Encryption Traffic Classification Method Based on ConvNeXt and Bilinear Attention Mechanism

Xiaohua Feng[1], Yuan Liu[2]
School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China[1]
Jiangsu Key Laboratory of Media Design and Software Technology, Wuxi, China[2]

***Abstract*—The rapid growth in internet traffic resulted to the emergence of network traffic categorization as a crucial area of research in network performance and management. This technological advancement has demonstrated its efficacy in aiding network administrators to identify anomalies within network behavior. However, the widespread adoption of encryption technology and the continual evolution of encryption protocols present a novel challenge in the classification of encrypted traffic. Addressing this challenge, this paper introduces an innovative methodology for classifying encrypted traffic by harnessing ConvNeXt and a fusion attention mechanism. Through the representation of traffic data as images and the integration of a bilinear attention mechanism into the model, our proposed approach attains heightened precision in the classification of encrypted network traffic. To substantiate the effectiveness of our methodology, experiments were conducted employing the publicly available ISCX VPN-nonVPN dataset. The experimental findings showcase superior recognition performance, underscoring the efficacy of the proposed approach.**

Keywords—Encryption traffic recognition; end-to-end; convolutional neural network; bilinear attention module

I. Introduction

As the number of Internet-connected devices continues to rise, the overall volume of network traffic is experiencing a significant expansion. Consequently, the significance of network traffic classification in contemporary network management has become increasingly evident. Moreover, with the prevalence of encrypted traffic in modern network applications, the task of classifying encrypted network traffic has assumed a pivotal role in network management. This technology has proven to be effective in assisting administrators in identifying and locating network anomalies, detecting network security threats, and is widely applied in specific areas such as network management, security monitoring, and Quality of Service (QoS) management.

In previous studies, researchers have proposed various methods to tackle the challenge of encrypted network traffic classification. These methods encompass a range of approaches, including port-based methods [1], payload-based traffic methods [2], and machine learning-based approaches [3]. These approaches have made notable contributions to the field, but they also possess certain limitations and drawbacks that warrant further exploration and improvement.

The port classification method, which is based on the transport layer, primarily categorizes ports according to the standards provided by the Internet Assigned Numbers Authority (IANA). For instance, ports like 80 and 443 are commonly utilized as defaults in web services. Port-based solutions, although simple and rapid, have proven insufficient in the present context due to the rise of network protocols and technologies like dynamic ports.

On the other hand, payload-based methods analyze traffic by examining keywords or patterns in data packets. However, the use of encryption technology for network traffic, driven by concerns for security and privacy, renders this method less applicable. In traditional machine learning methods, the performance of algorithms heavily relies on features designed by professionals for different types of traffic. However, in the complex traffic landscape of today, these designed traffic features are unable to cover all categories of traffic.

In recent years, with the significant development of deep learning in the field of image processing, many researchers have also applied deep learning to traffic classification. Deep learning reduces the need for manually designed features by directly learning features from the complex traffic patterns. This approach has shown promise in improving the accuracy and efficiency of traffic classification.

Nevertheless, with the rapid growth of the Internet and the increasing complexity of network applications, traditional traffic classification methods are facing more and more challenges. There is a need for innovative approaches and methodologies that can effectively handle the evolving nature of network traffic. Integrating interdisciplinary methods and exploring new theoretical frameworks may offer potential solutions to overcome these challenges and enhance the accuracy and efficiency of traffic classification in the future.

Deep convolutional neural networks (CNNs) have made remarkable progress in computer vision and natural language processing due to the advancements in deep learning. In the domain of network traffic classification, convolutional neural networks are also widely applied. A notable architecture ConvNeXt [4], has demonstrated comparable performance to models like ResNet while maintaining a relatively low number of parameters. Additionally, the self-attention mechanism, proposed by Google, has emerged as an alternative to traditional recurrent neural networks (RNNs), offering lower computational complexity and enhanced focus on crucial features.

The paper presents a novel model that integrates ConvNeXt with attention processes to improve the precision of network traffic categorization. This paper introduces many novel advancements:

- A bilinear attention mechanism module is proposed that effectively enhances the accuracy of convolutional neural networks in fine-grained tasks. This module enables the model to concentrate on important attributes and enhances the overall performance.
- A novel method for encrypting traffic classification based on the ConvNeXt fusion attention mechanism. By leveraging information from different scales in the network, ConvNeXt improves model performance. The introduction of the attention mechanism further enhances the model's ability to accurately learn key information in network traffic, prioritizing features that are crucial for classification.
- Adapt the new ConvNeXt model's parameters to traffic classification methods and conducting experiments on multiple datasets. Through these experiments, the feasibility and effectiveness of the proposed method in traffic classification tasks are demonstrated.

This paper is organized into four Sections. Section II provides a review of prior work and introduces the ConvNeXt network. Section III presents the proposed methodology, including the bidirectional linear attention mechanism module. Section IV focuses on the experimental aspects, describing the setup, dataset, and analysis of results. Finally, this work is concluded in the Section VI where possible future research directions are indicated.

II. Related Work

Previous studies have proposed numerous mature methods for network traffic classification. This section will elaborate on these classification methods, providing an overview of the approaches developed by researchers. It aims to provide a concise yet comprehensive understanding of the existing techniques used in this field.

A. *The Methods Based on Payload and Machine Learning Approaches*

With the emergence of new network protocols and dynamic ports, traditional port-based classification methods have become inadequate for the current network environment. In response, researchers have proposed Deep Packet Inspection (DPI) technology. DPI technology goes beyond traditional packet inspection of the first four layers of IP packets and reads and reassembles the application layer data to achieve traffic classification. P. Khandait et al. [5] introduced a system called Length-Based Matching (LBM) which includes an innovative acceleration approach for RegEx matching. Wang et al. [6] introduced a framework called lightweight Deep Packet Inspection (LW-DPI). S. Fernandes, R. Antonello et al. [7] introduced a Bitcoding system, which is a method for generating traffic classification signatures at the bit level using DPI. These paper findings have shown commendable performance. However, DPI technology faces two inherent challenges.

The first challenge is privacy concerns. In the current era of the Internet, users have become increasingly concerned about the privacy of their transmitted data. DPI technology involves reading the content of user transmissions, which inevitably violates user privacy. This raises ethical and legal concerns, as users expect their data to remain confidential and secure.

The second challenge is encryption. In response to the growing apprehension among users regarding the confidentiality of network traffic, a significant number of products have adopted the practice of encrypting their network traffic. According to Google's Transparency Report, over 90% of traffic in Google's products is encrypted. Encrypted traffic does not allow its data to be read, rendering DPI detection methods ineffective for classification. As more and more internet traffic becomes encrypted for security reasons, the limitations of DPI technology become increasingly apparent.

