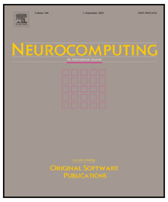
Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Deep learning and pre-training technology for encrypted traffic classification: A comprehensive review

Wenqi Dong [a,b], Jing Yu [a,b], Xinjie Lin [c,a,b], Gaopeng Gou [a,b], Gang Xiong [a,b,*]

[a] *Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China*
[b] *School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China*
[c] *Zhongguancun Laboratory, Beijing, China*

ARTICLE INFO

Keywords:
Encrypted traffic classification
Deep learning
Graph representing learning
Pre-training
Fine-tuning
Large language model

ABSTRACT

Network traffic classification has long been a pivotal topic in network security. In the past two decades, methods like port-based classification, deep packet inspection, and machine learning approaches have significantly progressed. Still, they are now facing reduced effectiveness due to the evolving complexity of the Internet, new encryption protocols, and advanced defense strategies. Given the problem that traditional models cannot efficiently generalize encrypted traffic, two promising technology paths are currently: deep learning and pre-training. On the one hand, deep learning-based methods effectively dissect complex network structures and unearth pivotal relational patterns. These approaches excel due to the neural networks' robust generalization capabilities, significantly boosting the accuracy and efficiency of recognition processes. Graph representation learning stands out as the most compelling contemporary model for such intricate analysis, adeptly revealing the critical relationships within network communication structures. We emphatically introduce mainstream deep learning-based methods, and the mechanism and scenarios are also analyzed. On the other hand, recognizing that although the analysis based on large models is the trend of the field, the application is truly limited now, we underscore the importance of pre-training, which aligns with the future trajectory towards the adoption of large-scale models in encrypted traffic analysis. The pre-trained model can overcome various defects of previous models and achieve more remarkable performance through its low labeled data dependency and strong scenario adaptability. We provide a comprehensive overview of existing pre-training-based approaches from the three stages of operation: pre-processing, pre-training, fine-tuning, and comparing representative relevant work. Finally, because of the current needs and the improvement space of the existing pre-training methods in the field, we synthetically analyze the challenges and opportunities for interested researchers to explore.

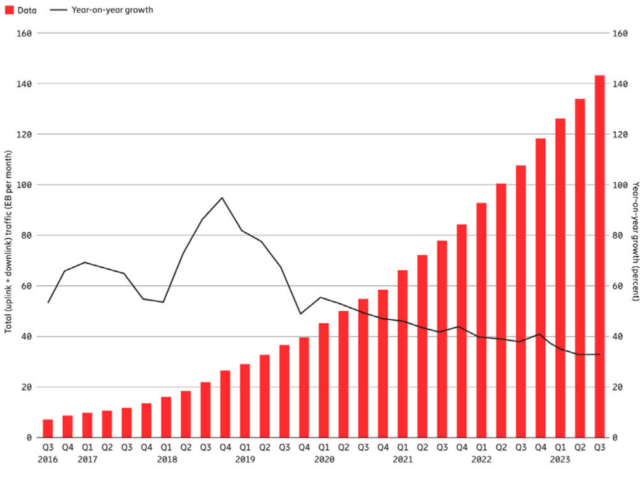
1. Introduction

With the rapid development and broad application of the Internet, the continuous growth of network traffic scale has become an inevitable trend. As shown in Fig. 1, according to the latest survey results of mobile equipment supplier Ericsson, from the third quarter of 2022 to the third quarter of 2023, mobile network data traffic increased by 33%. Since the first quarter of 2021, global mobile network data traffic has experienced a meteoric rise, doubling in just two years. By the third quarter of 2023, the total monthly global mobile network data traffic soared to 143 exabytes (EB) [1]. The driving forces of this trend include global digital transformation, the rapid expansion of Internet of Things (IoT) devices, and the popularity of online media. People increasingly rely on the Internet in their daily lives, and the Internet is not only used for information transmission and social interaction but also plays a major role in significant fields such as financial transactions, industrial control, and medical and health monitoring. Behind this explosive growth in network traffic is the need for more efficient network infrastructure to support this demand while also giving rise to higher requirements for data transmission security [2].

With the increasing importance of security and privacy protection for network traffic, encrypted communication has become the most common security means in the contemporary Internet. Many applications and services have adopted encryption technology to protect user privacy and data security, especially with the continuous evolution of encryption protocols such as TLS and DNS encryption and the promotion of the Quick UDP Internet Connection (QUIC) protocol. The overall

[*] Corresponding author at: Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China.
E-mail addresses: dongwenqi@iie.ac.cn (W. Dong), yujing02@iie.ac.cn (J. Yu), linxj@zgclab.edu.cn (X. Lin), gougaopeng@iie.ac.cn (G. Gou), xionggang@iie.ac.cn (G. Xiong).

https://doi.org/10.1016/j.neucom.2024.128444
Received 30 January 2024; Received in revised form 16 June 2024; Accepted 19 August 2024
Available online 22 August 2024



Source: Ericsson traffic measurements (Q3 2023).

Fig. 1. Global mobile network data traffic scale,2016–2023.

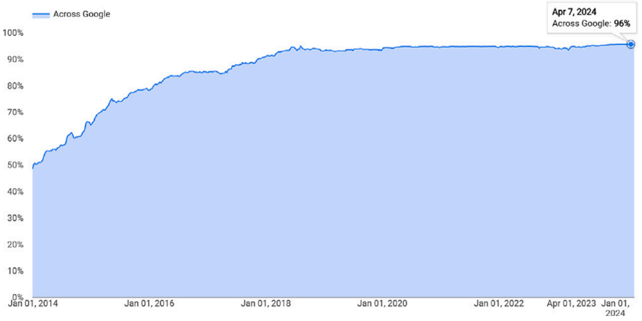


Fig. 2. Share of encrypted traffic in Google products,2014–2024.

popularity of encryption applications and the complete encryption of network communication traffic have become an irresistible trend. As of April 2024, the proportion of encrypted traffic in Google products has reached 96% and is still growing [3], as shown in Fig. 2. However, while encrypted communications utilize encryption to protect users from malicious surveillance and attacks, for network administrators and security experts, the popularity of encrypted traffic also brings challenges, especially regarding traffic classification and threat detection. In this case, traditional traffic classification methods, including port-based and deep packet inspection (DPI), are no longer applicable.

The origin of using machine learning and deep learning for classification dates back to the need for more advanced analysis techniques as encryption technologies evolved. Deep learning, particularly effective due to its ability to learn complex patterns and features, plays a crucial role in identifying encrypted traffic. Its primary advantage lies in its ability to automatically and efficiently learn intricate patterns from large datasets, essential in differentiating various types of encrypted traffic. Among deep learning methods, those based on graph representation learning stand out. These techniques treat network traffic as a graph, capturing complex interactions and dependencies between elements. By representing traffic data in this manner, graph-based methods can effectively uncover underlying structures and behavioral patterns in encrypted traffic, which might be challenging to detect using traditional methods. However, in the face of more intricate scenarios, larger datasets, more complex new encryption protocols, Out-of-Distribution (OOD) traffic classification, zero-day application identification, and other problems, all kinds of defects of these methods have been exposed, such as the need for large-scale labeled data, difficulty in feature extraction, easy underfitting or overfitting, low confidence of classification results, limited application scenarios, etc.

In 2017, Google researchers proposed Transformer [4], a deep learning framework based on a self-attention mechanism, which has significantly impacted natural language processing. Subsequently, BERT [5], GPT [6–8], XLNet [9], and other language models based on extensive corpus pre-training technology came out one after another. By successfully integrating a self-supervised learning mechanism and Transformer component, these models have established the mainstream position of pre-training and fine-tuning two-stage models, and pre-training technology has been rapidly developed [10]. The core concept of pre-training is to pre-train the model through large-scale data to learn the general representation containing the data context information. Then, for specific downstream tasks, fine-tuning can be conducted with only a small amount of labeled data. Through pre-training techniques, models can acquire more generalized data representations, subsequently utilizing this generalized representation information to initialize downstream tasks. The pre-training technique provides better performance and helps speed up the training process for downstream tasks. This milestone breakthrough makes the pre-training model shine in computer vision, natural language processing, speech processing, etc. Until the recent explosion of generative large models like ChatGPT, based on InstructGPT [11], pre-trained models have already proved to the industry that they are the future. These models represent a significant trend in the domain of encrypted traffic analysis, signaling a shift towards more sophisticated, AI-driven approaches.

In this article, we have focused on deep learning and pre-training because they are inextricably linked. Deep learning is a key component of pre-training techniques and large models. According to [12,13], with deep learning, we can build multi-layer neural networks with strong generalization ability that automatically extract and transform features from large amounts of data end-to-end. The pre-training technology has further expanded on these ideas, enhancing the universality and ease of use of deep learning models, including their generalizability and transferability. Meanwhile, although the current application of such large models in this area remains quite limited, the use of pre-trained models, serving as a foundational technological approach, holds monumental significance. They not only pave the way for the adoption of larger models in encrypted traffic analysis but also highlight the potential for groundbreaking advancements in this field. This paradigm shift underscores the importance of continued research and development to fully harness the capabilities of large models in practical, real-world scenarios of encrypted traffic management and security.

In encrypted traffic classification, some studies based on the pre-training technology have shown its high efficiency because of the low dependency on large amounts of labeled data, strong generalization capacity, and extensive range of downstream scenarios. However, although a certain degree of progress has been made, the relevant applications are still limited and need to improve. In addition, there needs to be a more systematic summary and overview of the current state of pre-training techniques. Therefore, the contribution of this paper includes the following three aspects:

1. We classify the existing encrypted traffic classification technologies according to their structural features and summarize them according to their usage characteristics, model architecture, and task scenarios. The characteristics of plaintext rule-based and machine learning methods are also discussed.
2. We showcase the effectiveness of deep learning – particularly graph-based methods – and pre-trained models in encrypted traffic. We also highlight the distinct advantages that pre-trained models offer for classifying encrypted traffic. We are the first to survey existing studies that employ pre-training techniques in encrypted traffic classification and identification.
3. We discuss the problems and shortcomings of existing classification models on encrypted traffic, and point out the future research direction of pre-trained models in this field.

The remainder of this paper is organized as follows. Section 2 provides an overview of the encrypted traffic classification. Section 3 offers an exploration of traditional methods used for classifying encrypted traffic. A comprehensive overview of the deep learning methods is presented in Section 4. The theory and application of the classification based on pre-training technology are thoroughly outlined in Section 5. Then, the deficiencies in the existing work and the opportunities for future traffic classification based on pre-training are summarized in Section 6, while the conclusion is recapitulated in Section 7.

