.

AppNet: Wang et al. [14] introduced AppNet, a multimodal learning framework that processes network traffic data by combining bi-directional Long Short-Term Memory networks (bi-LSTMs) and 1D CNNs. This model uniquely uses initial packet-length sequences and payload bytes to enhance the predictive accuracy of network traffic applications.

MIMETIC: Proposed by Aceto et al. [15], MIMETIC is another multimodal learning framework designed for network traffic classification. It extracts and utilizes four protocol field features from the bidirectional flow of initial data packets, alongside payload bytes. The protocol field features are processed using a Gated Recurrent Unit (GRU), while payload bytes are handled with a 1D CNN, optimizing both temporal and spatial data features for improved classification performance.

The rationale for selecting these baseline models is that they represent different technical approaches and frameworks, enabling a comprehensive evaluation of the performance of various models in encrypted traffic classification. The 1D CNN, by converting byte sequences into image data and using convolutional neural networks to extract spatial features, provides an effective reference for applying image processing methods to traffic classification. XGBoost, as an ensemble decision tree model, has powerful nonlinear modeling capabilities and fast training advantages, making it an efficient benchmark for encrypted traffic classification tasks. Particularly in traditional machine learning methods, it contrasts sharply with deep learning approaches. The ACID model is particularly well-suited for the complexity of encrypted traffic, using clustering methods to process encrypted data streams, thus assessing the classification performance of unsupervised learning and clustering-based approaches in secure communication.

AppNet and MIMETIC represent multimodal learning frameworks; the former combines bidirectional LSTM and 1D CNN, while the latter optimizes temporal and spatial feature processing through a combination of GRU and 1D CNN. Both models effectively fuse multiple traffic features (such as protocol fields and payloads), offering innovative perspectives for efficient encrypted traffic classification. Therefore, these models cover a range of strategies from traditional machine learning to deep learning and multimodal learning, enabling a comprehensive validation of different approaches in complex encrypted traffic, thus assessing whether the proposed DMF model is sufficiently competitive.

(5) Implementation Details: When extracting raw bytes, we take $N = 1$, $L_h = 80$, $L_b = 240$, so for each bidirectional flow, we extract a total of 1600 bytes. In addition, the length of the packet length sequence is $M = 30$. Both MLP models adopt a two-layer structure: for the MLP model that processes raw bytes, the two

layers are (1600,512) and (512,256), respectively; For the MLP model that processes packet length sequences, the two layers are (30,128) and (128,256), respectively. Two CNN models adopt the same structure, including three convolutional layers with kernel sizes of 9, 7, and 5, and stride sizes of 3, 2, and 2, respectively. The values of α, β and γ are all 1.

5.2 Comparison with Baseline

We compared the classification performance of the DMF model against the baseline methods, which included the top-performing methods in current research on encrypted traffic classification. Tables 2 and 3 present the classification performance of various models. On two datasets, we randomly divided the selected encrypted traffic into training set, validation set, and test set in a ratio of 8:1:1, and obtained the average results through multiple independent experiments. Specifically, the results are the averages of 30 independent experiments. Most baseline methods achieved high scores across all four indicators, demonstrating their claimed state-of-the-art (SOTA) performance. However, the proposed DMF model outperforms the other methods across all metrics, particularly excelling in accuracy and showing a significant improvement in classification performance.

Table 2: Comparison experimental results on USTC-TFC2016

	Method	Accuracy	Recall	FPR	Precision
Non-multimodal	1D CNN	90.37	91.63	0.37	90.76
	XGBoost	86.34	85.88	0.39	85.27
	ACID	91.94	92.31	0.28	92.07
Multimodal	AppNet	93.89	93.26	0.22	93.56
	MIMETIC	95.31	95.87	0.23	94.92
	DMF	98.23	97.89	0.22	98.12

Table 3: Comparison experimental results on Malicious_TLS

	Method	Accuracy	Recall	FPR	Precision
Non-multimodal	1D CNN	89.68	88.76	0.39	90.14
	XGBoost	84.27	85.25	0.42	83.82
	ACID	92.16	92.73	0.25	91.55
Multimodal	AppNet	93.24	92.87	0.24	93.15
	MIMETIC	94.26	94.78	0.24	94.03
	DMF	97.63	98.19	0.21	97.54

From the experimental results presented for both datasets, USTC-TFC2016 and Malicious_TLS, it is evident that multimodal models generally outperform their non-multimodal counterparts in terms of classification performance. A key observation is that models designed to leverage multiple features or modalities, such as AppNet, MIMETIC, and DMF, consistently achieve higher accuracy, recall, precision, and lower false positive rates (FPR) across both datasets.

