Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.049904

ARTICLE

Combo Packet: An Encryption Traffic Classification Method Based on Contextual Information

Yuancong Chai, Yuefei Zhu*, Wei Lin and Ding Li

State Key Laboratory of Mathematical Engineering and Advanced Computing, Information Engineering University, Zhengzhou, 450001, China

*Corresponding Author: Yuefei Zhu. Email: zhuyuefeil@126.com

Received: 22 January 2024 Accepted: 08 March 2024 Published: 25 April 2024

ABSTRACT

With the increasing proportion of encrypted traffic in cyberspace, the classification of encrypted traffic has become a core key technology in network supervision. In recent years, many different solutions have emerged in this field. Most methods identify and classify traffic by extracting spatiotemporal characteristics of data flows or byte-level features of packets. However, due to changes in data transmission mediums, such as fiber optics and satellites, temporal features can exhibit significant variations due to changes in communication links and transmission quality. Additionally, partial spatial features can change due to reasons like data reordering and retransmission. Faced with these challenges, identifying encrypted traffic solely based on packet byte-level features is significantly difficult. To address this, we propose a universal packet-level encrypted traffic identification method, Combo Packet. This method utilizes convolutional neural networks to extract deep features of the current packet and its contextual information and employs spatial and channel attention mechanisms to select and locate effective features. Experimental data shows that Combo Packet can effectively distinguish between encrypted traffic service categories (e.g., File Transfer Protocol, FTP, and Peer-to-Peer, P2P) and encrypted traffic application categories (e.g., BitTorrent and Skype). Validated on the ISCX VPN-non VPN dataset, it achieves classification accuracies of 97.0% and 97.1% for service and application categories, respectively. It also provides shorter training times and higher recognition speeds. The performance and recognition capabilities of Combo Packet are significantly superior to the existing classification methods mentioned.

KEYWORDS

Encrypted traffic classification; packet-level; convolutional neural network; attention mechanisms

1 Introduction

The current network environment is complex and ever-changing, with an accelerated digitalization process, leading to a growing demand for personal privacy and data security among users. This demand has spurred the widespread application of encryption technologies, resulting in a surge of end-to-end hardware devices and applications that employ encryption mechanisms. Most of these use advanced, universally recognized encryption protocols, as well as private, non-standard encryption protocols, and have become the standard practice for protecting data transmission security. These encryption technologies play a vital role in ensuring data resources are protected from leaks and unauthorized access. However, the proliferation of encryption technology also presents new challenges for network traffic monitoring and risk management. For instance, encrypted traffic affects the effective operation of Intrusion Detection Systems (IDS), traffic trend analysis systems, as well as Quality of Service (QoS) and Quality of Experience (QoE) assurance [1]. In recent years, the issue of encrypted traffic identification and classification has attracted a research boom in both academia and industry [2].

With the rapid development and widespread application of encryption technology, traditional identification methods based on visible data content [3], such as those relying on port numbers and packet content, are no longer suitable for identifying and classifying encrypted traffic. Some researchers have attempted to use pattern-matching methods to identify encrypted traffic, for example, by constructing a certificate library to match traffic fingerprints [3–5]. However, with the continuous evolution of new encryption protocols like Transport Layer Security (TLS) and Quick UDP (User Datagram Protocol) Internet Connections (QUIC), the visible text in communications becomes increasingly sparse and obscure, rendering these methods ineffective and obsolete. Faced with this challenge, many scholars have turned to using machine learning and deep learning techniques for encrypted traffic identification and classification [6]. Machine learning relies on expert-designed features for data recognition, requiring significant time and effort for feature extraction, and the model's generalization ability is limited by the quality and range of the features used. This leads to a bottleneck in model training and can even result in misjudgments due to feature issues. Deep learning has achieved significant success in fields such as text classification, image classification, and sentiment recognition. By leveraging its advantage in automatically extracting complex features [7], its application in the field of encrypted traffic identification and classification has shown promising research results and application prospects. However, deep learning models heavily depend on data quality. Especially, insufficient sample preprocessing, imbalanced data samples, and abnormal sample distribution can lead to biased features in the model, resulting in overfitting and anomalously high recognition and classification abilities for specific samples.

In real-world environments, encrypted traffic presents a challenge due to the complexity and diversity of communication quality and communication lines. Achieving rapid and accurate attribute recognition and classification of encrypted traffic remains a difficult problem. In this field, since network behavior often relies on a large amount of data interaction and cannot be accomplished by a single packet, the goal of recognition and classification is to differentiate a set of traffic attributes. To enhance the performance and capability in recognizing and classifying different encrypted traffic attributes, this paper proposes a novel model, Combo Packet. This model uses contextual information as input, aiming to use peer-level input to improve the model's confidence. Its advantages include:

1. By using non-stringent contextual information as input, the model leverages both byte-level features of the current packet and the structural features of contextual information. This enhances the robustness and fault tolerance of the traffic classification model, resulting in improved recognition accuracy.

2. Utilizing a convolutional neural network structure and two types of attention mechanisms, the model focuses on more effective features within the peer-level input, discarding irrelevant features. This further enhances the traffic recognition and classification capabilities.

3. From a practical standpoint, the model is designed to be lightweight. Compared to existing models, it uses fewer parameters, and both training duration and prediction recognition classification time are shorter, yielding higher recognition efficiency.

The remainder of this paper is organized as follows. Section 2 reviews some of the most important and recent methods of encrypted traffic identification and classification based on deep learning, as well as attention mechanism techniques. Section 3 provides a comprehensive overview of the proposed new model, Combo Packet. Section 4 presents comparative and ablation experimental results and discusses these results. Section 5 proposes possible future directions for work based on this model. Section 6 concludes the paper.

2 Related Work

Network traffic classification tasks can be primarily categorized into the following types: First is the encrypted traffic identification task [8], which aims to distinguish traffic as either encrypted or non-encrypted. Second is the traffic representation classification task, which focuses on identifying the various application service types to which the business belongs, such as chat applications, file transfer applications, etc. Third is the traffic application classification task, aimed at identifying the specific application associated with the traffic, such as YouTube, Gmail, etc. [7,9].

In this section, we will review the most important network traffic classification methods. Based on the different technologies used, we categorize these methods into four types: Traffic classification methods based on ports, traffic classification methods based on deep packet inspection, traffic classification methods based on statistical features, and traffic classification methods based on deep learning. We will also introduce the spatial and channel attention mechanisms used in this paper.

2.1 Port-Based Traffic Classification Methods

In the early network environment, distinguishing network traffic was a relatively simple task. By utilizing a correspondence table between port numbers and application types, one could extract the corresponding information from the traffic data header and associate the traffic data with its application type [10]. The Internet Assigned Numbers Authority (IANA) standardized the relationship between applications and port numbers, such as using port number 21 for FTP and port number 80 for Hypertext Transfer Protocol (HTTP). As this method only involves a 2-byte plaintext comparison and is unaffected by encryption, it is simple and efficient, commonly used in firewall configurations for Access Control List (ACL) restrictions [11]. However, with the widespread application of technologies such as Network Address Translation (NAT), port forwarding, random port allocation strategies, and port obfuscation, Madhukar et al. [12] in their research on port-based P2P traffic classification found that 30%–70% of traffic data was misclassified. Moore et al. [13] conducted a series of classification experiments based on port-matching technology, achieving only a 50%–70% classification accuracy. This has led to the realization that port-based traffic recognition methods are gradually becoming unsuitable, and more complex methods are needed to classify current network traffic.