2. Overview of encrypted traffic classification

In the technical field of encrypted traffic classification, various scenarios have different classification requirements, and there are apparent divergences between the input and output forms and evaluation metrics involved. Moreover, different research work also has a distinct emphasis on introducing encrypted traffic technology, so the classification framework of its definition is also very different.

In this section, the definition of the encrypted traffic problem is briefly introduced. In addition, the dimensions and evaluation metrics of encrypted traffic classification are provided. Finally, various classification scenarios and comparing encrypted traffic classification surveys are presented.

2.1. Problem description

Encrypted traffic refers to the actual encrypted plaintext content transmitted during network communication. Specifically, in the process of network communication, encrypted traffic usually refers to the encryption of data through encryption protocols to ensure that only authorized receivers can decrypt and read data. Encrypted traffic usually has the characteristics of high entropy, unobvious statistical characteristics, and weak correlation between adjacent bytes [14]. This encryption can be applied to various scenarios, including Internet browsing, email, instant messaging, and other online data transmission scenarios, to improve data security and privacy protection.

Encrypted traffic classification refers to the construction of a classification model using a specific algorithm, and the classification model is used to classify and identify the encrypted traffic of various applications, devices, services, and so on. The encrypted traffic classification framework is shown in Fig. 3[1], which can be summarized as follows: Initially, researchers, addressing their security analysis and network management demands, pinpoint specific traffic classification scenarios like mobile encrypted application classification, encrypted malware detection, and anonymity encrypted traffic identification. They then capture and pre-process particular types of traffic within these scenarios to extract the direct content/side channel information for model input. Utilizing feature engineering, the features better aligning with the scenario's communication mode are chosen, followed by employing various models, including methods based on machine learning/deep learning or pre-training. Post the training and verification phases, the result of classification and identification is a specific application or application-layer protocol or a particular service type divided according to quality of service (QoS) requirements [15], such as downloading, browsing, calling [16], etc. Lastly, these results are analyzed and verified, culminating in the completion of a traffic classification and identification task.

In short, its primary task is to determine the type of input and output according to the actual needs, then choose the appropriate classification method according to the scene needs and metrics for evaluation. According to the evolution of the technical route, the classification methods of encrypted traffic can be mainly divided into four categories: the classification method based on plaintext rules, the classification method based on machine learning model, the classification method based on deep learning model, and the method based on pre-trained model. The technical types of encrypted traffic classification is shown in Table 1.

2.2. Public datasets

For different application scenarios, existing studies have introduced numerous traffic datasets. Some common datasets, along with their types and characteristics, are presented in Table 2, and the detailed introduction is described below.

For the ETCV scenario, one of the most common datasets is ISCX VPN-nonVPN [65]. This dataset contains 7 categories of network services, each of which has VPN and non-VPN traffic, so it contains a total of 14 labels of data, including VOIP, VPN-voip, P2P, VPN-P2P, and so on. The total amount of data is 28 GB. The VPN part uses an external VPN service provider and connects to it using OpenVPN (UDP mode).

For Tor traffic classification, the ISCX Tor-nonTor [66] dataset is classified similarly as ISCX VPN-nonVPN: a total of 8 categories are defined, which are divided into Tor traffic and non-Tor traffic, namely Browsing, Email, Chat, Audio-streaming, Video-streaming, File Transfer, VoIP, and P2P. The authors build this dataset by capturing workstation and gateway outgoing traffic, collecting a pair of .pcap files: regular traffic pcap (workstation) and Tor traffic pcap (gateway) files. Unlike the former, dataset DF focuses on WF, which analyzes patterns in traffic to identify web pages under Tor encryption [30]. The dataset is divided into closed-world and open-world sections, which are further divided into non-defended part and defended parts by WalkieTalkie and WTFPAD, respectively.

In the field of malicious traffic classification, USTC-TFC2016 [67], CIC-IDS 2017 [68], and IoT-23 [69] are commonly used datasets. Among them, the former is a collection of traffic composed of malicious software and benign applications, including 10 categories of malicious software, such as Cridex, Zeus, etc., and 10 categories of normal traffic. The length of each flow in this dataset is large, and the plaintext traffic accounts for a considerable proportion. For intrusion detection, CIC-IDS2017 was collected from July 3 to July 7, 2017, spanning five days. Benign traffic was captured on Monday, while attacks such as Brute Force FTP and DDoS were executed on Tuesday through Friday, both morning and afternoon. The latter's goal is to offer a large dataset of real and labeled IoT malware infections and IoT benign traffic, and it is composed of 20 malware captures executed in IoT devices and 3 captures for benign IoT device traffic.

Overall, the lack of disclosure regarding the collection details and data composition of self-collected or private datasets undermines research credibility. The introduction of public datasets significantly boosts the reproducibility of studies and allows researchers to compare different encrypted traffic classification methods readily.

2.3. Classification dimension

The dimension of encrypted traffic classification refers to the input and output forms of different levels used in the classification process. The input form can be classified into direct content information and side-channel information. The output form can be classified into

[1] The original meaning of the scenarios abbreviation in the subsequent tables is as follows:

ETCV – Encrypted **T**raffic **C**lassification on **V**PN
MEAC – **M**obile **E**ncrypted **A**pplication **C**lassification
WF – **W**ebsite **F**ingerprinting
GEAC – **G**eneral **E**ncrypted **A**pplication **C**lassification
EMC – **E**ncrypted **M**alware **C**lassification
EACT – **E**ncrypted **A**pplication **C**lassification on **T**or
EAC-TLS 1.3 – **E**ncrypted **A**pplication **C**lassification on **TLS 1.3**
EAC-QUIC – **E**ncrypted **A**pplication **C**lassification on **QUIC**
MATD – **M**alicious **A**ttack **T**raffic **D**etection
IoTDTC – **IoT D**evice **T**raffic **C**lassification
EDAC – **E**ncrypted **D**ecentralized **A**pplications **C**lassification

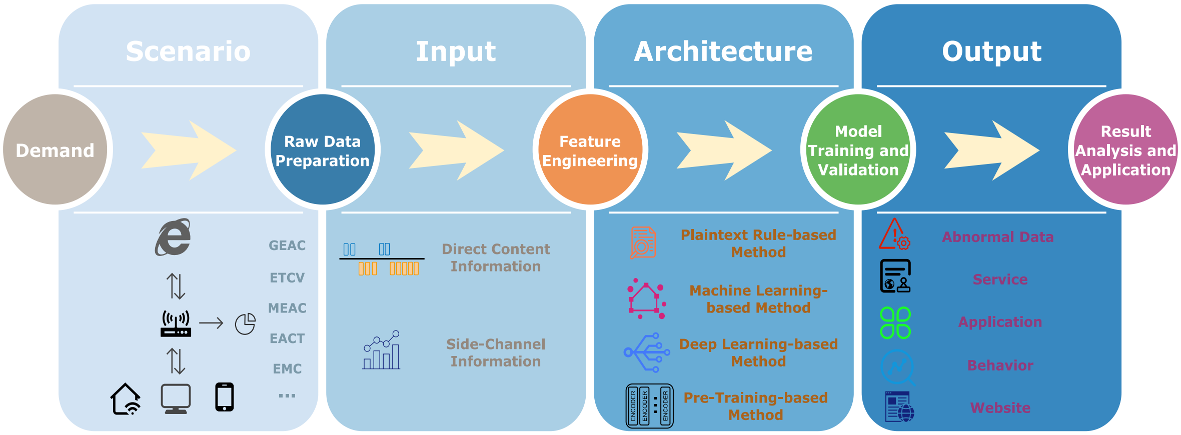

Scenario
Input
Architecture
Output
Demand
Raw Data Preparation
Feature Engineering
Model Training and Validation
Result Analysis and Application
GEAC
ETCV
MEAC
EACT
EMC
...
Direct Content Information
Side-Channel Information
Plaintext Rule-based Method
Machine Learning-based Method
Deep Learning-based Method
Pre-Training-based Method
ENCODER
Abnormal Data
Service
Application
Behavior
Website

Fig. 3. Encrypted traffic classification framework.

Table 1
Encrypted traffic classification technology framework.

Class	Methods		Characteristics
Plaintext Rule-based Method	Port-based Method		Invalid for Encryption
	Deep Packet Inspection		Challenges with Privacy
Machine Learning-based Method	Unsupervised Learning	[17,18]	Strong Interpretability
	Semi-supervised Learning	[19,20]	Dependency on Expert Knowledge
	Supervised Learning	[21–26]	Easy Feature Degeneracy
Deep Learning-based Method	Unsupervised Learning	[27,28]	Strong Generalization Ability
	Supervised Learning	[29–41]	Large, Labeled, and Evenly Distributed Datasets Needed
	Graph-based	[42–48]	Prone to Overfitting
Pre-training-based Method	Transformer Encoder-based	[49–57]	Strong Generalizability for Multi-scenarios
	Transformer Decoder-based	[58–60]	Low Large-scale Labeled Data Dependency
	Transformer-based	[61,62]	No Complex Feature Engineering
	Classic DL-based	[63,64]	Large Computing Resources Required

protocols, services, applications, behaviors, anomaly traffic detection, etc. Correctly distinguishing the classification dimensions of encrypted traffic can help network administrators and security experts analyze and manage network traffic more effectively. Different classification dimensions provide different perspectives to help identify potential security risks, monitor network performance, and diagnose problems.

2.3.1. *Input format*

The input form of encrypted traffic classification refers to the type and structure of data the model receives for the prediction and analysis of classification tasks. Common input forms are listed below.

1. Direct content information

In encrypted traffic classification, the direct content information primarily includes two components: the payload and the header of traffic packets, where the header contains the specific information transmitted in the encrypted traffic communication and the protocol information to be complied with.

The header is the structured information attached to the payload to describe and manage the transmission of that packet across the communication nodes. Together with the payload, it forms a complete traffic packet. The early port-based classification method involves identifying traffic types by the ports they use, which is a fundamental technique in network management. Generally speaking, most current classification methods using headers as inputs retain IP and TCP/UDP headers as inputs to reduce the confusion of irrelevant content, including the Ethernet layer, but relevant work [31] has proved that using IP addresses as input can easily lead to overfitting of models because some applications use the identical content delivery networks (CDNs) or apply the same third-party libraries. Some deep learning-based work uses raw packet bytes, including the header as input [31], or just uses the header for classification [73].