In the case of USTC-TFC2016, a benign and malicious traffic classification task, the DMF model achieved the highest performance across all metrics with an accuracy of 98.23%, a recall of 97.89%, and a precision of 98.12%. Compared to the non-multimodal models, such as 1D CNN, XGBoost, and ACID, which showed accuracy values ranging from 86.34% to 91.94%, DMF shows a clear superiority in classification, particularly in handling the diverse and dynamic nature of malicious traffic.

Similarly, for the Malicious_TLS dataset, which includes encrypted traffic data from both benign and malicious sources, DMF again outperforms other models with an accuracy of 97.63%, a recall of 98.19%, and a precision of 97.54%. This performance is significantly better than non-multimodal models, with ACID achieving a top performance of 92.16% in accuracy, and XGBoost at the lower end with 84.27%. This difference in performance suggests that the multimodal models, especially DMF, are better suited to handle the complexity and variability in malicious traffic, which may be less predictable and more dynamic, particularly in encrypted TLS traffic.

DMF has better classification performance compared to other models on both datasets. We attribute this success to two main factors: First, in contrast to general single-modal methods, DMF, as a multimodal method, can leverage a broader range of features. By utilizing the complementarity among these features, the model gains access to more comprehensive information from encrypted traffic, thereby enhancing classification performance and improving model generalization. Second, compared to other multimodal methods that rely on simple feature-level-fusion, DMF uses deep fusion for modality fusion. This facilitates the model to combine data from different modalities at an earlier stage, enabling it to learn more complex and abstract feature representations. This deeper level of feature integration helps the model better understand the overall data structure, typically proving more effective than processing each modality separately and then merging them.

5.3 Ablation Studies

We conducted three sets of ablation experiments, which include modality ablation and modality fusion method ablation. Modality ablation refers to the exclusion of either the packet length sequence or the raw bytes, making the model a unimodal model. Modality fusion method ablation refers to the removal of the model's multi-scale fusion module, causing the modality fusion method to degrade to the classic late fusion approach. The experimental results of the ablation studies on two datasets are shown in Tables 4 and 5.

Table 4: Experimental results of ablation on USTC-TFC2016

Model	Accuracy	Recall	FPR	Precision
w/o Packet length sequence	94.26	93.74	0.31	93.65
w/o Raw byte	92.37	93.08	0.36	92.79
w/o Multi-scale fusion	93.25	92.86	0.34	93.19
DMF	98.23	97.89	0.22	98.12

Table 5: Experimental results of ablation on Malicious_TLS

Model	Accuracy	Recall	FPR	Precision
w/o Packet length sequence	92.45	91.87	0.36	92.43
w/o Raw byte	91.26	91.73	0.38	90.87
w/o Multi-scale fusion	93.64	93.19	0.31	93.86
DMF	97.63	98.19	0.21	97.54

As shown in Table 4, on the USTC-TFC2016 dataset, when the packet length sequence is removed, the model's accuracy drops to 94.26%, which is a decrease of approximately 3.37% compared to the full model (97.63%). Both recall and precision also decrease, but the FPR (False Positive Rate) slightly increases, indicating that the model, after losing the packet length sequence, becomes less accurate but more conservative when identifying negative samples. When the raw bytes are removed, the model's accuracy drops to 92.37%, a decrease of about 5.86% compared to the full model. Both recall and precision decrease, and the FPR increases to 0.36, suggesting that this modality significantly impacts the model, particularly in improving accuracy and reducing false positives. After removing the multi-scale fusion module, the accuracy drops to 93.25%, a reduction of approximately 4.98% compared to the full model. Recall and precision also decrease, and the FPR slightly increases, indicating that multi-scale fusion is crucial for enhancing the model's performance, especially in the effective integration of modality information.