In light of these challenges, machine learning has emerged as a promising solution to the traffic classification problem. In stark contrast to DPI, machine learning-based methods classify traffic by learning statistical features from the traffic data. Classification schemes in classical machine learning are classified according to the degree of supervision, which includes supervised, unsupervised, and semi-supervised methods. Using supervised machine learning, K.L. Dias [8] explored real-time video traffic to categorize real-time applications. Feature engineering and classifier construction were employed by Cao et al. [9] in order to increase the accuracy of traffic categorization in the SVM-based model. Using K-means clustering as a tool for unsupervised learning, Wang et al. [10] examined the effectiveness of grouping network flows based on similarity using the algorithm. Höchst et al. [11] developed a method for classifying traffic flow by utilizing statistical characteristics derived from a neural autoencoder algorithm. Using semi-supervised learning, B. Ghita et al. [12] used two-phase learning approach for traffic classification. The semi-supervised method developed by F. Noorbehbahani et al. [13] consists of Clustering using X-means and propagation of labels. However, the effectiveness of machine learning-based classifiers heavily relies on well-designed features. Designing optimal features requires experienced professionals to invest a significant amount of time in manual design. Furthermore, the overall generalization performance of such methods is relatively poor. Additionally, due to the involvement of human intervention, these methods require professionals to adapt the classifier when the current environment changes or when it is not suitable for a different distribution of similar datasets. This reliance on human intervention can be a cumbersome and time-consuming process.

In summary, while DPI technology offers a more in-depth approach to traffic classification, it faces challenges related to privacy concerns and the rise of encryption. Machine learning-based methods, on the other hand, provide a promising alternative, but they require optimal feature design and can be limited by the need for human intervention.

B. *The Methods Based on Deep Learning*

In contrast to machine learning methods, deep learning methods eliminate the reliance on manually designed features. Deep learning operates through an end-to-end process, where raw data is directly inputted into the model. The model then autonomously learns its own features based on the outcomes, facilitating global optimization. Although deep learning necessitates a substantial amount of training data, it plays a pivotal role in reducing human intervention throughout the entire workflow. Employing specialized techniques for raw data processing can effectively enhance performance. Shapira, Tal [14] approached the problem by treating packet size and arrival time in network traffic as correlation graphs. They eliminated packets larger than 1500 bytes, focusing on packets arriving within the first 60 seconds. This mapping generated a 1500x1500 histogram, which was then fed into a convolutional neural network based on the LeNet-5 architecture for classification. Lan, Jinghong [15] and others tackled the issue from multiple feature levels, employing different extraction methods for each feature and subsequently processing the combined features. D'Angelo, Gianni [16] proposed a method of data collection through window sampling, followed by statistical analysis of its features. This approach primarily involved counting various data exchanged within the window, such as the number and length of packets. The information was then inputted into an SAE network for classification.

Achieving satisfactory results can also be accomplished by combining multiple models to identify and classify traffic features. Wang, Wei [17] [18] transformed traffic into images and employed one-dimensional convolutional neural networks and representation learning. Both methods demonstrated strong performance. Maonan, Wang [19] combined ResNet and AutoEncoder models for classification. They divided the data, extracting the original data into images as input for feature extraction in the ResNet network. They also inputted the statistical features of the data flow into a network based on AutoEncoder to extract statistical features. The two sets of features were then combined into complex features for classification.

Additionally, attention mechanisms have proven effective in enhancing model performance. Attention mechanisms can be integrated with traditional convolutional neural networks or recurrent neural networks to augment the capabilities of the original models. Barut, Onur [20] incorporated position and time information for each bit into the original PCAP stream data. They utilized a model that combines multi-head self-attention pooling and 2D CNN for classification. Liu, Xun [21] utilized intercepted packets as input data and employed a model that combines attention and GRU for classifying network traffic. Xiao, Xi [22] utilized side channel data to improve algorithm performance. They inputted this data into a model consisting of RNN and attention mechanisms for classification.

C. ConvNeXt Convolutional Neural Network

With the continuous advancements in deep learning, the field of image classification has witnessed the emergence of new algorithm networks. Among these networks, Swin Transformer has been at the forefront of fine-grained classification since 2020, demonstrating exceptional performance and gradually replacing the conventional convolutional neural networks. However, in 2022, scholars such as Liu and Zhuang [4] proposed ConvNeXt, a novel approach that builds upon the Swin Transformer by studying its layer structure, downsampling methods, activation functions, and data processing techniques. This research has led to significant improvements in the accuracy of convolutional neural networks, reaffirming their crucial role in image classification.

The network architecture of ConvNeXt is not notably distinctive, as it integrates various components that have been previously employed in research. Fig. 1 exhibits similarities to the layer arrangement of ResNet50. ConvNeXt incorporates downsampling techniques and layer normalization, while also sharing commonalities in its recurrence approach and structural combination with ResNet50. Fig. 2 can be juxtaposed with the layer architecture of MobileNetV2. Inverted residuals and the Swin Transformer's MLP structure are combined to form ConvNeXt blocks. ConvNeXt improves the accuracy of coarse-grained image categorization by combining the processing methods of the Swin Transformer with the inherent properties of convolution.

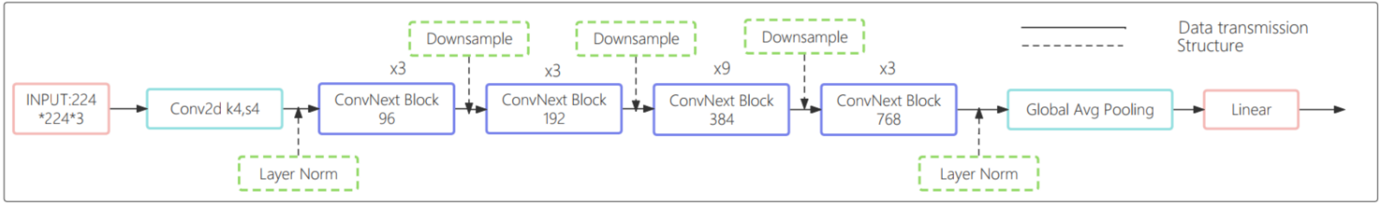

Fig. 1. Network diagram chart of ConvNeXt.

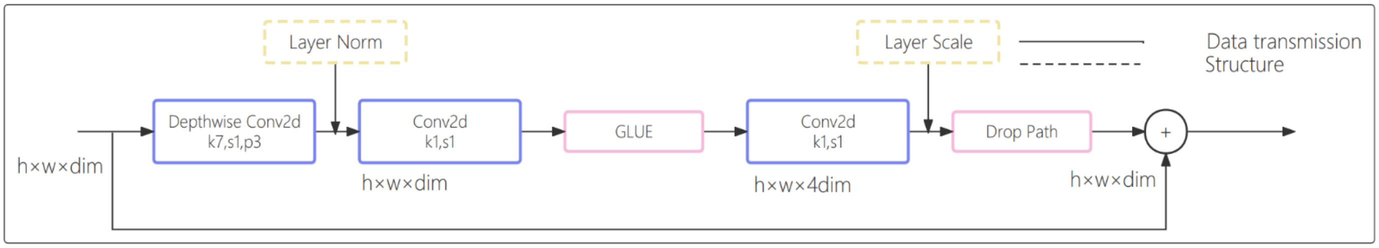

Fig. 2. Structure diagram of ConvNeXt block.