2.2 Deep Packet Inspection-Based Traffic Classification Methods

Research on encrypted traffic classification began in the early 21st century. The pioneering work of Roughan et al. [14] sparked the first wave of interest in classification methods. Addressing the limitations of port-based traffic classification, researchers proposed network traffic classification methods based on Deep Packet Inspection (DPI) [7]. ‘Deep’ in this context means that the method examines not only the packet header of a single packet but also the entire content of the packet, including the header and payload. Once specific key fields are matched in the traffic data, the traffic type can be judged based on the category to which these fields belong. These feature fields are also known as ‘fingerprints’, and usually, a fingerprint feature library is maintained to map fingerprints to traffic categories. This approach avoids the problem of relying solely on port numbers. However, with the widespread development of encryption protocols leading to payload randomization, DPI is no longer applicable to encrypted traffic tasks. Later, van Ede et al. [4] proposed the FlowPrint model, which uses unencrypted protocol field information (such as size, certificates, devices, and time characteristics) to represent each flow. This allows for earlier identification of traffic based on the first few packets of a network flow. Nonetheless, these methods highly depend on the plaintext information visible in encrypted traffic, which can be easily tampered with during transmission, thus losing its correct meaning [15].

2.3 Traffic Classification Methods Based on Statistical Features

In the field of encrypted traffic identification, some scholars have transferred machine learning methods to this area. They extract feature information of traffic data on different dimensions, such as packet size and inter-arrival time, and then selectively train classification models with optimal features, overcoming the drawbacks of port matching and deep packet inspection. Taylor et al. [16] proposed the AppScanner model, which uses statistical features like the mean, variance, maximum, and minimum values of packet sizes to train Support Vector Machines (SVM) and Random Forest classifiers (RF) for application classification. BI-directioNal Dependence (BIND) [17] uses time-related statistical features, avoiding byte-by-byte inspection of packet contents, thus significantly reducing computational complexity. Draper-Gil et al. [18,19] published the ISCX VPN-nonVPN and ISCX Tor-nonTor datasets, employing machine learning models such as C4.5 and K-Nearest Neighbors (KNN) to classify encrypted traffic data based on time-related features in sessions. However, as machine learning methods heavily depend on expert-extracted features and are sensitive to the choice of such features, many scholars have shifted their focus to deep learning methods.

2.4 Deep Learning-Based Traffic Classification Methods

Deep learning models do not rely on manually designed features and can automatically capture characteristics from raw traffic, achieving high accuracy. Consequently, in recent years, the use of deep learning methods for identifying encrypted traffic has gained favor in both academia and industry. Lotfollahi et al. [7] proposed the traffic classification framework Deep Packet based on deep learning methods. This framework integrates the feature extraction and classification stages, eliminating the need for experts to extract network traffic-related features. Deep Packet includes two deep neural network structures: Stacked autoencoders and convolutional neural networks, achieving excellent classification performance in traffic service type classification and traffic application classification tasks. The deep Fingerprinting (DF) model [20] uses convolutional neural networks to automatically extract traffic representations from the sequence of original packet sizes in encrypted traffic, while Flow Sequence Network (FS-NET) [21] employs Recurrent Neural Networks (RNN), both achieving notable results.

2.5 Attention Mechanism

Attention plays a crucial role in human vision, enabling selective focus on important areas through scanning, thus capturing useful information more effectively. Similarly, in computational models, the attention mechanism can assign higher weights to important information among a vast amount of data, thereby reducing the interference of useless information and noise and improving the accuracy of model classification.

Traditional attention mechanisms average the feature information of different positions and subspaces, which can lead to the loss of some important features. However, in the field of encrypted traffic identification, different features of data packets play varying roles in recognition effectiveness [22]. Therefore, traditional attention mechanisms may not meet the needs of encrypted traffic recognition. The literature [23] proposed the simultaneous use of spatial and channel attention mechanisms. This mechanism can selectively enhance features that carry more information across different dimensions while effectively suppressing ineffective features. This allows subsequent utilization of effective features for more efficient traffic identification.

3 Methodology

3.1 Model Architecture

Deep Packet [7] employs Convolutional Neural Networks (CNNs) and Sparse Autoencoders (SAEs) to simply and effectively accomplish the identification and classification of encrypted traffic. Inspired by this approach, we continued with the concept of identifying encrypted traffic through byte-level features of data packets. Building upon this, we increased model inputs, altered convolution rules, and integrated attention mechanisms. This method ensures not only the capture of byte-level features but also the deep-level features of contextual information, without significantly increasing model complexity and achieving better recognition and classification results. Since the identification and classification of encrypted traffic are not merely at the packet level, but essentially about recognizing a set of traffic attributes, the method of using peer-level input to gain confidence is the core idea of this model. This concept also establishes a foundation for the model's application in various scenarios.

In this work, we introduce the Combo Packet system for application identification and traffic characterization classification tasks. This system is an end-to-end architecture based on deep learning, comprising a preprocessing module and a deep learning network module. Fig. 1 illustrates the framework structure of Combo Packet, which receives labeled Pcap files. These files first undergo preprocessing, which involves segmenting, cleaning, anonymizing, padding, and normalizing to transform them into an intermediate input form, i.e., multiple preprocessed contextual data, marked as Packet 1 to Packet N in the diagram. Subsequently, these packets enter the deep learning network, where each Packet undergoes a one-dimensional convolution operation to aggregate individual packet features and employs a spatial attention mechanism to highlight effective features. The system then performs convolution operations on the context to aggregate contextual information features, once again utilizing spatial and channel attention mechanisms to emphasize effective contextual features. Finally, the classification is completed through a series of fully connected networks. The specific details of the work will be explained in detail in the following sections.

3.2 Datasets

For the classification task in this paper, we used the ISCX VPN-nonVPN dataset [18], which provides original encrypted traffic of various categories. To balance the samples, we supplemented some missing flows with the ISCX Tor-nonTor dataset [19] (for the service classification task, supplementing nonTor's P2P traffic to nonVPN's P2P traffic, and for the application classification task, supplementing Browsing-Tor traffic to nonVPN's Tor type traffic).

Based on the services performed or operations executed during traffic capture (such as chatting, file transfer, or video calling) and the applications from which the traffic originates (such as YouTube, Facebook, Netflix, etc.), this paper categorizes traffic into 12 different service types, including 6 types of regular encrypted traffic and 6 types of VPN encrypted traffic. It also categorizes traffic into 17 different application types based on encrypted application categories. This dual classification method allows for a comprehensive analysis of encrypted traffic, thereby verifying the model's performance and recognition capabilities. In addition, based on the number of contextual data packets selected (1, 3, 5), the dataset is further organized, as shown in Table 1.

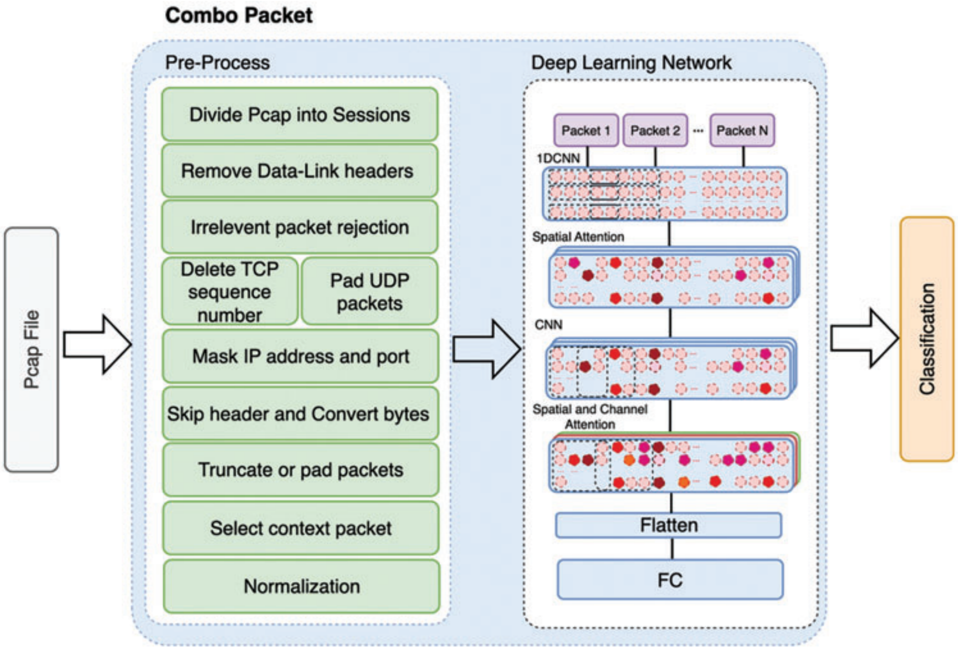

Figure 1: Model architecture

Table 1: The statistical information of the datasets

Dataset	Number of samples	Number of labels
ISCX-Datast-Service_1	60000	12
ISCX-Datast-Service_3	56436	12
ISCX-Datast-Service_5	53761	12
ISCX-Datast-Application_1	82827	17
ISCX-Datast-Application_3	73357	17
ISCX-Datast-Application_5	64148	17