The payload is the actual data transmitted over the network. It is the core communication component, from simple text messages to complex multimedia content. Identification techniques like Deep Packet Inspection (DPI) delve into plaintext payload content for detailed insights, crucial for tasks like malware identification and content filtering. Meanwhile, deep learning approaches have been increasingly employed to analyze encrypted payloads, learning intricate patterns for sophisticated tasks like anomaly detection and content categorization. Specifically, these methods adopt the idea of presentation learning to avoid cumbersome rule construction, automatically extract essential information from encrypted original information, and generate differentiated encrypted traffic fingerprints [29]. Additionally, pre-trained learning models are utilized to decode and understand complex payload structures, thereby enhancing the efficiency and accuracy of network traffic analysis. With the increasing model parameters and data scales, payload-based classification methods will become more and more critical in the future.

2. Side-channel information

Unlike the direct content information, which includes payload and header, side-channel information is derived indirectly from observing traffic patterns and characteristics, mainly statistical and time-series features. This type of information is crucial for analyzing encrypted traffic where direct inspection of plaintext content is not feasible. Even if the specific transmission content cannot be directly obtained,

Table 2
Summary of public encrypted traffic datasets.

Name	Year	Scope	Description
ISCX VPN-nonVPN [65]	2016	ETCV	By capturing one regular session and one session through VPN, the dataset contains 14 traffic categories
ISCX Tor-nonTor [66]	2016	EACT	The dataset includes both regular traffic and Tor traffic, with Tor traffic comprising 7 categories
USTC-TFC2016 [67]	2016	EMC	The dataset consists of 10 types of benign traffic and 10 types of malicious software traffic, and the amount of data is unbalanced among different types
CIC-IDS-2017 [68]	2017	MATD	Over a 5-day period, 50 GB of normal and eight types of malicious attack traffic, including Brute Force FTP, were collected
Cross Platform [70]	2018	GEAC	Collected from the top 100 apps in the US, China and India for both platforms (iOS and Android), the dataset contains 196 and 215 apps, respectively
DF [30]	2018	WF	The datasets of web traffic traces produced for the closed-world evaluations on non-defended, WTF-PAD and Walkie-Talkie datasets
IoT-23 [69]	2020	IoTDTC	IoT-23 is the dataset of network traffic from Internet of Things (IoT) devices. It has 20 malware captures executed in IoT devices, and 3 captures for benign IoT devices traffic
DApp-60 [71]	2021	EDAC	The dataset contains encrypted traffic for 60 of the most used Ethereum decentralized applications, containing nearly 290,000 encrypted network flows
CSTNET-TLS 1.3 [50]	2022	EAC-TLS 1.3	A collection of 120 applications under CSTNET, taken from Alexa Top-5000 with TLS 1.3 deployed
CIC-MalMem-2022 [72]	2022	EMC	The dataset, made up of malware such as spyware, ransomware and Trojans, has 58,596 samples, with a 50/50 split between benign traffic and malware traffic
FGNet53 [41]	2023	MEAC	The dataset contains two subsets containing the same 53 apps: D1 and D2. It can be used for the experiment of concept drift adaptation of version update

studying these side-channel details makes it possible to infer network behavior and detect anomalies even in environments where data encryption is prevalent.

Time-series features refer to characteristics that are extracted over time. These features encapsulate the temporal aspects of data [74], including packet arrival intervals, sequence of packet sizes, and the timing of traffic flow. They are crucial for understanding the dynamics of network behavior and identifying patterns over time. They are particularly useful in applications like anomaly detection, network performance analysis, and predicting future network states. By capturing the temporal dependencies in traffic data, time-series features provide a deep insight into the evolution of network activities [48].

Statistical features involve the aggregation and summary of traffic data into statistical metrics. These features include metrics like average packet size, total number of packets, variance of packet sizes, and distribution of flow durations. For instance, Van et al. [20] proposed a method based on machine learning using statistical characteristics, including certificates in the payload of TLS 1.2 handshake packets as model input for fingerprint construction. They provide a comprehensive overview of the traffic characteristics, allowing for identifying trends and patterns in network behavior. Statistical features are precious in scenarios requiring a macro-level understanding of the traffic, such as network capacity planning, quality of service measurement, and general traffic profiling. They provide a solution when direct access to a packet's payload is impossible [75]. However, statistical characteristics are imperfect; confusing traffic and malicious traffic can change specific statistical characteristics through field filling or camouflage. Moreover, many work with statistical features that require processing of the entire stream, which means real-time detection is difficult.

2.3.2. Output format

The output form of encrypted traffic classification refers to the expression of model classification results, which is used to reflect different aspects and characteristics of encrypted traffic. Common output forms include:

1. Encryption protocol

Identify the transport protocols used by encrypted traffic, such as SSH, IPSec, SSL/TLS, QUIC, etc. Through protocol classification of encrypted traffic, traffic of different protocols can be identified and then processed and analyzed according to their characteristics.

2. Service type

Indicates the type of network service to which the encrypted traffic belongs, such as browsing, file transfer, remote login, and so on. By classifying encrypted traffic into service categories, researchers can understand the type of service being performed on the network.

3. Applications

Identifies the specific applications to which the encrypted traffic is attributed, including but not limited to web browsers, email clients, and video streaming media. Classifying the encrypted traffic based on its corresponding application enables a more detailed segmentation and analysis of the usage patterns and behaviors associated with different applications.

4. Behavior type

Pinpoints specific behaviors or patterns contained within encrypted traffic, such as sending voice messages, interacting in WeChat Moments, or making video calls within WeChat [25]. By classifying behaviors in encrypted traffic, a more granular analysis can be conducted on the patterns of specific applications to which the encrypted traffic belongs.

5. Website category

Classifies the websites to which the encrypted traffic belongs. Part of the work [44] realizes fine-grained identification of various pages under the same website through package length sequence information, such as Yahoo's news, finance, sports, and shopping pages.

6. Malicious traffic detection

Indicates abnormal or malicious traffic in encrypted traffic, such as traffic surges, abnormal flows, DDoS attacks, APT attacks, port scanning, FTP-Patator, Bot [67], etc. By detecting and classifying encrypted traffic, researchers can promptly identify and handle network anomalies and malicious traffic, improving network security and stability.

Table 3
Confusion matrix of true and predicted values.

Confusion matrix		True value	
		P	N
Prediction Value	P'	TP	FP
	N'	FN	TN

7. Content parameter identification

Identifies the application traffic further from the content parameters, such as video format, picture resolution, and other transmission content-specific parameters [76].

In summary, the dimension of encrypted traffic classification depends on the input form and output form chosen. The refined classification and analysis of encrypted traffic can be realized through reasonable selection and combination of these forms to meet the requirements of different application scenarios.

2.4. Evaluation metrics

The current Internet environment is vast and complex, and different scenarios require different goals for encrypted traffic classification. Therefore, reasonable metrics are needed to objectively and comprehensively evaluate the advantages and disadvantages of encrypted traffic classification technology. Common classification metrics can be summarized as follows.

1. Confusion matrix

The confusion matrix evaluates classification accuracy in encrypted traffic classification by comparing predicted versus actual labels (Table 3). For class i, TP (True Positives) and TN (True Negatives) count samples correctly identified as class i and not class i, respectively. FP (False Positives) and FN (False Negatives) count misclassifications as class i and misidentifications of class i, respectively. Summations $TP + FN = P$ and $FP + TN = N$ represent the total actual samples for class i and other classes, with $TP + FP = P'$ and $FN + TN = N'$ tallying the predicted counts.

2. Accuracy

Accuracy measures how well a model classifies correctly, calculated as the ratio of correctly predicted samples to the total samples. Accuracy is defined as

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

However, accuracy can be misleading if the dataset is imbalanced, with one class significantly larger than others, possibly biasing the model towards that class. Thus, it is important to also consider metrics like precision, recall, and F1-score for a comprehensive evaluation of a model's performance.

3. Precision

Precision measures how many of the model's positive predictions were correct, defined as

$$precision = \frac{TP}{TP + FP}$$

High precision indicates few false positives.

4. Recall

Also known as TPR, recall is the proportion of actual positives correctly identified by the model, defined as

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

It assesses the model's ability to detect all positive cases.

5. FPR

FPR quantifies the rate at which the model misclassifies negative cases as positive, defined as

$$FPR = \frac{FP}{TN + FP}$$

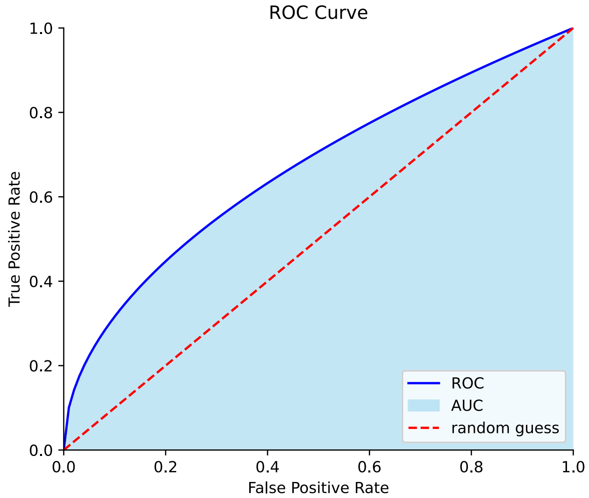


Fig. 4. ROC Curve.

6. Specificity

Specificity measures the model's accuracy in identifying negative cases, with its sum with FPR being 1.

$$Spec = \frac{TN}{TN + FP}$$

7. F1-score

The F1-score is the harmonic mean of precision and recall, useful in balancing the two, especially in unbalanced datasets.

$$F - measure = \frac{2 \times precision \times recall}{precision + recall}$$

8. AUC-ROC

The ROC curve, essential in scenarios like malicious traffic identification, plots TPR against FPR (Fig. 4). AUC, or the area under the ROC curve, quantifies the classifier's effectiveness; a larger AUC indicates better detection performance.

9. Stability

Stability measures a model's ability to consistently perform well over time, unaffected by changes in data over different periods. It is calculated as

$$stability = \sqrt{\frac{\sum_{t_i=1}^{T} (acc_{t_i} - acc_0)^2}{T}}$$

where acc_0 is initial accuracy and acc_{t_i} is accuracy at time t_i, reflecting the model's resilience to data changes over time.

10. Real-time

This metric assesses a model's capability to classify traffic online. Effective real-time models are typically compact, easy to implement, and have lower computational complexity $O(n)$ and storage complexity $T(n)$. However, the need for real-time processing may compromise accuracy and compatibility, so it is crucial to evaluate real-time capabilities based on specific needs.