III. Methods

This section provides an overview of the dataset utilized in this paper, the data processing methodology, the construction of the model, and the parameter configuration within the model.

A. *Data Preprocessing*

The dataset utilized in this work consists of capture files, with each file corresponding to a distinct program, traffic type, and encryption method. These capture files are stored in the PCAP format. The initial 24 bits of a PCAP packet contain crucial data information, including a 4-byte file magic number, a 2-byte major version number, a 2-byte minor version number, a 4-byte local timezone, a 4-byte timestamp, a 4-byte maximum storage, and a 4-byte link type. Subsequently, each file consists of a combination of packet headers and packet data. The packet header is composed of four 4-byte fields, namely the high-order timestamp, low-order timestamp, current packet length, and offline data length. The data structure is visually depicted in Fig. 3.

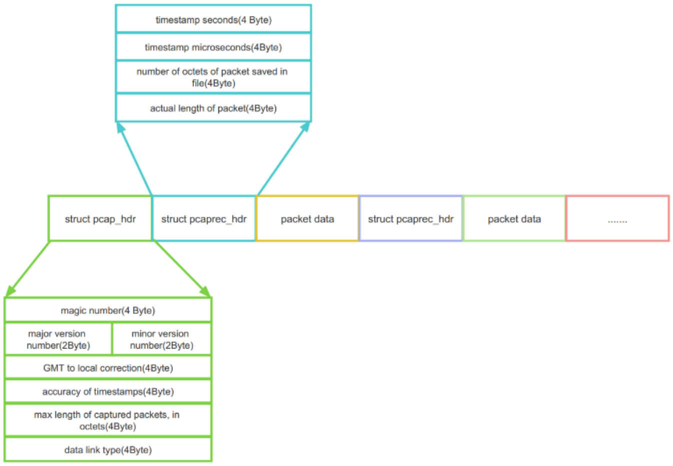
timestamp seconds(4 Byte)
timestamp microseconds(4Byte)
number of octets of packet saved in file(4Byte)
actual length of packet(4Byte)
struct pcap_hdr
struct pcaprec_hdr
packet data
struct pcaprec_hdr
packet data
....
magic number(4 Byte)
major version number(2Byte)
minor version number(2Byte)
GMT to local correction(4Byte)
accuracy of timestamps(4Byte)
max length of captured packets, in octets(4Byte)
data link type(4Byte)

Fig. 3. Structure of pcap file.

In the data cleaning section, this paper undertakes the removal of IP addresses and port information from the source data. This step is crucial to prevent the model from relying on these details and making incorrect judgments, thereby eliminating biases associated with IP addresses and ports. Additionally, as the data remains in the PCAP format even after being divided into flows, the PCAP header also needs to be eliminated during the data cleaning process.

This work utilizes the five-tuple {source IP, source port, destination IP, destination port, protocol} to divide each PCAP file into numerous unidirectional flows. Subsequently, the data undergoes transformation into images through the following key steps:

1) *Data refinement:* Duplicate or blank data can significantly impact the training of deep learning models, introducing biases in learning features and reducing classification accuracy. Hence, it is imperative to remove any duplicate or blank content present in certain packets.

2) *Privacy processing*: In order to diminish the model's reliance on IP addresses within the data, this paper employs zero-padding on the IP-related information by filling them with zeros.

3) *Data trimming:* Based on data analysis, to comprehensively capture feature information of network traffic and achieve more accurate network traffic classification, this paper trims the payload data to a fixed length of 576 bytes. If the file size exceeds 576 bytes, the excess portion is deleted. Conversely, if the length is less than 576 bytes, zeros are added at the end.

4) *Data transformation:* Each data segment with a length of 576 bytes is transformed into a 24x24 image to facilitate processing by the model.

The aforementioned steps ensure the cleanliness, privacy protection, and appropriate formatting of the data, enabling effective analysis and classification of network traffic. By applying the above processing methods, 24x24 images of each category will be obtained, and some of the images are shown in Fig. 4.

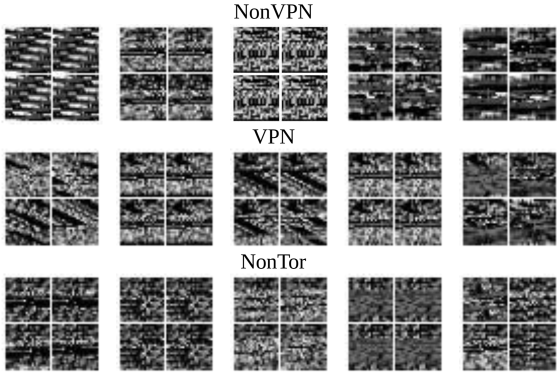

Fig. 4. Partial preprocessed images.

B. *Model Structure*

The primary model utilized in this paper is ConvNeXt-Tiny, selected for its lower parameter count and enhanced feature extraction capabilities, thereby reducing hardware requirements during training. In the task of traffic classification, the information content of the split pcap files is limited. To maximize the inclusion of information, the length of the traffic is truncated to 576 based on an analysis of the length feature of the original data flow. Subsequently, it is transformed into a 24x24 image.

1) *ConvNeXt model incorporating attention mechanism:* The network structure of ConvNeXt comprises four different kinds of blocks, each with a different number of channels. Each block type undergoes a specific number of cycles: the first kind goes through three times, resulting in a feature map with 96 dimensions, the second kind goes through three times, resulting in a feature map with 192 dimensions; the third kind goes through three times, resulting in a feature map with 384 dimensions, and the fourth kind goes through three times, resulting in a feature map with 192 dimensions768. Before each block, the feature map is subjected to a downsampling procedure, which decreases the final output size (W, H) by half compared to the original size, while simultaneously increasing

the output dimension by convolution. Adding weight values to channels is crucial in this scenario.