3.3 Pre-Processing

In the experiment, we used the ISCX dataset, captured at the data link layer, which includes Ethernet headers containing Media Access Control (MAC) addresses. These addresses are irrelevant for the classification of encrypted traffic, so our first step was to remove these headers. We then divided the Pcap files based on the five-tuple criteria (source address, destination address, protocol, source port, destination port). This division aimed to gather packets from the context, which in this case is not strictly defined. Next, we removed packets unrelated to encrypted traffic, such as those belonging to Link-Local Multicast Name Resolution (LLMNR), Network Time Protocol (NTP), Network Basic

Input/Output System (NetBIOS), Domain Name System (DNS), Internet Control Message Protocol (ICMP), and packets like Synchronize (SYN), Acknowledge (ACK), and Finish (FIN) from the Transmission Control Protocol (TCP) handshake phase, which only confirm connections and contain no payload. Regarding the transport layer, since UDP headers are only 8 bytes, we padded them to 20 bytes to match the TCP protocol headers. We anonymized TCP's Sequence Number because it precisely locates the next packet, which is not a desirable feature for our model to focus on. We also skipped over the global header (the 24-byte header created when capturing into Pcap files using tools like Wireshark or Tcpdump) and the 16-byte packet header containing timestamps, as this data could lead to the model misclassifying packets captured in close temporal proximity, causing overfitting in the validation and test sets. Next, we converted the packets into byte sequences and truncated them. Considering the wide variation in packet sizes in the dataset, and to keep a consistent length while preserving all packet information, we retained the first 1500 bytes of data after processing, based on common Maximum Transmission Unit (MTU) limits and statistical findings. Each category's sample count was set to 5000, and we implemented a downsampling strategy, splitting the dataset into training, testing, and validation sets in an 8:1:1 ratio. Packets were grouped in sets of N for experimental purposes. For efficient model input, we transformed each byte α of the input data with the $2 \times \alpha/255 - 1$ operation, resulting in input values ranging from $[-1,1]$, facilitating faster model convergence.

Taking the input of 3 contextual packets as an example, Table 2 shows the names of each classification category and their respective sample counts.

Table 2: Distribution of ISCX datasets

Service type	Number of samples	Application name	Number of samples
Chat	5000	Aimchat	1104
Email	4406	Email	3743
File	5000	Facebook	5000
P2P	5000	Gmail	5000
Streaming	5000	Hangout	2693
VoIP	5000	ICQ	5000
VPN-Chat	5000	Netflix	1443
VPN-Email	2030	SCP	5000
VPN-File	5000	Skype	5000
VPN-P2P	5000	Spotify	5000
VPN-Streaming	5000	Tor	5000
VPN-VoIP	5000	Torrent	4374
		Vimeo	5000
		VoIP Buster	5000
		VPN-FTPS	5000
		VPN-SFTP	5000
		YouTube	5000

3.4 Deep Learning Network

3.4.1 Convolutional Neural Networks

CNNs are known for their strong feature extraction and local perception capabilities. Due to the structural properties of their convolutional kernels, they can effectively extract features within their receptive fields. The characteristic of parameter sharing in CNNs significantly reduces the model's complexity and computational demands. Additionally, the simple structure of convolutions makes them easy to integrate with other network structures. Thus, our network employs two consecutive convolutional layers as its backbone, initially extracting byte-level features from individual packets and then capturing structural-level features of context packets, which are pre-aligned at the IP, transport, and application layers.

To illustrate our convolutional behavior with three contexts as an example, the first convolutional operation uses one-dimensional CNNs with 200 filters, each of length 5 and a stride of 3. This setup is designed to capture byte-level features from each input, as depicted in Fig. 2. The second convolutional operation employs two-dimensional CNNs with 200 filters of size (5,3) and a stride of 3. Building upon the byte-level features captured by the first convolutional layer, this layer further captures the structural features of the context packets, as shown in Fig. 3.

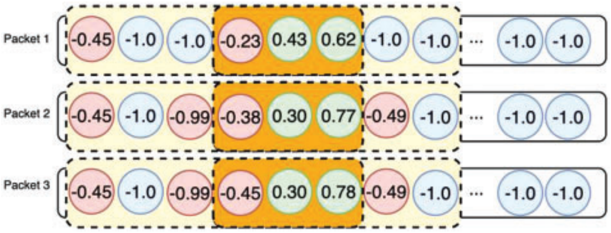

Figure 2: Convolution operation I

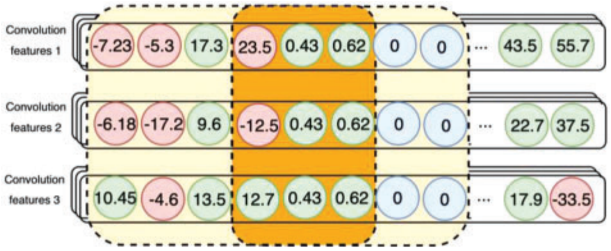

Figure 3: Convolution operation II

3.4.2 Attention Mechanism Network

The spatial attention mechanism is primarily used to enhance a neural network's focus on specific parts of the input data. This mechanism, by assigning different weights to different regions, enables the network to automatically identify and concentrate on features that are more critical for the classification task.

In the recognition and classification of encrypted traffic, the spatial attention mechanism can be applied to identify patterns and features within the encrypted data, even when these aspects are not so apparent in an encrypted context. For instance, by analyzing the size distribution of the encrypted data, the spatial attention mechanism can discern the distribution of traffic and the traffic characteristics of specific applications. This approach enhances classification results by learning the statistical properties of the traffic, as specifically illustrated in Fig. 4.

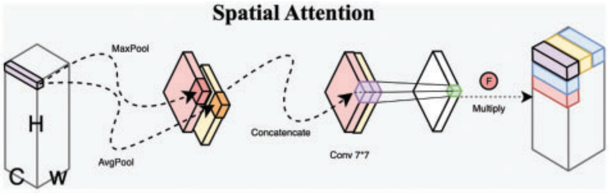

Figure 4: Spatial attention in encrypted traffic classification

The spatial attention mechanism algorithm used in the article is shown as Algorithm 1. It aggregates the channel information of the feature map by performing maximum pooling and average pooling operations along the channel dimension of the input feature map. The aggregated maximum and average features are represented by Fmax and Favg, respectively, creating two overlapping 2D maps. Then, a shared convolutional layer and a Sigmoid function are used to generate the enhanced and suppressed spatial region locations. These locations are then dot-multiplied with the feature input to obtain the weighted feature map.