As shown in Table 5, on the Malicious_TLS dataset, when the packet length sequence is removed, the model's accuracy drops to 92.45%, a decrease of approximately 5.18% compared to the complete model. Both recall and precision decrease slightly, and the False Positive Rate (FPR) increases slightly, indicating that the packet length sequence has a significant impact on the model's accuracy and false positive control. When the raw bytes are removed, the accuracy drops to 91.26%, a decrease of 6.37% compared to the complete model. Both recall and precision show a certain decline, and the FPR increases further, suggesting that the raw byte features are particularly important for the model's performance, and their absence significantly affects the model's predictive ability. After removing multi-scale fusion, the model's accuracy drops to 93.64%, a decrease of 4.01% compared to the complete model. Recall and precision are quite close to the complete model, but the FPR is lower, indicating that multi-scale fusion also has a positive effect on the model's precision and false positive control.

The experimental results on both datasets indicate that multimodal learning and better modality fusion methods play a significant role in improving model performance. By integrating information from different modalities, the model can capture more comprehensive and detailed features, ultimately leading to a substantial performance boost. Therefore, adopting effective modality fusion strategies, especially those capable of handling information from different scales, can achieve superior performance in traffic classification tasks.

5.4 Complexity Analysis

In this section, we analyze the computational and space complexities of DMF.

The computational complexity of DMF can be broken down into two key components: that resulting from the CNN and that from the Transformer encoder.

For CNNs, the main focus is on the complexity of convolution and pooling operations. The complexity of each convolutional layer is $O(L \cdot C_{in} \cdot C_{out} \cdot K)$, where L is the sequence length; C_{in} and C_{out} are the numbers of input and output channels, respectively; and K is the size of the convolution kernel. The complexity of the pooling layer is $O(L \cdot C_{in})$. For the Transformer, the total complexity of a N-layer Transformer is $O(N \cdot S^2 \cdot d)$, where N is the number of layers; S is the length of the intermediate feature map; and d is the model dimension.

Taking into account the factors discussed, the overall computational complexity of the DMF model is

$$O(L \cdot (L \cdot C_{in} \cdot C_{out} \cdot K) + N \cdot S^2 \cdot d) \tag{16}$$

Regarding space complexity, the primary focus is on the Transformer. Each Transformer encoder layer contains multiple weight matrices (such as those used in the AM for queries, keys, values, and the feedforward network weights), which have a space complexity of $O(S \cdot d)$, where S is the sequence length and d is the model dimension. If there are N layers, the overall space complexity for storing the weights is $O(N \cdot S \cdot d)$.

In addition, the activation values of each layer, which store intermediate calculation results, also contribute to the space complexity. This complexity is $O\left(S \cdot d\right)$ per layer.

Therefore, the overall space complexity of the DMF model is

$$O(N \cdot S \cdot d) \tag{17}$$

The computational complexity represented by Formula (16) covers both convolution operations and feature map processing, which can comprehensively evaluate the computational cost of the model. For deep networks, high channel counts, large convolution kernels, or large feature maps, the computational complexity may be very high, resulting in the need for a large amount of computing resources and memory for training and inference. Therefore, when constructing the model, we used relatively small values for these values, indicating that the model construction is relatively simple, but still able to achieve good classification results. This further demonstrates the advantages of using deep fusion for multimodal network traffic classification. The model space complexity given by Formula (17) mainly reflects the memory requirements for storing feature maps. In fact, memory optimization techniques such as memory sharing and model compression can be used to further reduce the storage space required for the model.

6 Discussion

In this section, we will discuss some limitations of this work and potential avenues for improvement.

Data Impact In real-world scenarios, issues such as noisy data, data imbalance, and traffic bursts may arise, all of which can affect the stability and performance of the model.

Noise may originate from signal interference, data loss, transmission errors, and other factors in the network environment. In severe cases, it can impair the model's ability to correctly identify encrypted traffic, thereby degrading classification performance. For multimodal learning-based encrypted traffic classification methods, noise can disrupt traffic features, leading to the loss of valuable information. Potential strategies to mitigate this include: firstly, selecting robust features by optimizing feature selection to identify those less sensitive to noise, thereby reducing its impact on classification results; secondly, employing reinforcement learning algorithms during training to enhance the model's tolerance to noise; and thirdly, utilizing data cleaning techniques to filter out invalid data.