Precise localization of the target is crucial in tasks that involve fine-grained categorization, as it enables successful feature extraction. In order to enhance the network's capacity to learn particular target features during training, we provide two methods called Embedded Block Attention (EBA) and Sequential Block Attention (SBA), both of which rely on embedded locations. The ConvNeXt-Tiny network incorporates the attention mechanism into each block, enabling the attention mechanism to be integrated into the internal loop of each block type. The term used to describe this integration is EBA. The feature maps of inverted residuals are subjected to attention weights, and attention parameters within each block are periodically trained. This process results in the formation of a feature map with attention weights for each block's cyclic object. This method allows for the iterative training of attention settings. To obtain a comprehensive depiction of the altered structure of the ConvNeXt Block, refer to Fig. 5.

$$\boldsymbol{F_{a1}} = \sigma(\mathrm{Con\ v}(\mathrm{GeL\ U}(\mathrm{L\ N}(\mathrm{Con\ v}(\mathrm{Conca\ t}(\mathrm{Avgpoo\ l}(\boldsymbol{F}), \mathrm{Maxpoo\ l}(\boldsymbol{F})))))) = \sigma(\boldsymbol{W_1}(\mathrm{GeLU}(\mathrm{LN}(\boldsymbol{W_0}(\mathrm{Concat}(\boldsymbol{F^c_{avg}}, \boldsymbol{F^c_{\mathrm{max}}})))))) \quad (1)$$

In contrast to EBA, the integration of the attention mechanism through SBA in ConvNeXt-Tiny does not disrupt the cyclic process of the blocks. Instead, it applies the attention mechanism to the feature maps after the cyclic process of each type of block. This allows the training of attention weights to take into account the entire cyclic feature map, rather than individual blocks. As a result, the number of training iterations required for attention parameters is reduced. Additionally, to address challenges such as degradation, gradient explosion, and gradient vanishing in deep networks, residual shortcuts are employed to add the downsampled feature maps to the output of the attention module based on channels. For a more detailed illustration of the specific structures, (see Fig. 6).

When incorporating the attention mechanism using SBA, it is crucial to observe that the remaining connections of attention, except for the third kind of block, are removed (see Fig. 6). The rationale behind this decision stems from the observation that the third kind of block present in all iterations of ConvNeXt experiences the greatest extensive amount of iterations, specifically nine cycles in the Tiny version. Introducing external residual connections would result in the inclusion of a significant amount of untreated and superficial semantic information into the attention feature map. The integration would ultimately diminish the network's overall ability to extract features and learn.

2) Bilinear attention mechanism: Convolutional Block Attention Module (CBAM)[23] is enhanced in this paper, and a new attention mechanism is proposed to improve the accuracy of ConvNeXt. The attention mechanism also allocates feature weights in ConvNeXt-Tiny. This section offers a comprehensive overview of the planned CBAM enhancement.

This paper proposes a Bilinear CBAM (BLCBAM) by enhancing the CBAM utilizing the bilinear method, building upon the concept of bilinear CNN. Moreover, the Channel Attention Module (CAM) of CBAM incorporates the feature information from both the channel and spatial dimensions to boost its performance. In order to enhance the algorithm described in this research, it is important to take into account the bilinear nature of BCNN. The first max pooling and average pooling processes are kept concurrent to retain the preservation of feature information from the original maps to the maximum degree feasible. Subsequently, the output results are combined together according to the channel. The fully connected layer in the original architecture is substituted with a 1x1 convolutional layer, and batch data processing is conducted using layer normalization. ConvNeXt employs the Gaussian Error Linear Unit (GeLU) activation function, which is identical to the one used in the Swin Transformer. CBAM replaces the ReLU activation function with GeLU, which includes stochastic regularity. The proposed CBAM configuration is shown in Fig. 7.

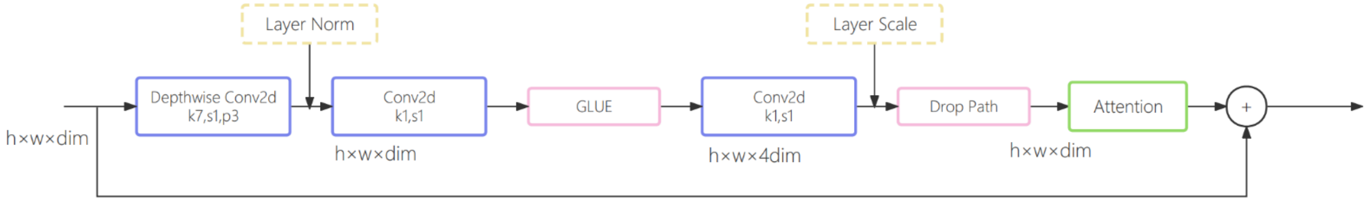

Fig. 5. Structure chart of EBA.

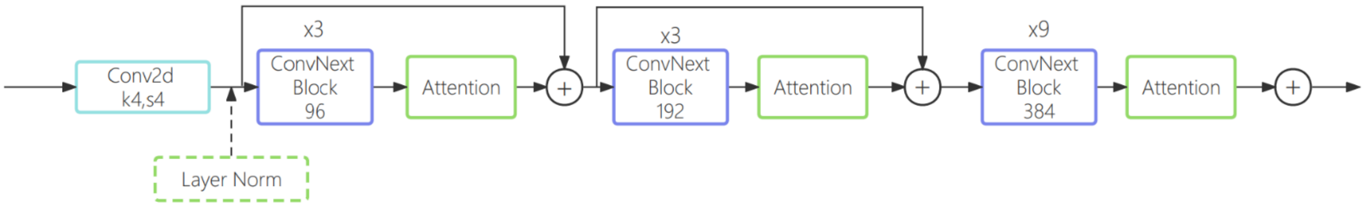

Fig. 6. Structure chart of SBA.

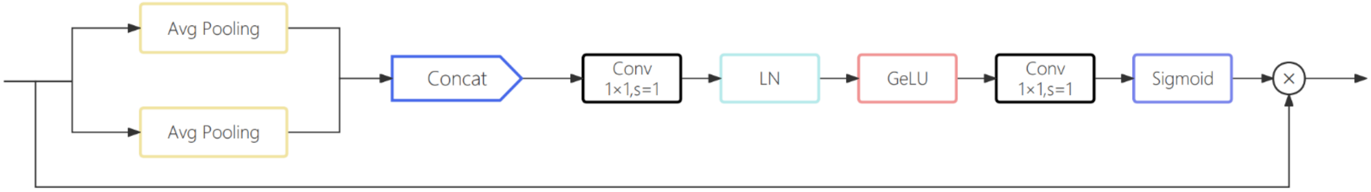

Fig. 7. Enhanced CAM structure.

Eq. (1) depicts the manner in which the network processes input data. The symbol F_{a1} denotes the map of feature generated by Enhanced CAM structure. Conv refers to the convolution operation and GeLU refers to Gaussian Error Linear Units, which are a type of activation functions. Once GeLU function applies, F becomes a tensor of shape $F \in \mathbb{R}^{(B,2C/r,W,H)}$. Upon the application of the function of sigmoid activation, F becomes a tensor of shape $F \in \mathbb{R}^{(B,C,W,H)}$. Avgpool is an abbreviation for average pooling, while Maxpool is an abbreviation for maximal pooling.