Algorithm 1: Spatial Attention Algorithm

```
Input:  input feature map I.
Output: weighted feature map O.
Initialize a convolutional layer Conv with a kernel size of $7 \times 7$, number of output channels as 1,
  and sigmoid as the activation function.
Perform max pooling on the input feature map I  to obtain MaxPool.
        $MaxPool = reducemax\ (I,\ axis = 3,\ keepdims = True)$
Perform average pooling on the input feature map I to obtain AvgPool.
        $AvgPool = reducemean\ (I, axis = 3,\ keepdims = True)$
Concatenate MaxPool  and AvgPool  along the channel dimension to get X.
        $X = concatenate\ ([AvgPool,\ MaxPool],\ axis = -1)$
 Apply the initialized convolutional layer Conv to X to obtain the attention map A.
        $A = Conv\ (X)$
Multiply the input feature map I and the attention map A element-wise to obtain the weighted
  feature map O.
        $O = multiply\ ([I,\ A])$
Return the weighted feature map O.
```

In deep learning networks, the channel attention mechanism has become an innovative and effective method. Its core idea is to assign different levels of importance to different feature channels, allowing the network to emphasize those features most crucial to the task.

In the specific field of encrypted traffic classification, the application of the channel attention mechanism is particularly important. Since encrypted traffic inherently conceals the direct content of the data, the channel attention mechanism, by weighting different feature channels of network traffic data, can effectively identify and emphasize those features that are most helpful in distinguishing different types of traffic and suppressing ineffective features [22]. For example, certain channels may better reflect the traffic patterns of specific types of applications, like video streaming or specific behaviors of social media apps. The channel attention mechanism can automatically learn and highlight these channels, thereby improving classification accuracy and efficiency, as specifically illustrated in Fig. 5.

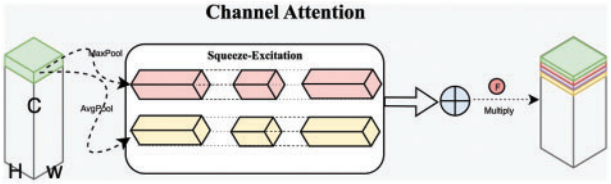

Figure 5: Channel attention in encrypted traffic classification

The channel attention mechanism algorithm used in the article is shown as Algorithm 2. It generates two spatial descriptors, Favg and Fmax, through average pooling and maximum pooling operations over feature map spatial regions. These descriptors are then processed through a shared network composed of Multi-Layer Perceptrons (MLP), which compress and then excite them according to the ratio of a, to obtain the enhanced and suppressed channel locations. Finally, these locations are dot-multiplied with the feature input to obtain a weighted feature map channel. In practical applications, channel attention can be combined with spatial attention to further enhance the performance of processing encrypted traffic.

Algorithm 2: Channel Attention Algorithm

```
Input: input feature map I.
Output: weighted feature map O.
Define a global max pooling layer, MaxPool.
Define a global average pooling layer, AvgPool.
Define two dense layers, Dense1 and Dense2.
Perform global average pooling on the input I to obtain $avg_x$.
        $avg_x = AvgPool\ (I)$
Perform global max pooling on the input I to obtain $max_x$
        $max_x = MaxPool\ (I)$
Reshape $avg_x$ and $max_x$ into four-dimensional tensors.
Use Dense1 to reduce the dimensions of $avg_x$ and $max_x$.
Use Dense2 to increase the dimensions of $avg_x$ and $max_x$.
Add the dimensionally increased $avg_x$ and $max_x$ together to obtain X.
        $X = add\ ([Dense2\ (avg\ _x),\ Dense2\ (max_x)]$
Apply the sigmoid activation function to X to get the excitation.
        $X = sigmoid\ (x)$
Multiply the input feature map I by the excitation X element-wise to obtain the weighted feature
  map O.
        $O = multiply\ ([I,\ x])$
Return the weighted feature map O.
```

4 Experimental Results

4.1 Experiment Setting

The experiments were conducted in an Ubuntu 22.04 operating system environment with Python. The system was equipped with an Intel Xeon E5-2640 CPU, 64 GB of memory, and 2 × NVIDIA TITIAN RTX 16 G GPUs.

All experiments were based on TensorFlow 2.15 and Scikit-learn [24]. The model parameters were set as follows: The batch size was set to 32, a dynamic learning rate was employed, the number of training epochs was set to 40, and Adam was used as the optimizer [25].

4.2 Evaluation Metrics

In this study, to comprehensively evaluate the performance of the hybrid model in encrypted traffic classification tasks, the following four metrics were used: Accuracy (Ac), Precision (Pr), Recall (Rc), and $F1$-score (i.e., $F1$). When assessing the model, TP (True Positive) refers to cases where the model correctly classifies positive samples as positive, TN (True Negative) refers to cases where the model correctly classifies negative samples as negative, FP (False Positive) refers to cases where the model incorrectly classifies negative samples as positive, known as ‘false alarms’, and FN (False Negative) refers to cases where the model incorrectly classifies positive samples as negative, known as ‘misses’. The specific calculation formulas are as follows:

$$\mathrm{Ac} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{1}$$

$$\mathrm{Pr} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{2}$$

$$\mathrm{Rc} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{3}$$

$$F_1 = \frac{2 \cdot \mathrm{Pr} \cdot \mathrm{Rc}}{\mathrm{Pr} + \mathrm{Rc}} \tag{4}$$

Accuracy assesses the overall capability of the classification model, while Precision and Recall reflect the model’s recognition rate in a specific category. The $F1$-score is the harmonic mean based on Recall and Precision, providing a comprehensive evaluation of the model’s performance.

4.3 Comparison Experiment of Different Number of Contexts

To study the impact of the number of contextual packets on the model’s classification accuracy, that is, to determine the appropriate number of context data inputs for the model, we conducted multiple comparative experiments. We selected 1, 3, and 5 contextual packets as subjects for these experiments. By comparing the classification accuracy, precision, and $F1$-score for both service and application categories, it should be noted that the scenario of using a single packet as input is consistent with the Deep Packet model described in the literature.

Fig. 6 shows the change in accuracy over training epochs for 12 different service classification tasks with the selection of 1, 3, and 5 contextual packets as peer-level inputs. Similarly, Fig. 7 illustrates the change in accuracy over training epochs for 17 different application classification tasks with 1, 3, and 5 contextual packets chosen as peer-level inputs.