2.5. Related survey

Since network traffic analysis and classification involve all aspects of various industries, the existing related surveys also start from different points of view, as shown in Table 4.

In the paper, as in [77,78], the field of encrypted traffic classification and analysis is summarized from a macro perspective. The former summarizes the categories, requirements, classification methodology, and challenges of early encrypted traffic classification; the latter focuses on the mechanism of network traffic encryption, including the common

Table 4
Comparisons of existing surveys of encrypted traffic classification.

Publication	Year	Description	Scope
Cao et al. [77]	2014	Summarizing the classification of encrypted traffic from a macro perspective	Plaintext rule, statistical feature and behavior-based method
Velan et al. [78]	2015	Overviewing from the mechanism of network traffic encryption	Payload, statistical feature and machine learning-based method
Rezaei et al. [12]	2019	Introducing a general framework of traffic classification based on deep learning and common deep learning models and their applications	Deep learning-based method
Wang et al. [13]	2019	Proposing a general framework for mobile encrypted traffic classification based on deep learning and model architecture design of this field	Deep learning-based method
Alqudah et al. [80]	2020	Introducing the application of machine learning in traffic analysis briefly	Machine learning-based method
Rodriguez et al. [81]	2021	Reviewing the research work on using deep learning methods to improve network security systems from three main areas: infrastructure, software and privacy	Deep learning-based method
Papadogiannaki et al. [82]	2021	Categorizing the encrypted traffic inspection work by use case, technique and objective	Payload, statistical feature, machine learning and deep learning-based method
Ankit et al. [79]	2022	Surveying the analysis architecture for mobile encrypted traffic, including scenarios, data processing techniques and related work	Plaintext rule, machine learning and deep learning-based method
Shen et al. [83]	2022	Surveying the mechanism and related work of four scenarios, including network asset identification, network characterization, privacy leakage detection, and anomaly detection	Knowledge, machine learning and deep learning-based method
Wang et al. [84]	2022	Proposing a general machine learning framework, and the existing machine learning classification models and datasets are compared	Machine learning and deep learning-based method
Dong et al.	2024	Summarizing the existing work on encrypted traffic classification including pre-training-based approaches from the technical view, and looking forward to the future research direction comprehensively	Plaintext rule-based, machine learning-based, deep learning-based and *pre-training-based* method

encryption protocols and the generation and propagation of encrypted traffic.

Several surveys [13,79] pay special attention to the encrypted traffic classification on mobile devices due to the proliferation of mobile devices and the widespread adoption of encryption technology in mobile services, including E-commerce, search engines, social networking, etc. Among them, Wang et al. [13] and Ankit et al. [79] both propose a classification framework to categorize the existing work in the field of mobile encrypted traffic classification. The difference is that the former focuses on the combination of deep learning technology and mobile encrypted traffic classification, and several common deep learning models are introduced; the latter focuses on the differences between encrypted traffic characteristics in mobile and non-mobile scenarios and the detailed data collection and tagging technology path for encrypted mobile traffic is introduced in this work.

Because machine learning and deep learning technology have been widely used in encrypted traffic classification, some surveys [12,80] have chosen to overview the existing work starting from a specific technical route. Alqudah et al. [80] give a brief overview and comparison among some existing machine learning approaches and products to combat cyber-attacks in traffic classification. Liu et al. [12] mainly introduce the whole process framework of encrypted traffic classification based on deep learning, including data collection, pre-processing, feature selection, model selection, training and validation, and periodic evaluation.

Due to the popularity and upgrade of encryption technology, malicious traffic encryption is proliferating indirectly, making the current cybersecurity situation extremely severe. Some work [81,84] emphasize the method of encrypted malicious traffic detection and identification. Rodríguez et al. [81] analyze related deep learning methodologies, covering three aspects of cybersecurity: intrusion detection, software attack detection, and privacy protection, as well as paying particular attention to the implementation, datasets, and results of the methods involved. Wang et al. [84] design a general framework for encrypted malicious traffic detection based on machine learning and combine datasets from 5 different sources to generate a comprehensive and fair dataset to aid future research in the field. On this basis, they also implement and compare 10 malicious traffic detection algorithms.

Because encrypted traffic classification and analysis is a whole-chain, multi-level, and wide-range technology, some researchers fully summarize the existing work from the perspective of their cognition. Papadogiannaki et al. [82] propose a taxonomy for encrypted network traffic inspection work categorized by use case, technique, and objective. Shen et al. [83] survey existing encrypted traffic classification work based on knowledge, machine learning, and deep learning technology under four scenarios in this field, including network asset identification, network characterization, privacy leakage detection, and anomaly detection.

In contrast, our work not only comprehensively summarizes existing encrypted traffic classification methods, including the newest pre-training-based approaches from a technical level, but also provides a comprehensive analysis and discussion of methods and detailed features based on pre-trained models.

3. Traditional encrypted traffic classification method

In recent years, with the continuous upgrading of encryption and obfuscation methods, encrypted traffic classification technology has gradually evolved, mainly divided into four classification methods: plaintext rules-based, machine learning-based, deep learning-based, and pre-training-based. Due to the earlier time proposal and wide practical application of the first two, it is also known as the traditional encryption traffic classification technology. In this section, we focus on these traditional encrypted traffic classification methods.

3.1. Plaintext rule-based classification method

Plaintext rule-based classification methods are mainly divided into port-based and DPI methods according to the specific implementation technology. In this section, we will look at both methods in detail.

Table 5
Common service-port mapping relationships.

Service	Port	Transport layer protocol	Summarize
FTP-DATA	20	TCP	File Transfer Protocol data connection
FTP	21	TCP	File Transfer Protocol control connection
SSH	22	UDP	Secure Shell Protocol
Telnet	23	TCP	Teletype Network Protocol
SMTP	25	TCP	Simple Mail Transfer Protocol
DNS	53	UDP	Domain Name System
HTTP	80	TCP	Hypertext Transfer Protocol
POP3	110	UDP	Post Office Protocol version 3
SNMP	161	TCP	Simple Network Management Protocol
SSL	443	TCP	Secure Sockets Layer Protocol

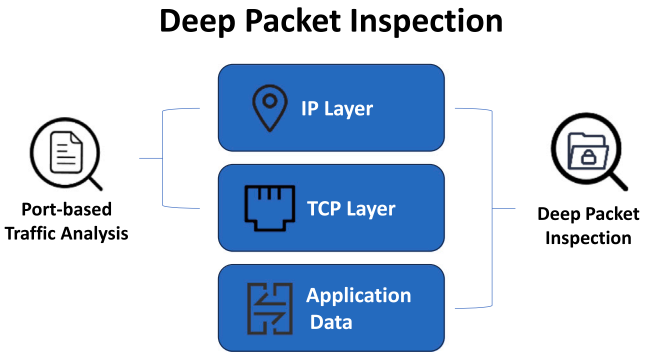


Fig. 5. The principle of deep packet inspection.

3.1.1. *Port-based classification method*

This method identifies the type of service or application by assuming that most applications use the default Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) port number. According to the port mapping table [85] assigned by the Internet Assigned Numbers Authority (IANA), the service type of network traffic packets is mapped to the corresponding port of its transport layer protocol (TCP or UDP) to implement classification. The main mapping combinations are shown in Table 5. In addition, for Peer-to-peer (P2P) traffic, although dynamic port numbers are widely used, the typical port number of P2P terminal software can still be mapped to realize P2P traffic classification.

The port-based method has achieved good results in early traffic classification scenarios. It distinguishes traffic packets by matching the port number of the transport layer header; to be specific, it only simply looks for the TCP handshake messages (SYN packets) exchanged between the sender and receiver during the connection establishment [79], resulting in low time complexity and making it suitable for high-speed real-time traffic classification; however, it also entails significant limitations. First, the dynamic ports, port hopping/obfuscation, and HTTP obfuscation technology are used, failing the original mapping combination. For example, Skype uses dynamic or commonly used ports such as 80 and 443 to get through firewalls or other restrictions [86]. Second, IANA currently does not provide the mapping rules for some new application layer service types to port numbers. Finally, the granularity based on port number classification is coarser. It can only distinguish the types of application layer services (protocols) but cannot satisfy the identification of finer-grained network behavior.

Due to the above limitations, this method's classification accuracy and reliability have been declining, and it has yet to meet the needs of current network traffic classification. In 2004, Sen et al. [87] experimented on the Kazaa P2P protocol, and the experimental results showed that the default P2P port number only accounted for about 30% of the total traffic tested. Moore and Papagiannak conducted similar experiments [88], Madhukar and Williamson [89], etc., by using the official IANA port number list for classification, but the accuracy was less than 70%. However, due to the low time complexity and simple implementation, there are still some demand scenarios for this method.

3.1.2. *Payload-based classification method*

The traditional packet detection methods represented by port-based classification only pay attention to the packet header information [90], which mainly includes five elements: source IP address, destination IP address, source port number, destination port number, and protocol type. Early firewalls often used such methods for intrusion/anomaly detection. They were often time efficient but needed to capture the actual content of the network communication, so it was impossible to make further judgments about it. Payload is the rest of the traffic packet except the header. Because in the network communication process, the actual transmitted application layer information, such as user behavior and application content, is saved in this part, researchers have developed a classification method based on the entire packet, including the header and payload, to realize network traffic identification. This method is also known as payload-based classification called Deep Packet Inspection (DPI). Fig. 5 shows the difference between DPI and port-based method.

The basic principle of DPI is to check the contents of the packet payload with specific rules pre-programmed by the user, administrator, or Internet service provider (ISP). What is found is then used to decide what to do with the packet. DPI can identify the presence of a threat and find out where it came from through the contents of a packet and its header. DPI can pinpoint the application or service that initiated the threat. DPI can also be set up to work with filters, enabling it to identify and reroute network traffic from a specific online service or IP address [91].

The advantage of DPI technology is its ability to monitor network traffic in real-time and identify various applications and protocols commonly used by enterprises and Internet Service Providers (ISPs) to block network attacks, track user behavior, block malware, and monitor network traffic. However, the disadvantages of DPI technology are also obvious:

1. DPI technology requires in-depth analysis of packets, which consumes many computing resources and may affect network performance.
2. With the development of encryption technology, DPI technology has difficulty identifying encrypted traffic. In addition, because DPI technology relies on predefined rules, it may not recognize new types of applications and protocols and must ensure regular updates and modifications.
3. There may be privacy concerns with DPI technology because DPI can access specific information about where the information came from and where it went.