The issue of data imbalance refers to a situation in the training data where the number of samples for certain categories is significantly lower than for others. In encrypted traffic classification, the sample sizes of normal traffic and malicious traffic are typically unequal. Most of the traffic is normal, while samples of malicious traffic are fewer. This imbalance can cause the model to be biased towards the majority class, making it difficult to accurately identify the minority class. To address this issue, one might consider using resampling methods, weighted loss functions, or adopting more comprehensive evaluation metrics to effectively improve the model's performance on imbalanced datasets and ensure accurate identification of minority class traffic. Alternatively, ensemble learning methods (such as Random Forest, XGBoost) that combine the results of multiple models can enhance the detection capability for minority class traffic.

Traffic bursts refer to a sudden surge in network traffic over a short period, often occurring during attacks or peak hours. Bursty traffic can make it difficult for the model to process large-scale data in a timely manner, and may even lead to performance degradation or crashes due to insufficient resources. To cope with this situation, employing online learning or incremental learning methods can enable the model to update promptly during traffic fluctuations, and enhance adaptability to traffic bursts through time series analysis, thereby effectively meeting the classification demands in high-concurrency scenarios.

Modality Fusion Method Currently, DMF employs a combination of deep fusion and late fusion strategies, which has shown promising results. However, this remains a relatively fixed fusion approach that may not be suitable for some modal features. We recognize that there is room for improvement, particularly in dynamically adjusting the fusion process based on the characteristics of the input data. Initially, a dynamic fusion strategy could be adopted, where the fusion method is dynamically adjusted according to the quality or importance of features from each modality, automatically selecting the most suitable fusion approach to enhance the model's adaptability to different data scenarios. Additionally, in the post-fusion phase, by introducing additional learning layers (such as self-attention modules or fully connected layers), the fused features can be further re-learned and optimized, thereby strengthening the expressiveness of the final representation and improving classification accuracy and robustness. The combination of these two approaches can effectively enhance the model's performance on multimodal data and avoid the limitations of fixed fusion methods.

Attention Mechanism When fusing intermediate feature maps, to reduce computational costs, Transformer employs fixed input and output dimensions. Although the current mechanism effectively captures interdependencies between modalities and enhances the model's ability to detect subtle changes in encrypted traffic, it may impact the model's performance when faced with more complex traffic features. A viable solution is to adopt a sparse attention mechanism, which reduces the complexity of attention calculations by focusing on the most relevant subsets of interactions, thereby improving computational efficiency while maintaining the model's classification performance. Additionally, visualization methods for attention weights can be used to further analyze the model's decision-making process and identify areas for improvement.

7 Conclusion and Future Work

In this paper, we proposed a novel multimodal encrypted traffic classification model, DMF. The model used a 1D CNN with convolutional layers of varying kernel sizes as the base model to extract multi-scale features from different modalities of encrypted traffic data. Additionally, a deep fusion method was employed for modal fusion, which fused intermediate feature maps, while late-fusion combined the outputs of the submodels. By extracting deep features of encrypted traffic in this way, and enabling the model to learn the correlation information among different modalities, the model demonstrated greater flexibility and stability compared to the modal fusion methods used in previous research. Validation experiments on real-world datasets showed that our method outperforms existing multimodal and single-modal encrypted traffic classification methods. This further demonstrated that multimodal methods have advantages over unimodal methods and that deep fusion is more effective than simple late fusion.

In future work, we will further analyze the model under more complex data conditions, such as noisy data and traffic bursts, to test the model's stability. We also plan to explore strategies for reducing computational overhead, such as model pruning or lightweight architectures, to enhance scalability. Moreover, further investigations into the interpretability of multimodal feature fusion will be valuable for gaining deeper insights into the decision-making process of the model, which could improve its practical applicability in real-world network environments. Additionally, we intend to analyze the model's performance under data drift conditions, as understanding how the model adapts to shifts in data distributions is crucial for ensuring its robustness and reliability over time.

Acknowledgement: The authors extend their sincere appreciation to the editor and reviewers for their valuable feedback and constructive comments.