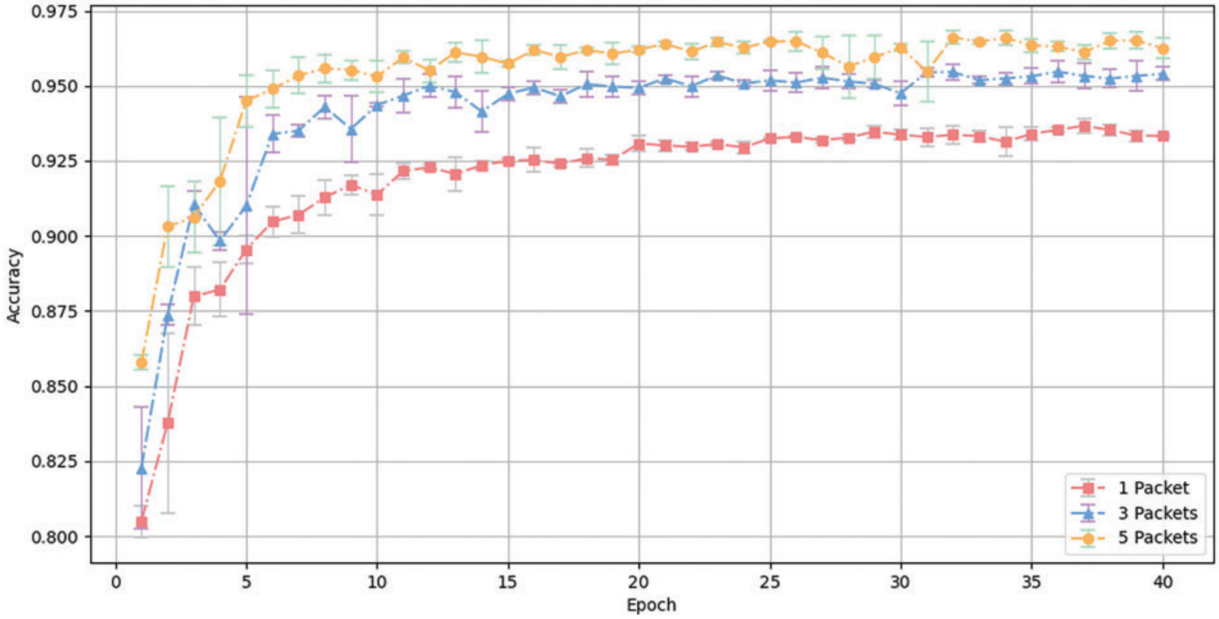

Figure 6: Accuracy change curve over time in service classification

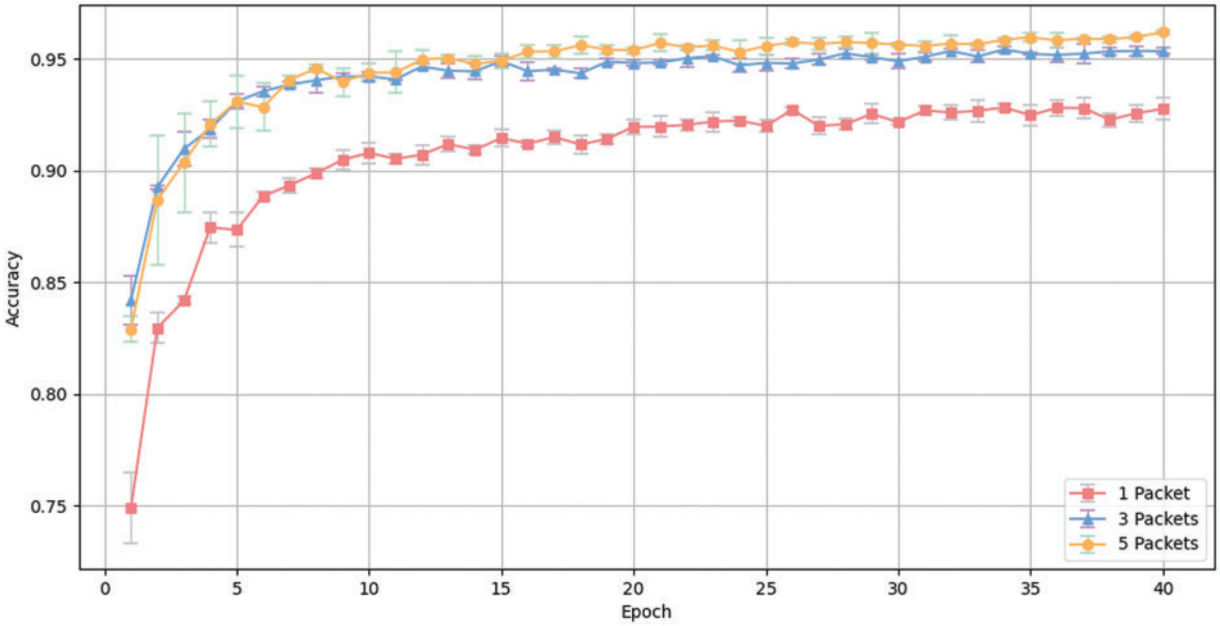

Figure 7: Accuracy change curve over time on application classification

As illustrated in the figures, as the number of contextual inputs increases, the accuracy of the model in both classification tasks improves. The standard deviation continues to decrease, and the final accuracy tends to stabilize.

As shown in Table 3, inputting three contextual packets results in higher accuracy, precision, recall, and $F1$-score compared to a single data input. In service classification tasks, there is an increase of 1.9% in accuracy, 1.9% in precision, 2.0% in recall, and 2.0% in the $F1$-score. In application classification tasks, the increases are 2.5% in accuracy, 1.4% in precision, 1.6% in recall, and 1.5% in the $F1$-score. When inputting five contextual packets, in service classification tasks, the increases are 2.8% in accuracy, 2.5% in precision, 2.6% in recall, and 2.6% in the $F1$-score. In application classification tasks, the increases are 3.3% in accuracy, 2.3% in precision, 2.2% in recall, and 2.3% in the $F1$-score. The experiment indicates that using contextual packets as model inputs enhances the model's ability to recognize and classify, and as the number of contextual data inputs increases, the model's recognition ability also grows.

Table 3: Encrypted traffic services and application classification capabilities

Packet number	Service				APP			
	Accuracy	Precision	Recall	$F1$-score	Accuracy	Precision	Recall	$F1$-score
1	0.933	0.934	0.933	0.933	0.928	0.926	0.926	0.925
3	**0.952 ± 1.9%**	**0.953 ± 1.9%**	**0.953 ± 2.0%**	**0.953 ± 2.0%**	**0.953 ± 2.5%**	**0.940 ± 1.4%**	**0.942 ± 1.6%**	**0.940 ± 1.5%**
5	**0.961 ± 2.8%**	**0.959 ± 2.5%**	**0.959 ± 2.6%**	**0.959 ± 2.6%**	**0.961 ± 3.3%**	**0.949 ± 2.3%**	**0.948 ± 2.2%**	**0.948 ± 2.3%**

4.4 Experiment on the Impact of Attention Mechanisms on Classification Results

As the number of peer-level inputs increases, the accuracy continuously improves. However, the relationship between the number of inputs and accuracy is contradictory because not every set of traffic has the same number of packets. Determining the optimal number of peer-level inputs for model training, and how to further enhance model performance on this basis, are pressing issues that need to be addressed by this method.

The experiment focuses on a manageable number of contextual packets, which is the amount of data that normal communication can afford. Such input determines the versatility of the model. Building on our previous experiments, we introduced Spatial Attention (SA), Channel Attention also called Squeeze and Excitation (SE), and SA + SE modules to explore their impact on model accuracy.

As shown in Table 4, incorporating both spatial and channel attention mechanisms effectively improved the model's accuracy. For the encrypted traffic service classification task, using both types of attention mechanisms increased the accuracy by 1.0% for three contextual inputs and 0.9% for five contextual inputs. For the encrypted traffic application classification task, the accuracy was improved by 0.5% and 1.0%, respectively.

It is worth mentioning that the introduction of spatial and channel attention mechanisms does not impact the main structure of the model or result in a significant increase in parameters. After determining the number of contextual inputs, introducing attention mechanisms is an effective way to further improve accuracy.

At the same time, we recorded the number of incorrectly classified samples by the model on the test set, namely Top-1 Error and Top-5 Error. These respectively represent the number of instances where the model's highest probability prediction is incorrect, and the number of instances where the correct category is not included among the model's top five most probable predictions.

Table 4: The impact of the attention mechanism on the model

Packet number	Service accuracy				APP accuracy			
	Base	+SA	+SE	+SA,SE	Base	+SA	+SE	+SA,SE
1	0.933	–	–	–	0.928	–	–	–
3	0.952	**0.955** ↑	**0.957** ↑	**0.964** ↑	0.955	**0.955**	**0.956** ↑	**0.962** ↑
5	0.961	**0.966** ↑	**0.966** ↑	**0.970** ↑	0.961	**0.966** ↑	**0.967** ↑	**0.971** ↑

Table 5 summarizes the experimental results of Top-1 Error and Top-5 Error after adding the two types of attention mechanisms in encrypted traffic service and application classification tasks. It is evident that the introduction of attention mechanisms did not significantly increase the number of model parameters, and the network outperforms the baseline network. This indicates that the inclusion of attention mechanisms generates more features, endowing the model with better classification capabilities.