3.2. *Machine learning-based classification method*

The method based on machine learning is one of the mainstream technologies of encrypted traffic classification at present, and such methods usually rely on statistical features or time-series features. Support vector machine (SVM) [92], decision tree, random forest (RF), and other algorithms are used for modeling and recognition. In addition, statistical models such as the Gaussian Mixture Model are also used to classify encrypted traffic. According to the learning strategy followed by the model, classification methods based on machine learning can be divided into unsupervised, supervised, and semi-supervised learning. The current representative machine learning methods are shown in Table 6.

Table 6
Summary of encrypted traffic classification methods based on machine learning.

Class	Work	Year	Input	Model	Scenario
Supervised learning	Taylor et al. [21]	2016	Flow length+Statistical features	RF/SVC	MEAC
	Al-Naami et al. [22]	2016	Size, direction and duration of the bi-burst	SVM	GEAC/WF
	Panchenko et al. [23]	2016	Packet size, direction, and order	SVM	WF
	Hayes et al. [24]	2016	20 dimensional statistical features of the flow	Random Decision Forests	WF
	Li et al. [25]	2022	Packet size sequence order	Sequence-XG-Boost+Hierarchical-Bags-of-Words	MEAC (Behavior)
	Li et al. [26]	2022	123 dimensional statistical features of the burst	RF	MEAC (Behavior)
Unsupervised learning	Chen et al. [17]	2020	Connection features, SSL features, certificate features	Density Peak Clustering	EMC
	Fu et al. [18]	2023	Flow Interaction Graph	K-means	MATD
Semi-supervised learning	Conti et al. [19]	2016	Weighted bidirectional packet length sequence	RF+Hierarchical Clustering	MEAC (Behavior)
	V. Ede et al. [20]	2020	Bidirectional packet size sequence, packet inter-arrival time, TLS certificate and other statistical features	RF	MEAC

3.2.1. Machine learning method based on unsupervised learning

Unsupervised-based machine learning methods are a class of techniques used to process unlabeled data, where the main goal is to discover hidden structures or patterns from the data and group or classify the data without providing category labels beforehand [93]. These methods are widely used in many fields, including clustering [94], dimension reduction, anomaly detection, etc.

In encrypted traffic classification, unsupervised machine learning methods, such as the K-means algorithm, are often used for malicious traffic detection, abnormal traffic detection, etc. Fu et al. [18] proposed a malicious traffic detection framework based on real-time unsupervised machine learning, HyperVision, which can detect abnormal interaction patterns by analyzing the connectivity, sparsity, and statistical characteristics of graphs constructed based on traffic patterns to detect various encryption attack traffic without any marked data of known attacks. Liu et al. [17] proposed THS-IDPC, an unsupervised architecture based on an improved Density Peak Clustering (DPC) algorithm. This malicious traffic detection method extracts the statistical characteristics of the current flow from connections, SSL, and certificates, divides the dataset into small clusters, and then removes normal clusters from the overall cluster. Finally, XGBoost, support vector machine, and random forest are used to detect malware in encrypted network traffic and verify performance.

3.2.2. Machine learning method based on supervised learning

Supervised-based machine learning methods refer to the process of adjusting the parameters of a classifier to achieve the required performance using a set of samples of known classes, also known as supervised training or teacher learning. The objective is to infer a function from the labeled training data that can generalize well to new, unseen data, facilitating accurate predictions in practical applications.

In encrypted traffic classification, commonly used supervised machine learning techniques include RF, k-NN, SVM, and other algorithms. Panchenko et al. [23] presented an advanced method of website fingerprinting (WFP) attack that innovatively utilizes the cumulative sizes of packets, applying an SVM classifier with a radial basis function (RBF) kernel for highly effective differentiation of websites in encrypted traffic. This attack on Tor achieved an accuracy of 55%. In concurrent work, Hayes et al. [24] showed that due to Panchenko et al.'s feature set dependency on order and packet counting, the attack suffers substantially under simple website fingerprinting defenses. They proposed a website fingerprinting technique based on random decision forests. They performed well on datasets containing noise, and defense means because random decision forests can effectively handle diverse and noisy data, which is common in internet traffic, so the model enables the maintenance of high accuracy even when faced with website fingerprinting defenses and variations in the web page content. Al-Naami et al. [22] proposed a method called BIND based on consecutive bursts sequence characteristics, and on this basis, a method to overcome data drift was proposed, that is, to verify the model performance threshold according to the subsequent examples in the sliding window, to judge whether new samples are obtained to retrain and update the model. This method enriched the discriminant factors of terminal nodes in the process of traffic pattern recognition. The AppScanner modular fingerprint framework proposed by Taylor et al. [21] achieved over 99% accuracy classification effect on the traffic generated by mobile device applications. The algorithm divided the flow into bursts according to the duration of the behavior, and features such as packet size were extracted according to the burst shape to construct fingerprints. It innovatively identified common pattern traffic, as well as realized the recognition of unknown traffic in subsequent updates [20]. Regarding open-world fine-grained traffic identification, Li et al. [26] proposed a method-level fine-grained user action classification method FOAP for Android applications in an open-world environment. First, they adapted to filter out traffic segments unrelated to the application of interest by designing a concept of structural similarity between traffic. FOAP then identified user actions on a particular UI component by inferring entry point methods associated with that particular UI component. FOAP achieved an F1-score of 0.911 in application identification and an F1-score of 0.885 in user behavior identification, demonstrating the method's effectiveness in fine-grained recognition in the open world.

3.2.3. Machine learning method based on semi-supervised learning

Although supervised machine learning classification can perform well in most scenarios, obtaining large amounts of labeled data in real situations is often difficult. Machine learning methods based on semi-supervised learning solve this problem by using supervised information from a small amount of labeled data and structural information from a large amount of unlabeled data to train models simultaneously.

Conti et al. [19] proposed a semi-supervised machine learning classification framework to identify fine-grained mobile app behavior in mobile encrypted traffic identification. Firstly, weighted bidirectional packet length sequences were clustered into flow clusters, and then an RF algorithm was used to classify general application traffic. The

Table 7
Summary of encrypted traffic classification methods based on deep learning.

Work	Year	Input	Model	Scenario
Wang et al. [29]	2017	Flow/Session level packet length/timestamp sequence	1D-CNN	GEAC/ETCV
Sirinam et al. [30]	2017	Packet timestamp+Directional length sequence	SDAE/CNN	WF
Lotfollahi et al. [31]	2018	IP header and the first 1480 bytes of the payload	SAE/1D-CNN	GEAC/ETCV
Wang et al. [32]	2018	Packet bytes sequence	MLP/SAE/CNN	ETCV
Liu et al. [33]	2019	Directional packet length sequence	AE composed of Bi-GRU	GEAC
Zhang et al. [34]	2019	Local request and response sequence	Deep Forest	WF
Rong et al. [35]	2020	RGB image consisting of the first 900 bytes of one session	ResNet-50	EMC/MATD
Lin et al. [36]	2021	Packet sequence	CNN+LSTM	ETCV/EACT/EMC
Shen et al. [37]	2021	Downlink burst sequence	CNN	WF
Horowicz et al. [38]	2022	Flow temporal features histogram	SimCLR	ETCV/EACT/EAC-QUIC
Guo et al. [39]	2022	Grayscale image of the first 784 bytes	CNN	Few-shot EMC
Malekghaini et al. [40]	2023	Header of TLS handshake packets+Temporal features+Statistical features	CNN+LSTM	GEAC/EAC-QUIC
Jiang et al. [41]	2023	Packet timestamp+Directional length sequence	CNN	Drifted MEAC

results showed that it can achieve accuracy and precision higher than 95%, for most of the considered actions. Based on AppScanner, V. Ede et al. [20] proposed FlowPrint, a semi-supervised method generating application fingerprints by automatically discovering temporal correlations between destination-related network traffic features and using these correlations. This method can effectively identify mobile apps with 89.2% accuracy and detect unknown apps with a precision of 93.5% without prior knowledge.

3.2.4. Problems of existing machine learning methods

While machine learning methods can solve many problems that port-based and payload-based methods cannot, there are still some limitations:

1. Machine learning models cannot automatically extract and select features and need to rely on the experience of domain experts, resulting in significant uncertainties when applying machine learning to encrypted traffic classification.
2. Due to the real-time uncertainty of network traffic, the previously designed features are easy to fail and need to be constantly updated.
3. The generalization performance of traditional machine learning models is weaker than those based on deep learning or pre-training.

We now focus on deep learning-based methods, which leverage the power of neural networks to learn and extract complex features from data automatically. These methods enhance capabilities in dealing with modern network traffic's dynamic and encrypted nature, offering improved accuracy and adaptability. In the following sections, we delve into how deep learning and pre-training strategies revolutionize the field of encrypted traffic classification.

4. Deep learning-based classification method

Deep learning models that automatically learn complex structured feature representations and train classifiers directly from input data have rapidly developed in recent years [95]. Deep learning models show significant characteristics such as strong generalization, no need to extract features manually, and vital ease of use. Therefore, encrypted traffic classification methods based on deep learning have gradually attracted wide attention from researchers. Depending on the learning strategy followed, these methods can be divided into unsupervised and supervised learning-based methods. The current mainstream deep learning works are shown in Table 7.

4.1. Deep learning method based on unsupervised learning

4.1.1. Encrypted traffic classification based on Autoencoder

An Autoencoder (AE) is an unsupervised learning neural network model whose primary goal is to encode input data into a low-dimensional representation via an encoder, and then decode this low-dimensional representation back to the original data via a decoder, thus achieving data reconstruction. By comparing the differences between the original and reconstructed data, the autoencoder learns an efficient representation that can represent the original data with fewer dimensions. Additionally, autoencoders are highly effective in extracting complicated nonlinear features during the encoding process [96], making them particularly valuable in tasks involving complex data structures.

Researchers often use its peculiarity for effective feature extraction and compression of traffic and then continue the fully connected layer for classification. In their work, Lotfollahi et al. [31] and Sirinam et al. [30] proposed their classification frameworks based on Stacked Autoencoders (SAE) and Stacked Denoising Auto Encoder (SDAE), respectively. The former is a stacked network of several autoencoders symmetric about the hidden layer, where each autoencoder's output is the previous autoencoder's input. The architecture takes raw packets as input, and the training process is carried out in a greedy hierarchical way. The weights of other SAE layers are frozen when each SAE layer is trained, and after all layers are trained, the whole network adopts a backpropagation algorithm to adjust each layer's weights. Finally, the softmax layer is added to the last layer of SAE for classification, which performs well on application and service traffic classification tasks. The latter uses SDAE, which is constructed by adding noise to the input on the original SAE architecture. By reconstructing the original value from the noisy input, they achieve efficient website fingerprinting attacks against Tor.