Subsequently, we incorporate bilinearity into the SAM. This work presents a new technique for extracting spatial attention, building upon existing methods. Firstly, we perform 1x1 and 3x3 convolution operations separately on the input feature maps. This allows model to obtain spatial features at different scales. Then apply layer normalization and the $GeLU$ activation function mapping to these two sets of features. Subsequently, individual channels are reduced in size using a convolution of 1 x 1. The two individual data channels are combined to generate spatial attention features at many scales. To create feature maps that combine spatial and channel weight features, features are multiplied by the feature map of channel attention weights. The data is then subjected to batch processing, and layer normalization is employed for two reasons. Firstly, we leverage the research findings of ConvNeXt, which utilizes layer normalization for batch processing in Transformer models. By applying layer normalization to the attention information in this paper, we align with the data processing method of ConvNeXt. Secondly, while batch normalization is widely used in CNNs and yields better results with larger batch sizes, Nevertheless, the hardware imposes a constraint on the batch size, which is independent of the layer normalization. When choosing the convolution kernel size, we opt for 1x1 and 3x3 convolutions. The 1x1 convolution not only allows us to obtain spatial features at different scales but also reduces the complexity and computational cost comprising the full network. Additionally, most of GPUs have implemented optimized algorithms for performing 3x3 convolutions, further improving efficiency. Based on the considerations mentioned above, we have optimized and modified the SAM structure accordingly. The improved SAM structure is illustrated in Fig. 8. The network structure's handling of input data is outlined in Eq. (2) to Eq. (4).

$$\boldsymbol{F_{a2}} = Sigmoid(\boldsymbol{F_1} \otimes \boldsymbol{F_2}) \tag{2}$$

$$\boldsymbol{F_1} = Conv_{1\times1}(GeLU(LN(Conv_{1\times1}(\boldsymbol{F})))) \tag{3}$$

$$\boldsymbol{F_2} = Conv_{1\times1}(GeLU(LN(Conv_{3\times3}(\boldsymbol{F})))) \tag{4}$$

F_{a2} refer to the map of feature generated by the structure of SAM. $Conv_{1\times1}$ refers to convolution using a 1 × 1 kernel. $Conv_{3\times3}$ refers to convolution using a 3 × 3 kernel. Once the $GeLU$ activation function applies, F has dimensions of $F \in \mathbb{R}^{(B,C/r,W,H)}$. Upon the application of the sigmoid activation function, F has dimensions of $F \in \mathbb{R}^{(B,1,W,H)}$.

In previous studies, attention mechanisms were primarily utilized to allocate weights solely between channel attentions, neglecting the consideration of spatial attention. This paper introduces a new and unique bilinear attention mechanism that consists of two separate branches. The first branch is tasked with supplying channel weights, and the secondary branch is dedicated to producing spatial weights. The initial CBAM process utilized a sequential processing approach. However, as we sought to enhance the algorithm's performance for fine-grained classification tasks, it became crucial for the network to accurately extract attention features. Consequently, we replaced the sequential processing with parallel processing to simultaneously train both channel and spatial attention. This approach ensures that multi-scale features of the target are obtained while preserving sufficient semantic information. For a detailed illustration of the specific attention network structure, please refer to Fig. 9.

The network indicated above demonstrates the processing of input data as depicted in Eq. (5):

$$F_{\text{out}} = (F_{a1} \otimes F_{\text{in}}) \otimes F_{a2} \oplus F_{\text{in}} \tag{5}$$

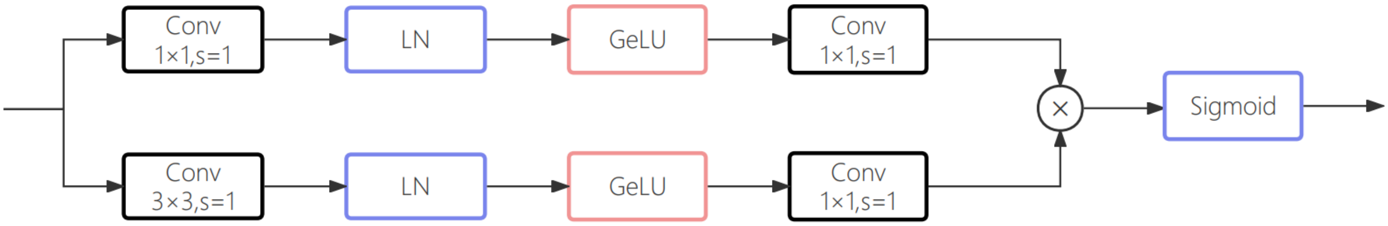

Fig. 8. Enhanced SAM structure.

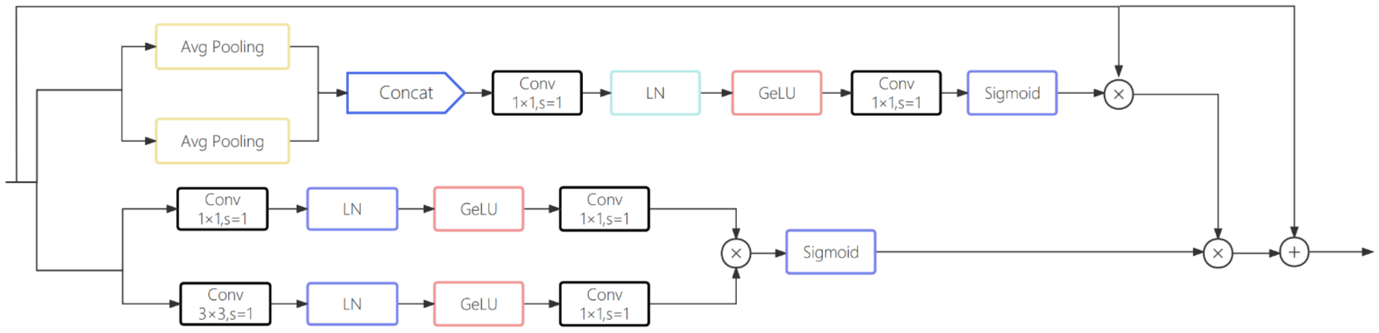

Fig. 9. Enhanced CAM structure.