Table 5: Model parameters and TOP-1 Error and TOP-5 Error distribution

Model structure	Parameter	Top-1 Error (%)	Top-5 Error (%)
Packet 1 (Service)	3.55 M	6.67	0.40
Packet 1 (APP)	3.55 M	7.21	0.57
Packet 3 (Service)	3.87 M	4.73	0.21
Packet 3 + SA and SE (Service)	3.89 M	3.63	0.14
Packet 3 (APP)	3.87 M	4.47	0.49
Packet 3 + SA and SE (APP)	3.89 M	3.83	0.25
Packet 5 (Service)	4.19 M	3.90	0.26
Packet 5 + SA and SE (Service)	4.20 M	2.87	0.47
Packet 5 (APP)	4.19 M	3.89	0.31
Packet 5 + SA and SE (APP)	4.20 M	2.88	0.30

Figs. 8 and 9 respectively show the error curves during the training intervals for encrypted traffic service classification tasks and encrypted traffic application classification tasks. It is visible that the two curves representing the models with attention mechanisms demonstrate lower training errors. Compared to the baseline model, the incorporation of attention mechanisms exhibits better generalization capabilities.

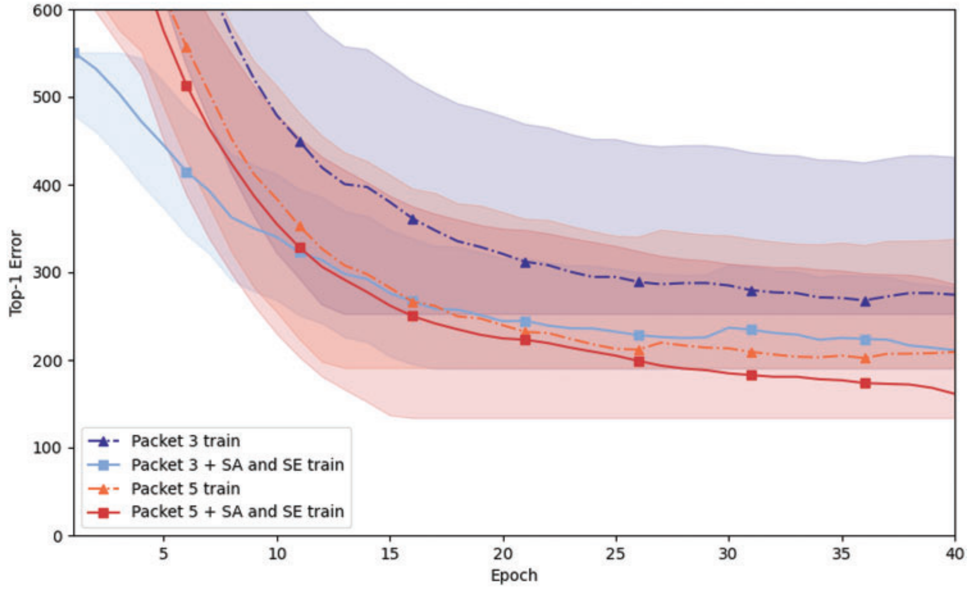

Figure 8: Top-1 Error change curve in service classification

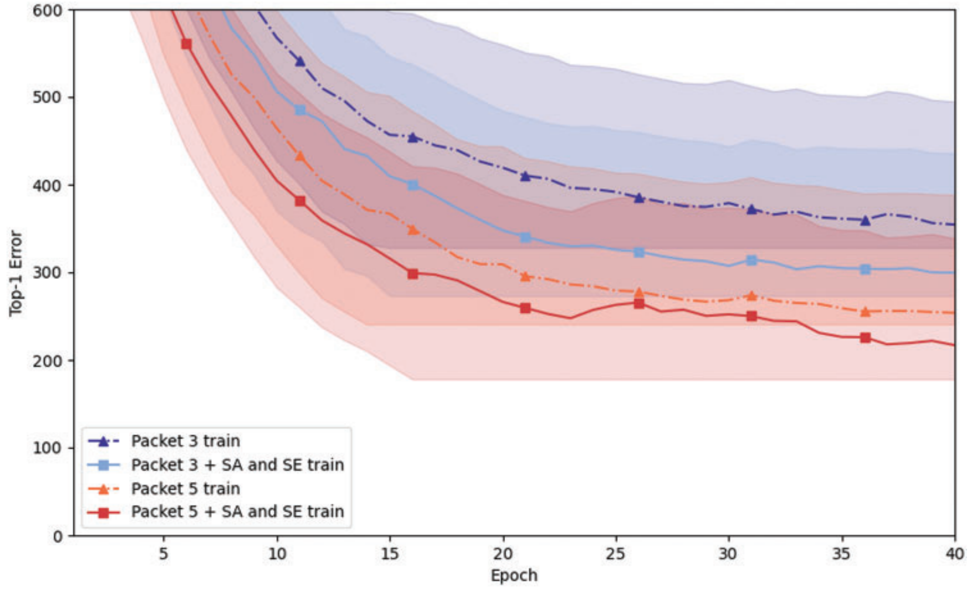

Figure 9: Top-1 Error change curve in application classification

4.5 Experiment Comparing Recognition Abilities and Performance of Different Models

We selected the model that uses five contextual data inputs and incorporates attention mechanisms for comparison with other models. Tables 6 and 7 display the results of service and application classification on the ISCX dataset.

Table 6: Comparison of encrypted traffic service classification capabilities

Method	Accuracy	Precision	Recall	$F1$-score
AppScanner [16]	72.62	73.29	72.15	72.71
BIND [17]	75.44	75.43	74.68	75.05
K-fp [26]	63.70	65.21	64.77	64.99
FlowPrint [4]	78.82	80.22	79.24	79.73
DF [20]	71.64	71.62	71.24	71.43
FS-Net [21]	72.15	75.22	72.58	72.40
Deep Packet [7]	93.29	93.77	93.06	93.41
PERT [27]	93.47	94.00	93.49	93.74
DRCN [28]	95.47	95.73	95.77	95.59
Proposed	**97.04**	**95.66**	**95.97**	**95.81**

Table 7: Comparison of encrypted traffic application classification capabilities

Method	Accuracy	Precision	Recall	$F1$-score
AppScanner [16]	61.35	46.63	50.58	48.52
BIND [17]	66.67	50.47	50.81	50.64
K-fp [26]	59.70	53.48	54.26	53.87
FlowPrint [4]	84.47	65.47	66.21	65.84
DF [20]	60.06	56.66	49.37	52.76
FS-Net [21]	61.37	49.15	49.18	49.16
Deep Packet [7]	92.81	92.61	91.57	92.09
PERT [27]	82.29	72.83	72.66	72.74
DRCN [28]	94.71	94.79	95.11	95.75
Proposed	**97.13**	**95.25**	**95.70**	**95.44**

Our proposed method achieved the best results across all four evaluation metrics, outperforming other models. In the service recognition task, the Combo Packet model attained 97.04% accuracy, 95.66% precision, 95.97% recall, and 95.81% $F1$-score. In the application recognition task, it reached 97.13% accuracy, 95.25% precision, 95.79% recall, and 95.44% $F1$-score. These results fully demonstrate that using contextual information can effectively distinguish between encrypted traffic's service and application types, and also mitigate the issue faced by most models that overly depend on data transmission quality, specifically spatiotemporal features, for recognition.

Figs. 10 and 11 show the confusion matrices of the Combo Packet model using five contextual data inputs with the addition of attention mechanisms. In the service recognition task, the accuracy for each category reached over 90%, but it is noticeable that confusion occurs between Chat and Email. In the application classification task, confusion occurs between FTPS and SFTP. This may be due to the reason described in Deep Packet, where the Euclidean distance is used as a metric to aggregate the similarity between service categories and application types. Chat and Email fall into the same cluster with high similarity, as do FTPS and SFTP, leading to potential confusion.