4.1.2. Encrypted traffic classification based on Generating Adversarial Network

Generative Adversarial Network (GAN) is a generative deep learning model proposed by Goodfellow et al. [97]. GAN is composed of two neural networks; one is a generative model, and the other is a discriminative model. The task of the generative model is to generate instances that look naturally real and resemble the original data, while the task of the discriminant model is to determine whether a given instance looks naturally authentic or is falsified by the generative model.

In encrypted traffic classification, the strong generation ability of GAN is often used to deal with the imbalance of traffic datasets. Vu et al. [28] use an auxiliary classifier GAN (AC-GAN), which differs from the standard GAN in that AC-GAN inputs category labels with random noise at the input stage, thus generating enhanced datasets that can significantly improve the performance of downstream machine learning classifiers. GAN is often used in adversarial learning to improve the robustness of classification models. Tang et al. [27] proposed a GAN-based encrypted traffic representation and enhancement model. First, the encrypted traffic is converted into a Markov image. Then, a multiplicity-maximizing Markov GAN based on the Simpson index is designed to generate new Markov images. Finally, the balanced set of Markov images is sent to CNN for classification. The experimental results show that the enhanced Markov image set can effectively alleviate the generalization performance bias caused by different network depths.

4.2. Deep learning method based on supervised learning

4.2.1. Encrypted traffic classification based on Multi-Layer Perceptron

Multi-Layer Perceptron (MLP) is a basic feedforward artificial neural network consisting of one or more hidden layers, each composed of multiple neurons. Due to its high complexity and low classification accuracy, MLP is used less frequently in encrypted traffic classification. Wang et al. [32] proposed an encrypted packet classifier DataNet based on deep learning to support distributed end-to-end network QoS management, which was developed using three methods, including MLP, SAE, and CNN. The MLP consists of an input, two hidden, and an output layer. The input layer has 1480 inputs. The two hidden layers are composed of 6 neurons, respectively. The output layer consists of 15 neurons and uses softmax as a classifier to classify 15 classes of applications.

4.2.2. Encrypted traffic classification based on Convolutional Neural Networks

The convolutional Neural Network (CNN) is a feedforward neural network composed of multiple convolutional layers, correlation weights, pooling layers, and fully connected layers and trained by a backpropagation algorithm. Compared with other neural networks, CNN requires fewer training parameters and performs better, making it one of the most popular deep learning structures in recent years.

Due to the excellent ability of CNN in local feature extraction, as well as strong multi-layer feature extraction ability and self-adaptability, much work in the field of encrypted traffic classification has chosen CNN as its model architecture in recent years. Lotfollahi et al. [31] propose an encrypted traffic classification method based on CNN called Deep Packet. Compared with traditional methods, this method integrates the feature extraction and classification phases into one system, which can handle network traffic service classification and application identification tasks. Due to the distribution pattern of encrypted bytes that can be learned through the feature extraction of the whole packet, Deep Packet is excellent at identifying difficult-to-classify applications that use advanced port obfuscation techniques, such as P2P applications. Wang et al. [29] is the first to propose an end-to-end classification architecture based on CNN. His proposed 1D-CNN model contains 2 convolutional layers, 2 pooling layers, and 2 fully connected layers, takes the one-dimensional vector of the first 784 bytes of the packet as input, and evaluates the model performance on 12 categories of encrypted application datasets. It significantly improved over the C4.5 method using time-series and statistical features. Meanwhile, by comparing the classification performance of 1D-CNN to that of 2D-CNN, the authors verify that the structure of bytes, packets, sessions, and overall traffic is very similar to the structure of characters, words, sentences, and entire articles in the field of natural language processing. Sirinam et al. [30] proposed DeepFingerprinting, a deep attack method targeting Tor fingerprint defense methods WTF-PAD and W-T. The CNN architecture consists of four basic blocks, including convolutional layers, a maximum pooling layer, a filter and activation layer for feature extraction, and two fully connected layers for classification. The packet timestamp and length sequence are taken as inputs, and an excellent attack effect is achieved in both closed and open-world scenarios, respectively. Shen et al. [37] proposed BurNet, a fine-grained website fingerprint construction method using CNN. To extract the differences between similar web pages, the author proposed a new concept called unidirectional burst, a sequence of packets corresponding to a piece of HTTP packets. BurNet uses a unidirectional burst sequence as input, making it suitable for local and remote attack scenarios. BurNet performs well in closed and open-world scenarios and achieves 0.99 accuracies and 0.99 recall in open-world environments.

In recent years, researchers have also explored domain adaptive, few-shot, and zero-shot learning tasks in the field of encrypted traffic based on CNN architecture. Horowicz et al. [38] proposed a traffic few-shot learning model named mini-FlowPic. This model expands the scale of training samples by replacing large labeled datasets with a small number of samples and combining data enhancement techniques. In continuous time slices, histograms of packet sizes are converted into pictures and fed into a CNN model for service traffic classification. This method achieves good accuracy with at most 10 samples per class. Jiang et al. [41] proposed a domain adaptive neural network (FDAN) to improve the accuracy of mobile application identification over non-independent and identically distributed (non-IID) traffic under zero-relabeling. FDAN uses two domain discriminators and a feature generator to enhance feature invariance under adversarial loss. This reduces the differences between the flow distributions in the trainable feature space and converts the drifting test flow into an approximate independent identically distributed (IID) sample.

4.2.3. Encrypted traffic classification based on Recurrent Neural Networks

Recurrent Neural Network (RNN) is a class of neural networks used to process sequential data. Unlike traditional neural networks, RNN has a cyclic structure that allows information to be passed around the network, enabling it to model and process sequence data.

Because of sequence modeling and recursive updating characteristics, RNN can deal with the structure and pattern of sequence features well. In encrypted traffic classification, RNN and its variants LSTM and GRU are often used to process time-series features, such as packet size, packet direction, and packet interval arrival time of a flow [98]. Liu et al. [33] proposed FS-Net, an end-to-end flow sequence network composed of encoder, decoder, classifier, and reconstruction layer, and reconstructed the original flow sequence through an autoencoder based on GRU. The end-to-end framework enables FS-Net to automatically learn representative information from the data, while the refactoring mechanism enhances feature representability and improves classification performance. The author verified the efficiency of FS-Net on real network traffic datasets, and an average of 99.14% TPR was achieved on general-purpose encrypted application traffic captured in the campus network environment.

Nowadays, researchers often use CNN in combination with RNN. CNN is used to extract spatial features, while RNN is used to extract temporal features. Such a combination improves the classification performance significantly. Lin et al. [36] proposed TSCRNN combining CNN and RNN. Firstly, abstract spatial features of flows are extracted, and sequential features are learned to achieve efficient identification. It performs traffic characterization tasks (16 classes) on the ISCXTor2016 dataset and achieves an average accuracy of 94.8%, better than traditional machine learning and deep learning methods. Malekghaini et al. [40] discussed the impact of data drift on the classification model based on the University of Waterloo third-party model (UW model). The UW model is constructed by a series of CNNs that extract shift invariant information from TLS handshake header bytes, a series of LSTMs that extract conventional timing features, and a series of dense layers that process statistical features, including 77 features.

Table 8
Comparisons of common graph neural networks.

Model	Structure	Advantage	Disadvantages
GNN [106]	Based on a recursive neighborhood aggregation scheme	Good at handling graph-structured data for various tasks	Sometimes struggles with complex graph structures and large-scale data
GCN [99]	Utilizes a layer-wise propagation rule based on neighborhood aggregation	Effective in capturing local graph structures and features of nodes	Poor scalability and likely over-smoothing when the adjacency matrix is sparse
ResGCN [107]	Incorporates residual connections to enhance learning in deeper network layers	Alleviates the vanishing gradient problem, enabling deeper architectures	A deep GCN with residual joins is sometimes no better than a common GCN
DGCNN [108]	Features deeper layers and more complex network architectures	Captures more complex, hierarchical, and long-range patterns in data	Potentially more prone to overfitting and requires more computational resources
GAT [100]	The self-attention mechanism used to allocate attention to a node and its neighbors	Different weights for different nodes	Only first-order neighbor information, no further depth for higher-order information

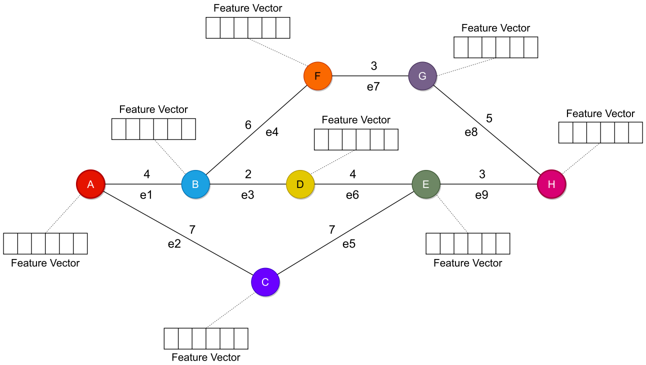


Fig. 6. Example of graph data.

4.2.4. *Encrypted traffic classification based on Graph Representing Learning*

In recent years, graph-based representation learning techniques have shown great potential in this field. The core of encrypted traffic classification technology based on graph representation learning is to convert traffic data into graphs and use Graph Convolutional Networks (GCNs) [99], Graph Attention Networks (GATs) [100], etc., to extract and learn the features of these graphs. Precisely, this technique captures complex interactions between entities by treating network traffic as a graph structure: entities in network traffic (such as IP addresses and port numbers) are treated as nodes of the graph, while edges represent communication behavior [101,102]. This graphical representation captures deep structural and temporal characteristics, which is particularly effective for identifying patterns in encrypted traffic.

As shown in Fig. 6, in the domain of graph theory, a graph G is mathematically conceptualized as a duo (V, E), where V signifies an assemblage of nodes and E delineates the edges forging connections between these nodes. Every node v in V is associated with a feature vector x_v in a d-dimensional realm, forming part of the comprehensive feature matrix X for all nodes, represented as $X \in \mathbb{R}^{d\times|V|}$ [103]. Additionally, edges in E, connecting nodes u and v, are quantified by a weight $e_{u,v}$ in $\mathbb{R}$, symbolizing the interaction strength or connectivity level within the node pair u and v in the graph structure. This representation encapsulates the nodes' individual characteristics and the graph's relational dynamics [104].