Funding Statement: This work is supported by the National Key Research and Development Program of China No. 2023YFB2705000.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Xiangbin Wang, Qingjun Yuan; analysis and interpretation of results: Xiangbin Wang, Qianwei Meng, Weina Niu; draft manuscript preparation: Xiangbin Wang, Qingjun Yuan, Yongjuan Wang, Chunxiang Gu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data available on request from the corresponding author.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Akbari I, Salahuddin MA, Ven L, Limam N, Tuffin S. A look behind the curtain: traffic classification in an increasingly encrypted web. Proc ACM Meas Anal Comput Syst. 2021;5(1):1–26.
2. Rezaei S, Liu X. Deep learning for encrypted traffic classification: an overview. IEEE Commun Mag. 2019;57(5):76–81. doi:10.1109/MCOM.2019.1800819.
3. Deng M, Zhang K, Wu P, Wen M, Ning J. DCDPI: dynamic and continuous deep packet inspection in secure outsourced middleboxes. IEEE Trans Cloud Comput. 2023;11(4):3510–24. doi:10.1109/TCC.2023.3293134.
4. Xu S, Geng GG, Jin X, Liu D, Weng J. Seeing traffic paths: encrypted traffic classification with path signature features. IEEE Trans Inf Forensics Secur. 2022;17:2166–81. doi:10.1109/TIFS.2022.3179955.
5. Zejdl P, Ubik S, Macek V, Oslebo A. Traffic classification for portable applications with hardware support. In: International Workshop on Intelligent Solutions in Embedded Systems; 2008; Piscataway, NJ, USA: IEEE. p. 1–9.
6. Qi Y, Xu L, Yang B, Xue Y, Li J. Packet classification algorithms: from theory to practice. In: INFOCOM 2009; 2009; Piscataway, NJ, USA: IEEE. p. 1–9.
7. Park JS, Yoon SH, Kim MS. Performance improvement of payload signature-based traffic classification system using application traffic temporal locality. In: Network Operations & Management Symposium; 2013; Piscataway, NJ, USA: IEEE. p. 1–6.
8. Doroud H, Aceto G, Donato WD, Jarchlo EA, Pescape A. Speeding-up DPI traffic classification with chaining. In: 2018 IEEE Global Communications Conference (GLOBECOM); 2018; Piscataway, NJ, USA: IEEE. p. 1–6.
9. Lotfollahi M, Zade RSH, Siavoshani MJ, Saberian M. Deep packet: a novel approach for encrypted traffic classification using deep learning. Soft Comput. 2020;24(3):1999–2012. doi:10.1007/s00500-019-04030-2.
10. Malekghaini N, Akbari E, Salahuddin MA, Limam N, Boutaba R, Mathieu B, et al. Deep learning for encrypted traffic classification in the face of data drift: an empirical study. Comput Netw. 2023;225(1):109648. doi:10.1016/j.comnet.2023.109648.
11. Shi Z, Luktarhan N, Song Y, Tian G. BFCN: a novel classification method of encrypted traffic based on BERT and CNN. Comput Netw. 2023;12(3):516. doi:10.3390/electronics12030516.
12. Yao H, Liu C, Zhang S, Yu CX. Identification of encrypted traffic through attention mechanism based long short term memory. IEEE Trans Big Data. 2022;8(1):241–52. doi:10.1109/TBDATA.2019.2940675.
13. Seydali M, Khunjush F, Akbari B, Dogani J. CBS: a deep learning approach for encrypted traffic classification with mixed spatio-temporal and statistical features. IEEE Access. 2023;11:141674–702. doi:10.1109/ACCESS.2023.3343189.
14. Wang X, Chen S, Su J. App-Net: a hybrid neural network for encrypted mobile traffic classification. In: IEEE INFOCOM 2020—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS); 2020; Piscataway, NJ, USA: IEEE. p. 424–9.
15. Aceto G, Ciuonzo D, Montieri A, Pescapé A. MIMETIC: mobile encrypted traffic classification using multimodal deep learning. Comput Netw. 2019;165(1):106944.1–106944.12. doi:10.1016/j.comnet.2019.106944.
16. Aceto G, Ciuonzo D, Montieri A, Pescapé A. DISTILLER: encrypted traffic classification via multimodal multitask deep learning. J Netw Comput Appl. 2021;183–184(2):102985. doi:10.1016/j.jnca.2021.102985.
17. Lin P, Ye K, Hu Y, Lin Y, Xu CZ. A novel multimodal deep learning framework for encrypted traffic classification. IEEE/ACM Trans Netw. 2023;31(3):1369–84. doi:10.1109/TNET.2022.3215507.