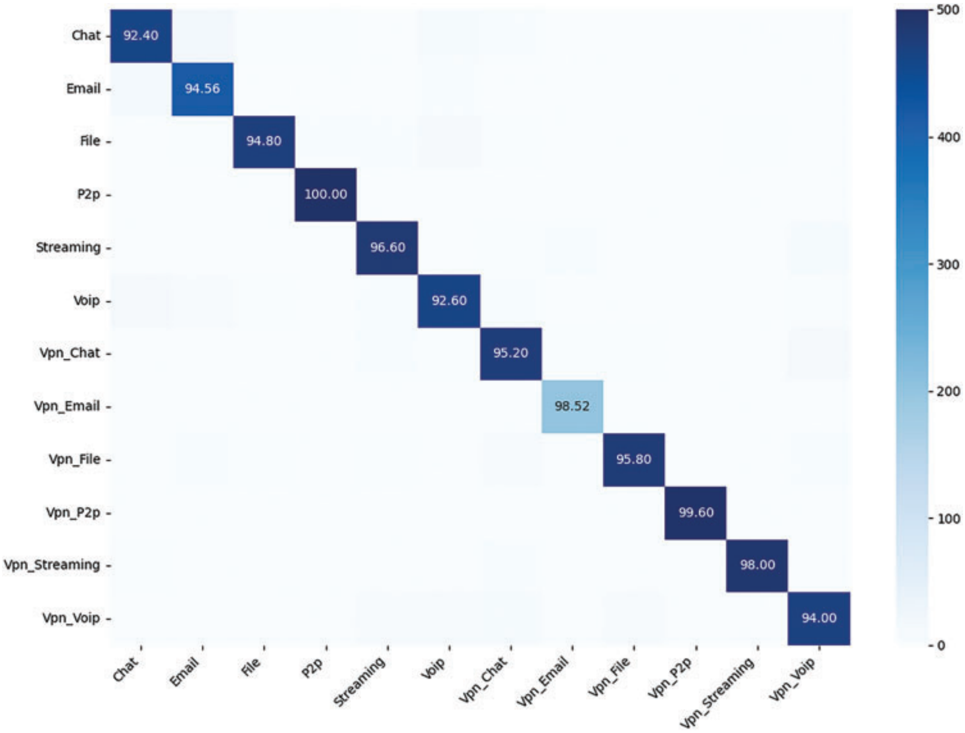

Figure 10: Confusion matrix for encrypted traffic service classification

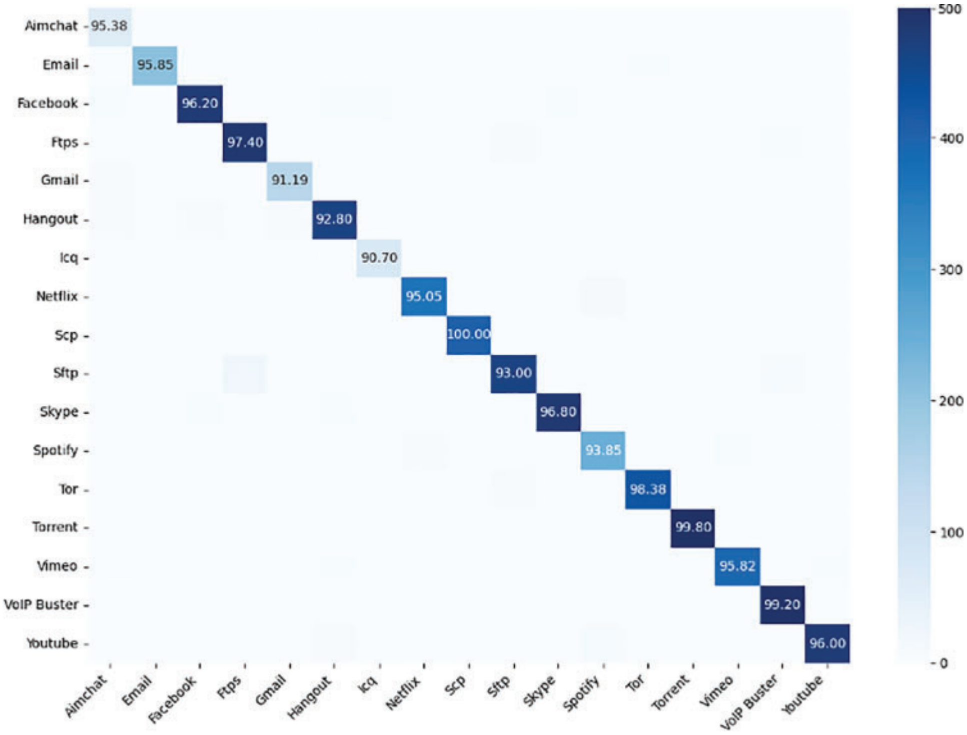

Figure 11: Confusion matrix for encrypted traffic application classification

Finally, to evaluate the training efficiency and recognition speed of the model, we conducted a comprehensive comparison with several classic models including 1D-CNN, BLSTM, CNN + LSTM, Transformer, and Capsule. The results are shown in Table 8. Due to the design of convolutional parameter sharing and local receptive fields, it hierarchically learns the features of input samples, presenting significant advantages in terms of training and recognition time. 1D-CNN has the lowest training and detection time among all the models. Combo Packet's backbone network is based on convolutions, hence it inherits the characteristics of convolutional networks, effectively enhancing the model's training and operational efficiency. Its training and detection times are only second to 1D-CNN, indicating high efficiency in both model training and recognition.

Table 8: Comparison results on application identification tasks

Method	Training time/s	Detection time/s
BLSTM [29]	2416	1.54
1D-CNN [8]	1575	0.56
CNN+LSTM [30]	8420	3.45
Transformer [31]	14150	6.93
Capsule [32]	3552	1.68
Proposed	2216	0.86

5 Future Work

Facing more complex encryption methods and different levels of encrypted traffic, such as link layer encryption and IP layer encryption, the traffic exhibits different patterns. Managing these types of traffic effectively requires continuous in-depth research and innovation in the field. Our proposed model has only been validated on application layer encryption. For different encryption scenarios, the system needs to evolve with the changing environment. Potential future work includes, but is not limited to, the following points:

1. Explore more scientific algorithms for selecting contextual information. For example, using data packets that carry more information or have unique characteristics as input samples to further enhance recognition and classification abilities.

2. Investigate the light-weighting of input data. Our preprocessing captures 1500 bytes of packet data, but whether this length is too long or if capturing less data can still achieve excellent classification results remains to be experimentally verified.

3. Implementing online recognition, as this system does not require spatiotemporal features of traffic and has high training and recognition efficiency, it is feasible to consider further development into an online recognition system. Such a system could be applied to the latest intelligent intrusion detection and traffic trend analysis.

6 Conclusion

This paper proposes an encrypted traffic classification method based on contextual information. The fundamental idea is to use peer-level input to enhance the model's confidence, suitable for encrypted traffic classification scenarios under various network conditions, especially when transmission media changes and transmission quality is low. Based on a simple convolutional neural network, contextual data are used as inputs, and attention mechanisms are integrated to strengthen the model's ability to learn key features. While ensuring model efficiency, this maintains high accuracy, precision, recall, and $F1$-score. Experiments show that our proposed method, compared to other models, improves in computational complexity, recognition efficiency, and classification level. Furthermore, it does not rely on the temporal features of traffic. Combo Packet provides solutions for the next generation of intelligent systems in intrusion detection and traffic trend analysis.

Acknowledgement: The authors would like to express their gratitude for the valuable feedback and suggestions provided by all the anonymous reviewers and the editorial team.

Funding Statement: This study was funded by the National Natural Science Foundation of China Youth Project (62302520).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Yuancong Chai, Yuefei Zhu; data collection: Wei Lin; analysis and interpretation of results: Yuancong Chai; draft manuscript preparation: Yuancong Chai, Yuefei Zhu, Ding Li. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The training data used in this paper was obtained from the UNB ISCX Dataset. Available online via the following link: https://www.unb.ca/cic/datasets/.