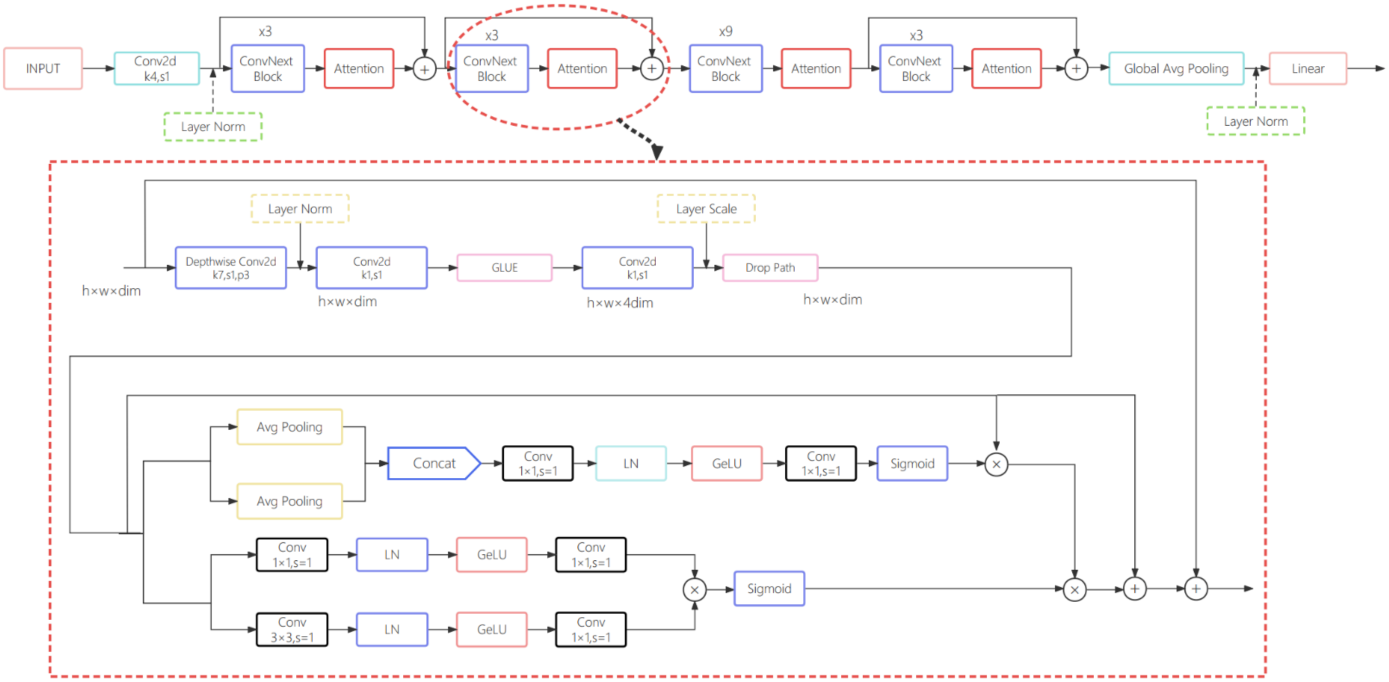

Fig. 10. ConvNeXt with BLCBAM model structure.

F_{a1} refer to the map of feature produced by CAM, F_{a2} refer to the map of feature produced by SAM, F_{in} refer to the input of BLCBAM map of feature, and F_{out} refer to the output of BLCBAM map of feature.

3) ConvNeXt Based on multiscale bilinear attention mechanism: Utilizing the proposed EBA, SBA, and multi-perspective attention framework, this paper presents a ConvNeXt model framework that incorporates bilinear attention. Through the training process, we not only extract multi-scale attention features from the feature maps iteratively but also effectively capture the fine-grained characteristics of the target.

Although ConvNeXt-Tiny already employs minimal parameters, the downsampling layers in its structure can still result in significant information loss after the conversion to images. Hence, each downsampling layer in the model is removed to preserve more information. This step enhances the information extraction ability of the embedded attention mechanism. Furthermore, considering that two-dimensional convolution can increase the dimensions of the data but may lead to information loss during convolution, the initial convolution stride in the relevant convolution modules of the original model is set to 1. This effectively strengthens the convolutional neural network's perception of the data.

Fig. 10 showcases how the proposed model structure seamlessly integrates the aforementioned components into the ConvNeXt network. As the network size and depth increase, the addition of attention modules theoretically enhances the accuracy of fine-grained classification.

IV. Implementation

In this section, the paper primarily introduces the experimental setup and implementation of the dataset, as well as presents the experimental results.

A. Dataset

To ascertain the viability of the model proposed in this paper, we conducted validation using meticulously labeled datasets obtained from the University of New Brunswick (UNB).

The datasets consist of the "ISCX VPN-nonVPN traffic dataset" (ISCX-VPN) and the "ISCX Tor-nonTor dataset" (ISCX-Tor). The primary data categorization of this dataset is displayed in Table I.

TABLE I. THE ORIGINAL DATA TYPES OF THE DATASET

Traffic Class	Application
Chat	ICQ,AIM,Skype,Facebook,Hangouts
Email	SMPT,POP3,IMAP
File transfer	Skype,FTPS,SFTP
VOIP	Facebook,Skype,Hangouts,Voipbuster
P2P	Torrent
Streaming	Viemo,Youtube,Netfilx,Spotify
VPN-Chat	ICQ,AIM,Skype,Facebook,Hangouts
VPN-Email	SMPT,POP3,IMAP
VPN-File	Transfer Skype,FTPS,SFTP
VPN-VOIP	Facebook,Skype,Hangouts,Voipbuster
VPN-P2P	Bittorrent
VPN-Streaming	Viemo,Youtube,Netfilx,Spotify

B. Data Classification Setting

After data processing, a comprehensive dataset containing all the classes was obtained for analysis in this paper. However, careful observation revealed an inherent imbalance within the dataset, which could potentially impede the overall classification performance. To address this concern, we undertook the task of reclassifying the data labels and subsequently created a balanced dataset. This was achieved by implementing the random undersampling method, which aims to maintain an equitable distribution of initial samples across each category.

The random undersampling technique functions by reducing the number of samples in the majority class, thereby rectifying the imbalance. Specifically, for the samples belonging to the majority class, random undersampling randomly selects a subset of them, ensuring that the sample size of the majority class approximates or equals that of the minority class. By adopting this approach, the model is prevented from exhibitingan excessive bias towards the majority class, consequently improving the overall classification performance. The utilization of random undersampling as a means to address the imbalanced dataset is an effective strategy that has been widely employed in various studies. Its application in this paper serves to mitigate the potential pitfalls associated with imbalanced data, ultimately enhancing the model’s classification capabilities. The data classification after filtering is shown in Table II.

This paper encompasses the design of two distinct multi-class classification problems, each serving a specific purpose within the research framework:

Traffic category recognition: The primary objective of this task is to accurately identify and classify traffic based on three encryption techniques: nonVPN, VPN, and Tor. To facilitate this, the paper generated three multi-class datasets utilizing the processed data.

Application recognition: In order to assess the model’s generalization capabilities, a dedicated application dataset was devised. The model underwent training on this dataset using transfer learning techniques, followed by fine-tuning using a limited amount of data from other protocols. This particular method sought to assess the model's competence in identifying various applications.

As previously mentioned, meticulous efforts were made to create filtered and balanced datasets for each sub-problem. Subsequently, the data was divided into a 90% training set and a 10% test set ratio, ensuring a robust evaluation of the model’s performance.

C. Evaluation Metrics

In this paper, the evaluation of model performance is based on two key metrics: accuracy and recall. Accuracy is defined as the ratio of correctly predicted samples to the total number of predicted samples. It provides an overall measure of the model’s correctness in predicting the class labels. On the other hand, recall assesses the model’s ability to accurately identify positive samples. It quantifies the proportion of true positive samples correctly identified by the model. The formal definitions are as follows:

$$\text{Accuracy} = \frac{\sum_{i\in \text{classes}} TP_i}{\sum_{i\in \text{classes}} (TP_i+FP_i)} \tag{6}$$

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

These evaluation metrics provide valuable insights into the model’s overall accuracy and its ability to correctly identify positive samples. They are crucial in assessing the model’s performance across different classification tasks.