18. Xu P, Zhu X, Clifton DA. Multimodal learning with transformers: a survey. IEEE Trans Pattern Anal Mach Intell. 2022;45(10):12113–32. doi:10.1109/TPAMI.2023.3275156.
19. Li W, Zhang XY, Bao H, Shi H, Wang Q. ProGraph: robust network traffic identification with graph propagation. IEEE/ACM Trans Netw. 2023;31(3):1385–99. doi:10.1109/TNET.2022.3216603.
20. Shen M, Ji K, Gao Z, Li Q, Zhu L, Xu K. Subverting website fingerprinting defenses with robust traffic representation. In: 32nd USENIX Security Symposium (USENIX Security 23); 2023; Anaheim, CA, USA: USENIX Association. p. 607–24.
21. Izadi S, Ahmadi M, Rajabzadeh A. Network traffic classification using deep learning networks and bayesian data fusion. J Netw Syst Manag. 2022;30(2):1–21. doi:10.1007/s10922-021-09639-z.
22. Aouedi O, Piamrat K, Parrein B. Ensemble-based deep learning model for network traffic classification. IEEE Trans Netw Serv Manag. 2022;19(4):4124–35. doi:10.1109/TNSM.2022.3193748.
23. Bovenzi G, Nascita A, Yang L, Finamore A, Aceto G, Ciuonzo D, et al. Benchmarking class incremental learning in deep learning traffic classification. IEEE Trans Netw Serv Manag. 2024;21(1):51–69. doi:10.1109/TNSM.2023.3287430.
24. Telikani A, Gandomi AH, Choo KKR, Shen J. A cost-sensitive deep learning-based approach for network traffic classification. IEEE Trans Netw Serv Manag. 2022;19(1):661–70. doi:10.1109/TNSM.2021.3112283.
25. Hu G, Xiao X, Shen M, Zhang B, Yan X, Liu Y. TCGNN: packet-grained network traffic classification via Graph Neural Networks. Eng Appl Artif Intell. 2023;123(2):106531. doi:10.1016/j.engappai.2023.106531.
26. Han X, Xu G, Zhang M, Yang Z, Yu Z, Huang W, et al. DE-GNN: dual embedding with graph neural network for fine-grained encrypted traffic classification. Comput Netw. 2024;245(5):110372. doi:10.1016/j.comnet.2024.110372.
27. Wang W, Zhu M, Wang J, Zeng X, Yang Z. End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In: IEEE International Conference on Intelligence and Security Informatics; 2017; Piscataway, NJ, USA: IEEE. p. 43–8.
28. Alshammari R, Zincir-Heywood N. Machine learning based encrypted traffic classification: identifying SSH and Skype. In: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications; 2009; Piscataway, NJ, USA: IEEE. p. 1–8.
29. Shahbar K, Zincir-Heywood N. How far can we push flow analysis to identify encrypted anonymity network traffic?. In: NOMS 2018—2018 IEEE/IFIP Network Operations and Management Symposium; 2018; Piscataway, NJ, USA: IEEE. p. 1–6.
30. Ou J, Li Y. Symmetric decomposition of convolution kernels. IEICE Trans Inf Syst. 2019;E102-D(1):219–22.
31. Chen H, Mao H, Li Y. Elliptical convolution kernel: more real visual field. Neurocomputing. 2022 Jul 1;492(4):107–16. doi:10.1016/j.neucom.2022.04.033.
32. Vaswani A, Shazeer NM, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: 31st Annual Conference on Neural Information Processing Systems (NIPS 2017); 2017; Red Hook, NY, USA: Curran Associates, Inc. p. 1–15.
33. jun Yuan Q, Liu C, Yu W, Zhu Y, Xiong G, Wang Y, et al. BoAu: malicious traffic detection with noise labels based on boundary augmentation. Comput Secur. 2023;131(3):103300. doi:10.1016/j.cose.2023.103300.
34. Wang W, Zhu M, Zeng X, Ye X, Sheng Y. Malware traffic classification using convolutional neural network for representation learning. In: 2017 International Conference on Information Networking (ICOIN); 2017; Piscataway, NJ, USA: IEEE. p. 712–7.
35. Wang Z, Ma B, Zeng Y, Lin X, Shi K, Wang Z. Differential preserving in XGBoost model for encrypted traffic classification. In: International Conference on Networking and Network Applications; 2022; Piscataway, NJ, USA: IEEE. p. 220–5.
36. Diallo AF, Patras P. Adaptive clustering-based malicious traffic classification at the network edge. In: IEEE INFOCOM 2021—IEEE Conference on Computer Communications; 2021; Vancouver, BC, Canada. p. 1–10. doi:10.1109/INFOCOM42981.2021.9488690.