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

References

[1] O. Bader, A. Lichy, C. Hajaj, R. Dubin, and A. Dvir, "MalDIST: From encrypted traffic classification to malware traffic detection and classification," in *Proc. 2022 IEEE 19th Annual Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 8–11, 2022, pp. 527–533.

[2] S. Soleymanpour, H. Sadr, and H. Beheshti, "An efficient deep learning method for encrypted traffic classification on the web," in *Proc. 2020 6th Int. Conf. Web Res. (ICWR)*, Tehran, Iran, Apr. 22–23, 2020, pp. 209–216.

[3] P. C. Lin, Y. D. Lin, Y. C. Lai, and T. H. Lee, "Using string matching for deep packet inspection," *Comput.*, vol. 41, no. 4, pp. 23–28, 2008. doi: 10.1109/MC.2008.138.

[4] T. van Ede *et al.*, "FlowPrint: Semi-supervised mobile-app fingerprinting on encrypted network traffic," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, San Diego, CA, USA, Feb. 23–26, 2020, vol. 27.

[5] A. Panchenko *et al.*, "Website fingerprinting at internet scale," in *Proc. NDSS*, San Diego, CA, USA, Feb. 21–24, 2016.

[6] S. Rezaei and X. Liu, "Deep learning for encrypted traffic classification: An overview," *IEEE Commun. Mag.*, vol. 57, no. 5, pp. 76–81, 2019. doi: 10.1109/MCOM.2019.1800819.

[7] M. Lotfollahi, M. J. Siavoshani, R. S. H. Zade, and M. Saberian, "Deep packet: A novel approach for encrypted traffic classification using deep learning," *Soft Comput.*, vol. 24, no. 3, pp. 1999–2012, 2020. doi: 10.1007/s00500-019-04030-2.

[8] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," in *Proc. 2017 IEEE Int. Conf. Intell. Secur. Inf. (ISI)*, Beijing, China, Jul. 22–24, 2017, pp. 43–48.

[9] M. Bhatia, V. Sharma, P. Singh, and M. Masud, “Multi-level P2P traffic classification using heuristic and statistical-based techniques: A hybrid approach,” *Symmet.*, vol. 12, no. 12, pp. 2117, 2020. doi: 10.3390/sym12122117.

[10] A. Dainotti, A. Pescape, and K. C. Claffy, “Issues and future directions in traffic classification,” *IEEE Netw.*, vol. 26, no. 1, pp. 35–40, 2012. doi: 10.1109/MNET.2012.6135854.

[11] Y. Qi, L. Xu, B. Yang, Y. Xue, and J. Li, “Packet classification algorithms: From theory to practice,” in *Proc. IEEE INFOCOM 2009*, Rio de Janeiro, Brazil, Apr. 19–25, 2009, pp. 648–656.

[12] A. Madhukar and C. Williamson, “A longitudinal study of P2P traffic classification,” in *Proc. 14th IEEE Int. Symp. Model., Anal., Simul.*, Monterey, CA, USA, Sep. 11–14, 2006, pp. 179–188.

[13] A. W. Moore and K. Papagiannaki, “Toward the accurate identification of network applications,” in *Int. Workshop Passive Active Netw. Meas.*, Berlin, Heidelberg, Springer, 2005, pp. 41–54.

[14] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield, “Class-of-service mapping for QoS: A statistical signature-based approach to IP traffic classification,” in *Proc. 4th ACM SIGCOMM Conf. Internet Meas.—IMC'04*, Taormina, Sicily, Italy, ACM Press, 2004, pp. 135–148.

[15] X. Lin, G. Xiong, G. Gou, Z. Li, J. Shi and J. Yu, “ET-BERT: A contextualized datagram representation with pre-training transformers for encrypted traffic classification,” in *Proc. ACM Web Conf. 2022*, Lyon, France, Apr. 25–29, 2022, pp. 633–642.

[16] V. F. Taylor, R. Spolaor, M. Conti, and I. Martinovic, “Robust smartphone app identification via encrypted network traffic analysis,” *IEEE Trans. Inf. Foren. Secur.*, vol. 13, no. 1, pp. 63–78, 2017. doi: 10.1109/TIFS.2017.2737970.

[17] K. Al-Naami *et al.*, “Adaptive encrypted traffic fingerprinting with bi-directional dependence,” in *Proc. 32nd Annual Conf. Comput. Secur. Appl.*, Los Angeles, CA, USA, Dec. 5–9, 2016, pp. 177–188.

[18] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, “Characterization of encrypted and vpn traffic using time-related,” in *Proc. 2nd Int. Conf. Inf. Syst. Secur. Priv. (ICISSP)*, Rome, Italy, Feb. 19–21, 2016, pp. 407–414.

[19] A. H. Lashkari, G. Draper-Gil, M. S. I. Mamun, and A. A. Ghorbani, “Characterization of Tor Traffic using time based features,” in *Proc. 3rd Int. Conf. Inf. Syst. Secur. Priv.*, Porto, Portugal, 2017, pp. 253–262.

[20] P. Sirinam, M. Imani, M. Juarez, and M. Wright, “Deep fingerprinting: Undermining website fingerprinting defenses with deep learning,” in *Proc. 2018 ACM SIGSAC Conf. Comput. Commun. Secur.*, Toronto, ON, Canada, Oct. 15–19, 2018, pp. 1928–1943.

[21] C. Liu, L. He, G. Xiong, Z. Cao, and Z. Li, “FS-Net: A flow sequence network for encrypted traffic classification,” in *Proc. IEEE INFOCOM 2019-IEEE Conf. Comput. Commun.*, Paris, France, Apr. 29–May 2, 2019, pp. 1171–1179.

[22] Z. M. Luo, S. B. Xu, and X. D. Liu, “Scheme for identifying malware traffic with TLS data based on machine learning,” *Chinese J. Netw. Inf. Secur.*, vol. 6, no. 1, pp. 77–83, 2020.

[23] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, “CBAM: Convolutional block attention module,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[24] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[25] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.

[26] J. Hayes and G. Danezis, “k-fingerprinting: A robust scalable website fingerprinting technique,” in *Proc. 25th USENIX Secur. Symp. (USENIX Secur. 16)*, Austin, TX, USA, Aug. 10–12, 2016, pp. 1187–1203.

[27] H. Y. He, Z. G. Yang, and X. N. Chen, “PERT: Payload encoding representation from transformer for encrypted traffic classification,” in *Proc. 2020 ITU Kaleidoscope: Industry-Driven Dig. Trans. (ITU K)*, Ha Noi, Vietnam, Dec. 7–11, 2020, pp. 1–8.

[28] G. Z. Shi, K. Y. Li, Y. Liu, and Y. J. Yang, “An encrypted traffic recognition method based on deep residual capsule networks and attention mechanism,” *J. Netw. Inf. Secur.*, vol. 9, no. 1, pp. 32–41, 2023.

[29] H. Yao, C. Liu, P. Zhang, S. Wu, C. X. Jiang and S. Yu, "Identification of encrypted traffic through attention mechanism based long short term memory," *IEEE Trans. Big Data*, vol. 8, no. 1, pp. 241–252, 2022. doi: 10.1109/TBDATA.2019.2940675.

[30] Z. Zou, J. Ge, H. Zheng, Y. Wu, C. Han and Z. Yao, "Encrypted traffic classification with a convolutional long short-term memory neural network," in *Proc. 2018 IEEE 20th Int. Conf. High Perform. Comput. Commun.*, 2018, pp. 329–334.

[31] A. Vaswani *et al.*, "Attention is all you need (NIPS 2017)," arXiv preprint arXiv:1706.03762, 2017.

[32] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 3859–3869.