TABLE II. THE FILTERING DATA TYPES OF THE DATASET

	VPN	Non-VPN	Non-Tor
Chat	AIM, Facebook, Hangouts,ICQ, Skype	AIM, Facebook, Hangouts,ICQ, Skype	-
Email	Email	Email	-
File Transfer	FTP,SFTP,Skype	FTP, SFTP, SCP	FTP,STP,POP,Skype
Streaming	Vimeo, Youtube	Hangouts,Netflix,Skype,Spotify,Vimeo,Youtube	Spotify,Vimeo, Youtube,Flash,Youtube,HTML5
VoIP	Facebook,Hangouts,Skype,Voipbuster	Facebook,Hangouts,Skype,Voipbuster	Facebook,HangoutsSkype
P2P	-	-	p2p_multipleSpeed, p2p_vuze
Browsing	-	-	Firefox, Chrome

TABLE III. THE EXPERIMENTAL OUTCOMES OF DEEP LEARNING TECHNIQUES FOR TRAFFIC IDENTIFICATION

Paper	Dataset	Method	Classification target	Accuaracy	Recall
This paper	ISCXVPN2016-NonVPN	the proposed method	Normal application traffic (10-category)	98.62%	98.95%
	ISCXVPN2016-VPN			97.43%	98.21%
	ISCXTor2016-NonTor			98.33%	98.61%
[24]	ISCXVPN2016	CAE2/ CNN	VPN and Non-VPN traffic / Traffic characterization	93.34%/92.9%	92.77%/93.5%
[25]	ISCXVPN2016	SAM(Attention Method)	Application protocol/Normal application Classification	98.62%/98.7%	98.65%/99.1%
[26]	ISCXVPN2016	PERT(Transformer)	Normal application traffic	93.27%	93.22%

D. Experimental

In this section, we have made necessary adaptations to the original model to address the specific requirements posed by smaller datasets. The original base structure, designed for classification tasks on larger datasets, needed adjustments to accommodate the significantly smaller data provided in this paper. Modifications were made to the initial convolutional parameters, and a stacking times ratio of 1:1:3:1 was implemented to prevent overfitting and safeguard the integrity of experimental accuracy.

Furthermore, to augment the model's perceptual prowess and maximize the retention of vital image information, while simultaneously safeguarding essential details, we made adjustments to the convolutional parameters and downsampling layers, as explicated in the preceding sections.

The primary objective of these adaptations is to make the model more adept at handling smaller datasets while improving its ability to perceive and interpret images. By doing so, we aim to enhance the model’s overall classification performance and ensure its suitability for the task at hand.

This paper presents a performance comparison between the improved structure and the original structure, as shown in Table III. The improved structure exhibits significant improvements in both accuracy and recall on non-VPN and VPN data. Particularly, the improved structure shows a more pronounced improvement on encrypted data compared to non-encrypted data. The average improvement rate of the improved structure is 14.75%.

For the adjusted model structure, the BLCBAM attention module is incorporated and used for classifying both encrypted and non-encrypted traffic. The traffic data is processed and transformed into images, which are then fed into the model for classification. The specific experimental results can be seen in Table IV.

TABLE IV. PERFORMANCE IMPROVEMENT AFTER ADAPTATION

	Non-VPN		VPN	
	Ac(%)	Re(%)	Ac(%)	Re(%)
Original version	78.53	77.95	82.35	81.32
Improved version.	94.35	94.67	95.14	95.3
promotion	15.8	16.7	12.6	13.9

The results in Table IV demonstrate that this paper achieves a recognition accuracy of 98.62% for non-VPN traffic, 97.43% for encrypted VPN traffic, and 98.33% for Tor traffic on this dataset. Compared to previous studies, the improved model in this paper shows a 5% improvement relative to the model proposed by Draper-Gil et al. [24] that uses CAE and CNN. Furthermore, compared to the model proposed by He H Y et al. [26] that combines Transformer and transfer learning, the model in this paper exhibits more accurate performance.

TABLE V. ADAPTATION RESULTS FOR SINGLE APPLICATION RECOGNITION

Classification	Accuracy (%)			
Chat	**Training/Test**	**Non-VPN**	**VPN**	**Non-Tor**
	Non-VPN	98.6	82.9	-
	VPN	75.7	96.4	-
	Non-Tor	-	-	-
Email	**Training/Test**	**Non-VPN**	**VPN**	**Non-Tor**
	Non-VPN	96.2	89.2	-
	VPN	69.7	98.2	-
	Non-Tor	-	-	-
File Transfer	**Training/Test**	**Non-VPN**	**VPN**	**Non-Tor**
	Non-VPN	98.8	73.9	81.6
	VPN	60.1	96.9	67.3
	Non-Tor	79.2	65.8	97.6
Streaming	**Training/Test**	**Non-VPN**	**VPN**	**Non-Tor**
	Non-VPN	99.8	71.9	70.6
	VPN	72.1	96.8	54.5
	Non-Tor	83.1	92.8	94.8
VoIP	**Training/Test**	**Non-VPN**	**VPN**	**Non-Tor**
	Non-VPN	95.6	69.4	85.2
	VPN	67.8	98.1	80.4
	Non-Tor	89.1	83.8	93.3

The comparison of traffic classification using different algorithms has been presented in the Table VI. The C4.5 machine learning algorithm exhibited suboptimal performance in accurately identifying both VPN and non-VPN data streams, with a precision value below 85%. Our proposed strategy in the domain of deep learning has exhibited a substantial enhancement in accuracy. Compared to algorithms utilizing a single model, the approach presented in this paper achieved an approximately 5% enhancement in VPN accuracy and a substantial 12% improvement in Non-VPN accuracy. Regarding the composite model approach, the method proposed in this paper exhibits similar performance in the VPN domain, while achieving an approximately 10% improvement in the Non-VPN domain. It is evident that, in comparison to prior research, the proposed method in this paper has achieved noteworthy advancements in traffic classification.

In certain real-world scenarios, it is necessary to identify whether specific protocols or applications exist within a large volume of network traffic. This paper conducts experiments specifically for this situation and trains the model using different encryption techniques to identify specific traffic categories. Transfer learning is employed using a small amount of data for the target protocols to improve the model's transferability. The results of the experiment are displayed in the Table V.

TABLE VI. OVERALL RESULT ON VARIOUS ALGORITHMS

Method	VPN		Non-VPN	
	Accuracy(%)	Recall(%)	Accuracy(%)	Recall(%)
C4.5 [27]	78.2	81.3	84.3	79.3
ID CNN [17]	92	95.2	85.8	85.9
SAE+1DCNN [28]	97.8	96.3	86.7	88.8
This Paper	97.4	98.2	98.6	98.9

By taking the average of the results for each encryption technique and considering only the cases where the test set has the same traffic category and encryption technique as the training set, the average accuracy obtained in this paper is as follows: 97.8% for non-VPN traffic, 97.28% for VPN traffic, and 94.9% for Tor traffic. Considering the overall average accuracy based on the above values, the overall average accuracy obtained in this paper is 96.66%. This paper has achieved significant success in describing and identifying internet traffic categories transmitted through different encryption techniques.