The commonly used partial graph neural networks in the domain are shown in Table 8. Although there are many kinds of graph representation learning models, they all have one thing in common: neighborhood aggregation, in which the embedding of each node is updated by the embedding of the neighborhood [105]. These methods adapt neural network methodologies to graphs, offering distinct strategies for interpreting and learning from graph-structured information.

Since traffic is the information transmitted by interaction between different communication entities, graph neural networks are naturally suitable for representing network traffic: nodes are usually regarded as different entities, and edges are regarded as interaction behaviors between entities. By learning the graph structure, the model grasps the whole communication mode; therefore, graph-based representation learning is widely used for traffic classification in various scenarios. Some typical GNN-based works are shown in Table 9.

To solve the only flow problem in the VPN proxy environment, Xu et al. [109] proposed VT-GAT, a VPN traffic graph classification model based on Graph Attention Network (GAT). Compared with the existing VPN encryption traffic classification technology, VT-GAT solves the problem that the previous approach ignores graph connectivity information in the traffic. VT-GAT first builds a traffic behavior graph by describing raw traffic data at the packet and flow level. Then, the GCN and attention mechanism are combined to extract the behavior features automatically from the traffic graph data. Many experimental results on the Datacon21 dataset show that VT-GAT can achieve more than 99% on all classification metrics. VT-GAT improved the F1-score by about 3.02% to 63.55% compared to existing machine learning and deep learning methods.

The decentralized app deployed on Ethereum implements the same front-end interface, uses a similar SSL/TLS protocol setting, and shares the same decentralized blockchain network to run back-end code and manage data. As a result, traffic from different DApps has many common characteristics that invalidate existing methods based on SSL/TLS packet flags and packet length statistics. Shen et al. [110] proposed GraphDApp, a novel fingerprint identification method for decentralized applications (DApp) based on graph neural networks. In the input phase, a graph structure named traffic interaction graph (TIG) is proposed to represent each individual encrypted flow, in which the vertices in TIG represent packets, and the edges represent a pair of packet-level interactions between client and server, thus implicitly preserving multidimensional features in two-way interactions. Thus, the DApp fingerprint identification problem is transformed into a graph classification problem, and GraphDApp has shown excellent performance in both closed and open real-world scenarios.

Pham et al. [45] proposed a mobile application classification method, MAppGraph. In order to overcome the problem that mobile applications often share third-party services, the author defines a graph node using the tuple of the destination IP address and the port number of the service-connected by the application, and each node contains 63 features. The edge of the graph is defined by whether there is a communication relationship between two nodes in the predefined time slice. Then, the author uses the graph convolution layer in DGCNN and the traditional convolution layer to perform multi-level feature extraction. In the experimental phase, the authors demonstrate that the proposed method outperforms Appscanner and Flowprint because DGCNN has multiple graph convolution layers and considers the diversity of mobile application behavior by training a single DGCNN model on multiple graphs. The advantage of MAppGraph over Flowprint is that it does not need to compare the fingerprints obtained from the test sample with all the pre-calculated fingerprints.

For effective traffic classification in unstable networks, Diao et al. [48] introduced EC-GCN, a multi-scale graph convolutional network

Table 9
Summary of encrypted traffic classification methods based on graph representation learning.

Work	Year	Input	Model	Scenario
Jiang et al. [42]	2019	Flow relationship graph consisting of the edge of intra-burst and inter-burst relationships and the arrival timestamp sequence	GNN	Drifted MEAC/EDAC
Sun et al. [43]	2020	k-NN traffic graph	GCN+AE	EMC/ETCV
Shen et al. [44]	2021	Traffic interaction graph consisting of packet length and inter-entity communication modes	GNN	EDAC
Pham et al. [45]	2021	Graph representing app-connected services in which four kinds features as nodes, with edges based on communication frequency	DGCNN	MEAC
Zhao et al. [46]	2022	Flow features sequence graph	ResGCN	ETCV/EACT
Zola et al. [47]	2022	Graph with nodes representing entities (IPs and ports) and edges indicating traffic exchange	GCN/NN	MATD
Xu et al. [109]	2022	Traffic behavior graph composed of nodes with 77-dimensional features and edges composed of communication frequencies	GAT	ETCV
Diao et al. [48]	2023	Sparse graph of packet length sequence	GCN	GEAC/GEAT

approach for encrypted traffic. This model uses a lightweight encoding layer based on metadata without packet arrival time, avoiding dynamic environment interference, and is independent of encryption protocols. Specifically, EC-GCN includes new graph pooling and structure learning layers for simplified graph representation learning. It consists of 6 temporal blocks for capturing time correlations and forming low-dimensional features, 6 spatial blocks for multi-level graphical representations, and a fully connected layer. Tests on three datasets showed EC-GCN outperforms existing methods with a 5%–20% accuracy improvement.

Zhao et al. [111] proposed a novel traffic identification framework based on residual graph convolutional networks (ResGCN). Unlike conventional graph-based representation methods, this work directly takes flow feature sequences as nodes and sets the weight of the edge based on the relationship between the flows. In order to retain the spatial structure of the gradient better, the residual structure is used to extract features. In addition, as an end-to-end framework for recognizing real-world traffic, it extracts rich features from the original traffic according to the traffic segmentation scheme. It uses the Light Gradient Boosting Machine (LightGBM) algorithm to select the optimal feature combination to improve model performance and efficiency. Compared with the methods of CNN, LSTM, and LDAE that use statistical features and flow sequences, this method can achieve higher accuracy because it mines the attribute and temporal relationship of flow sequences.

In terms of few-shot learning, Sun et al. [43] proposed a framework learning feature representations from structure and data, respectively. Specifically, the author constructs the k-NN traffic graph, which is constructed according to the similar flows in the Euclidean space selected by k-NN. Compared with the traditional flow graph, more connections are established between the flows of the same application. The model comprises a two-layer GCN and autoencoder, thus improving the ability to learn efficient feature representations from a low-resource environment. The first layer GCN receives the structure of the graph and the node feature matrix as inputs, and the original compressed representation of traffic learned by the autoencoder is passed to the learned representation of the first layer GCN as inputs to the second layer GCN to prevent from overemphasizing the association of adjacent nodes while ignoring the features of the nodes themselves.

To sum up, graph-based representation learning approaches are particularly prominent among deep learning-based approaches due to the high density of information stored, data construction in line with network communication patterns, and the strong topological retention and relationship modeling capabilities of graph-based neural networks when dealing with non-Euclidean data [112]. In addition, with the continuous progress of algorithms and computing power, the scalability and efficiency of graph neural networks in processing large-scale graph data are constantly improving, which further expands its application prospects in different realistic scenarios. In the foreseeable large model-driven future, graph-based representation learning is becoming a hot topic for researchers in this field.

4.3. Limitations of existing deep learning methods

Although classification method based on deep learning is currently a more promising direction in this field, and has achieved good results, it also has some inherent problems:

1. Deep learning models need a large amount of labeled data for training, and the data needs to be evenly distributed and representative. Nevertheless, one dataset that accurately and comprehensively characterizes all encrypted traffic types needs to be included.
2. Deep learning models can suffer from overfitting, resulting in reduced accuracy in real-world applications.
3. The generalization ability of models specific to a specific scenario decreases significantly in other scenarios. However, due to the complexity of reality, it takes human resources to design corresponding deep learning models for different scenarios to solve real needs.

In view of the above problems, a more robust classification framework is urgently needed, and the pre-training-based approach has emerged.

5. Pre-training based classification method

In recent years, the rapid development of deep learning and natural language processing has provided a new way to solve the problem of encrypted traffic classification. Among them, the encrypted traffic classification method based on a pre-trained model has been paid more and more attention. Pre-trained models are neural networks pre-trained on large-scale unlabeled data and can learn generic semantic information. Then, they can be fine-tuned on specific tasks to suit specific application scenarios.

Encrypted traffic classification based on a pre-training approach has essential research value and application prospects in network security. By exploring and developing new approaches in this area, we are expected to improve our ability to classify encrypted traffic accurately, thereby better responding to growing cybersecurity threats and privacy protection needs.

In this section, we will focus on applying encrypted traffic classification based on a pre-training approach. First, we will discuss the fundamentals and advantages of pre-training technology. Then, we will elaborate on the principles and technical characteristics of this kind of work. Finally, we will summarize the advantages and disadvantages of current approaches and look forward to future developments in this field.

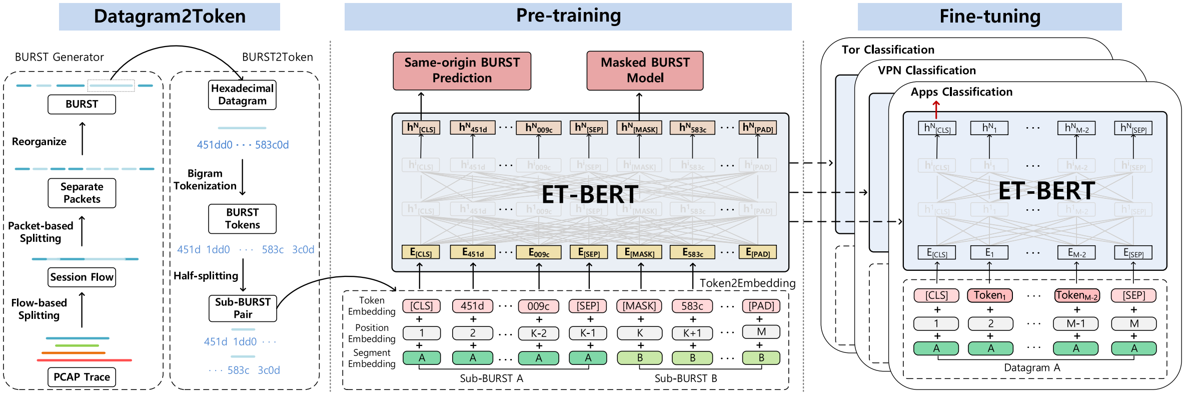


Fig. 7. The Framework of the Encrypted Traffic Classification based on Pre-training.