V. CONCLUSION

This paper presents a novel approach for traffic classification, employing ConvNeXt and a bilinear attention mechanism, with the goal of automatically extracting and analyzing traffic features to achieve accurate traffic characterization and application classification. The proposed method entails the conversion of data traffic into image representations, followed by the utilization of the ConvNeXt framework model, coupled with the bilinear attention mechanism, to enhance the model's perception of image features and optimize classification accuracy. Through extensive experimentation on various meticulously designed datasets, this paper thoroughly examines the performance of the proposed method and conclusively demonstrates its remarkable efficacy in accurately classifying diverse internet traffic categories.

REFERENCES

[1] Cotton, Michelle, Lars Eggert, and Dr. Joseph D. Touch. "Request for comments: No. 6335 Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry." 2011.

[2] Moore, A. W., and K. Papagiannaki. "Toward the accurate identification of network applications." Passive and Active Network Measurement: 6th International Workshop, PAM 2005, Boston, MA, USA, March 31-April 1, 2005. Proceedings 6. Springer Berlin Heidelberg, 2005. 41-54.

[3] Moore, A. W., and D. Zuev. "Internet traffic classification using Bayesian analysis techniques." Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems. 2005. 50-60.

[4] Liu, Z., H. Mao, C. Y. Wu, et al. "A convnet for the 2020s." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022. 11976-11986.

[5] P. Khandait, N. Hubballi, B. Mazumdar, "Efficient keyword matching for deep packet inspection based network traffic classification," in: Proceedings of the 2020 International Conference on COMmunication Systems & NETworkS (COMSNETS), IEEE, 2020, pp. 567–570.

[6] X. Wang, J. Jiang, Y. Tang, B. Liu, X. Wang, "Strid2fa: scalable regular expression matching for deep packet inspection," in: Proceedings of the 2011 IEEE International Conference on Communications, ICC, 2011, pp. 1–5.

[7] S. Fernandes, R. Antonello, T. Lacerda, A. Santos, D. Sadok, "Slimming down deep packet inspection systems," in: Proceedings of the IEEE INFOCOM Workshops, 2009, pp. 1–6.

[8] K.L. Dias, M.A. Pongelupe, W.M. Caminhas, L. de Errico, "An innovative approach for real-time network traffic classification," Computers & Networks 158 (2019), 143–157.

[9] J. Cao, D. Wang, Z. Qu, H. Sun, B. Li, C.-L. Chen, "An improved network traffic classification model based on a support vector machine," Symmetry 12 (2) (2020), 301.

[10] Y. Wang, Y. Xiang, J. Zhang, S. Yu, "A novel semi-supervised approach for network traffic clustering," in: Proceedings of the 2011 5th International Conference on Network and System Security, 2011, pp. 169–175.

[11] Höchst J, Baumgärtner L, M. Hollick, B. Freisleben, "Unsupervised traffic flow classification using a neural autoencoder," in: Proceedings of the 2017 IEEE 42nd Conference on Local Computer Networks, LCN, 2017, pp. 523–526.

[12] T. Bakhshi, B. Ghita, "On internet traffic classification: a two-phased machine learning approach," Journal of Computer Networks and Communications 2016 (2016), 21.

[13] F. Noorbehbahani, S. Mansoori, "A new semi-supervised method for network traffic classification based on x-means clustering and label propagation," in: Proceedings of the 2018 8th International C

[14] Shapira, T., and Y. Shavitt. "Flowpic: Encrypted internet traffic classification is as easy as image recognition." IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2019. 680-687.

[15] Lan, J., X. Liu, B. Li, et al. "DarknetSec: A novel self-attentive deep learning method for darknet traffic classification and application identification." Computers & Security 116 (2022): 102663.

[16] D'Angelo, G., and F. Palmieri. "Network traffic classification using deep convolutional recurrent autoencoder neural networks for spatial-temporal features extraction." Journal of Network and Computer Applications 173 (2021): 102890.

[17] Wang, W., M. Zhu, J. Wang, et al. "End-to-end encrypted traffic classification with one-dimensional convolution neural networks." 2017 IEEE international conference on intelligence and security informatics (ISI). IEEE, 2017. 43-48.

[18] Wang, W., M. Zhu, X. Zeng, et al. "Malware traffic classification using convolutional neural network for representation learning." 2017 International conference on information networking (ICOIN). IEEE, 2017. 712-717.

[19] Maonan, W., Kangfeng, Z., Ning, X., et al. "Centime: A direct comprehensive traffic features extraction for encrypted traffic classification." 2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS). IEEE, 2021. 490-498.

[20] Barut, O., Luo, Y., Li, P., et al. "R1dit: Privacy-preserving malware traffic classification with attention-based neural networks." IEEE Transactions on Network and Service Management (2022).

[21] Liu, X., You, J., Wu, Y., et al. "Attention-based bidirectional GRU networks for efficient HTTPS traffic classification." Information Sciences 541 (2020): 297-315.

[22] Xiao, X., Xiao, W., Li, R., et al. "EBSNN: extended byte segment neural network for network traffic classification." IEEE Transactions on Dependable and Secure Computing 19.5 (2021): 3521-3538.

[23] Woo, S., Park, J., Lee, J. Y., et al. "CBAM: Convolutional block attention module." Proceedings of the European conference on computer vision (ECCV). 2018. 3-19.

[24] Draper-Gil, G., Lashkari, A. H., Mamun, M. S. I., et al. "Characterization of encrypted and VPN traffic using time-related." Proceedings of the 2nd international conference on information systems security and privacy (ICISSP). 2016. 407-414.

[25] Xie, G., Li, Q., Jiang, Y., et al. "SAM: Self-attention based deep learning method for online traffic classification." Proceedings of the Workshop on Network Meets AI & ML. 2020. 14-20.

[26] He, H. Y., Yang, Z. G., Chen, X. N. "PERT: Payload Encoding Representation from Transformer for Encrypted Traffic Classification." 2020 ITU Kaleidoscope: Industry-Driven Digital Transformation (ITU K). 2020. DOI:10.23919/ITUK50268.2020.9303204.

[27] Lashkari, A. H., Draper-Gil, G., Mamun, M. S. I., et al. "Characterization of Tor traffic using time-based features." ICISSp. 2017. 253-262.

[28] Lotfollahi, M., Jafari Siavoshani, M., Shirali Hossein Zade, R., et al. "Deep packet: a novel approach for encrypted traffic classification using deep learning." Soft Computing, 24, 1999–2012 (2020).