5.1. Pre-training-based classification framework

Although it is easy to obtain massive traffic data in the current cyberspace environment, most of these samples are unlabeled, and it takes an expensive workforce to label them. In addition, most existing models are expert models designed for specific needs and are often difficult to migrate to other scenarios easily. At the same time, due to the complexity and variability of the network environment, the traffic transmission behavior in specific scenarios is more changeable, and the effect of the existing traditional model will be significantly reduced due to the poor robustness. All these problems together pose a serious challenge to the existing classification model.

In recent years, pre-training frameworks using large-scale unlabeled data have emerged in the fields represented by natural language processing (NLP) and computer vision (CV) and have shown the best performance in various scenarios. Like natural language data, encrypted traffic data has a sequential context. At the same time, different types of packets contain various information about the communication, similar to words or phrases in natural language [29]. Therefore, after a series of adjustments, the pre-training framework in NLP can be naturally transferred to the field of encrypted traffic. The pre-training framework represented by PERT [49] and ET-BERT [50] proves this strategy's feasibility and high efficiency in the field of encrypted traffic. Compared with traditional methods, the classification framework based on pre-training can automatically learn universal communication patterns from the unlabeled traffic data of multiple scenarios and achieve accurate classification of specific scenarios downstream by fine-tuning a small amount of labeled data.

As shown in Fig. 7, the encrypted traffic classification framework based on pre-training is divided into three stages: pre-processing, pre-training, and fine-tuning. Among them, the pre-processing stage is responsible for processing large-scale unlabeled original traffic data into the form required by the model. Through a series of pre-training tasks, models such as BERT and GPT learn the universal characteristics and structure of encrypted traffic data. Through these tasks, the pre-training stage enables the model to understand the characteristics of encrypted traffic data at different levels. The fine-tuning phase builds on the pre-training phase. It aims to fine-tune the model parameters with a small amount of labeled traffic data to fit a specific encrypted traffic classification problem better. To sum up, the encrypted traffic classification framework based on pre-training, through the combination of pre-processing, pre-training, and fine-tuning stages, enables the model to learn general characteristics from a large number of encrypted traffic data and adapts to specific tasks to achieve precise classification and identification of encrypted traffic, as well as subsequent expansion and update.

5.2. Pre-processing stage

The pre-processing stage is a crucial component of the pre-training framework, which significantly impacts the model's performance and universal applicability. The model accepts large-scale data in this stage, usually unlabeled raw encrypted traffic samples. In this phase, the original data is processed to a certain extent to facilitate learning functional feature representation and semantic information in the subsequent pre-training stage. Therefore, whether the design of the pre-processing stage is appropriate or not will significantly affect the success of the overall design of the pre-training framework.

5.2.1. Payload

Currently, the input content of the mainstream pre-training framework consisting of the Transformer encoder-based model like BERT is an unlabeled sequence of raw traffic bytes. Without loss of generality, this phase takes the tokenized traffic payload sequences $X = [x_1, x_2, \dots, x_N]$, as input, which is then transformed into the representation consisting of the sum of the token embedding and the positional encoding.

$$Em_{sum} = Em_{token} + En_{positional}$$

where the former is represented by a real-valued vector obtained by looking up the embedding matrix so as to realize the mapping of the original input sequence to the higher dimensional space. The token embedding is defined as:

$$Em_{token} = Lookup_{W^t}(X)$$

where $X \in R^{batch\_size \times N}$ represents the input sequence, N is the length of sequence, while $W^t \in R^{|V| \times em\_d}$ is the embedding matrix, $|V|$ is the size of vocabulary, em_d is the dimension of the hidden layer and is usually 768 [49,50,55]. The positional encoding is set to introduce position information to model inputs since self-attention mechanism-based methods cannot discern contextual relationships in input sequences, and different models have different positional encoding settings. Through the above series of operations, we get the final embedding, that is, as the input of the next pre-training stage.

He et al. [49] proposed PERT, which is the first time to apply the idea of "pre-training+fine-tuning" to the innovative work in encrypted traffic classification. They convert the payload byte data from the original traffic packet into a string form through appropriate tokenization. Specifically, the work treats the payload bytes of the packet as language-like strings that serve as inputs to the BERT model. In traditional natural language processing, the scope of a vocabulary is much larger than the range of a single byte. Therefore, when dealing with network traffic data, extending the range of single-byte representation is necessary to use NLP technology better. To expand the size of the vocabulary of traffic bytes, the author introduced a tokenization

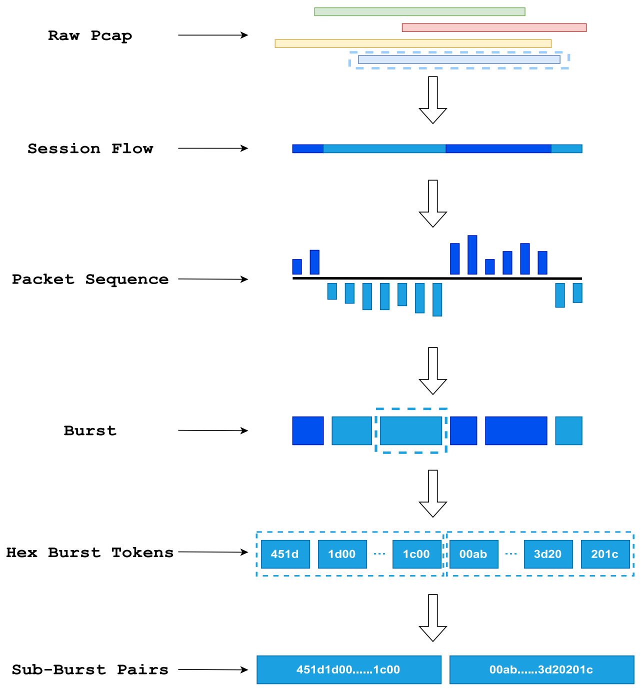


Fig. 8. Structure of burst.

method that uses byte pairs with values ranging from 0 to 65535 as base character units to generate binary strings. After that, NLP-related encoding methods can be applied directly to tokenized traffic bytes. BERT’s word embedding approach is then adopted, and the word embedding is dynamically adjusted after each encoder, in which a more abstract context representation of the original traffic load is further learned.

Lin et al. [50] made improvements on this basis and introduced the concept of burst. As shown in Fig. 8, the burst is regarded as packets with adjacent time originating from the request or response of a single session flow. The burst is defined as:

$$Burst = \left\{packet_n^t\right\},\ n \in N^+$$

$$t = \{IP_{src,\ dst}, Port_{src,dst}, Protocal\}$$

where n indicates the number of unidirectional packets, t indicates that these packets share the same five-tuple.

In the application layer, the personalization of web services leads to diverse Document Object Model (DOM) trees across web pages. Client-side rendering divides web data into objects like text and images, creating semantically distinct DOM fragments that subtly affect resource requests. Each fragment forms a network burst containing complete content of a specific type. From this perspective, a sequence of such bursts describes the network flow’s transmission pattern, and tokenizing the original traffic byte information based on these bursts more accurately reflects the traffic transmission characteristic. The author also added unique tokens for subsequent pre-training tasks: the first token of each sequence [CLS], which represents the hidden state of the sequence; [PAD] is used to fill the sequence to meet the minimum length requirement; as shown at the bottom of Fig. 8, [SEP] is used to separate two sub-burst pairs; [$MASK$] is used for traffic context learning in the pre-training phase. Then, each token obtained in the above process is represented by three kinds of embeddings: token embedding, position embedding, and segment embedding. Segment embedding is used to make the model learn burst context association. By combining the three embeddings as model inputs, multi-granularity raw traffic information is extracted. Most current pre-training frameworks based on raw bytes as input information follow a similar pattern.

Unlike most existing methods that represent traffic as natural language, the YaTC framework proposed by Zhao et al. [52] represents the original traffic as an MFR matrix with a formatted two-dimensional matrix containing information at three levels of bytes, packets, and flows. Specifically, each row of the matrix contains only one type of traffic bytes, divided into header and payload rows. Each packet is represented as a combination of a header matrix and a payload matrix, collectively forming a packet-level matrix of size $H/M * W$. Here, H and W denote the height and width of the MFR matrix, respectively, while M represents the number of packet-level matrices. Finally, since the flow comprises ordered packets, M adjacent packet-level matrices are stacked on the second dimension to form the final MFR matrix. By doing so, raw traffic can be used in the Masked Autoencoders (MAE) paradigm [113] for pre-training and fine-tuning tasks.

5.2.2. *Side-channel feature*

In part of the work, side-channel information, such as time-series features, which are also selected as model input. Cai et al. [61] proposed a self-supervised mobile encrypted traffic classification method, METC-MVAE, which is similar to ViT [114] in that the directed packet length sequence is divided into fixed-size windows, and then windows embedding is obtained through a linear transformation. In addition, position embedding is used to make the model learn the relationship between tokens. Finally, the combination of windows embedding and position embedding is used as the input of the pre-trained model. In the downstream, the raw traffic is reconstructed through the corresponding pre-training task to capture the context connection between the various traffic windows.

Compared with the input based on the payload, the number of pre-training methods based on the side-channel information as input is less, possibly because the raw byte sequence provides more abundant information. This data contains complete traffic details, allowing the model to capture deeper patterns and characteristics. In contrast, time-series features such as packet length and arrival intervals are more abstract and may not fully capture all the essential information about traffic data. At the same time, as traffic encryption and obfuscation technologies continue to evolve, existing side-channel feature construction methods may be insufficient to distinguish between different types of traffic. Researchers can continue to explore the potential of side-channel input on the existing basis.

5.2.3. *Synthesis input*

In addition to the above two separate input forms, the input to the model combines raw traffic byte sequences with side-channel information, enabling the learning of broader contextual patterns that capture the original traffic characteristics and communication environment information. Lin et al. [53] proposed a multi-modal end-to-end pre-training framework, PEAN, which uses raw bytes and packet length sequences as inputs to enable the model to capture traffic patterns and background information. Specifically, the architecture of PEAN has five layers. The pre-training layer is the first layer, which takes the Transformer encoder as the core component and takes the original message bytes as the input. Its multi-head self-attention mechanism enables it to learn the byte-level embedding model to acquire the ability to represent bytes through unsupervised pre-training. The second layer is the packet coding layer, which reuses the pre-trained model parameters in the previous layer and uses the [$packet$] token added before each network packet to represent packet-level embedding. The third layer, the sequence layer, uses the new Transformer encoder to learn the context between each packet embeddings to obtain the embedded representation of network flow. The fourth layer is the supplementary layer. Due to the sequential nature of traffic data, rich feature information can be captured from the traffic sequential information. It uses the packet length sequence to perform LSTM modeling and learn

Table 10
Summary of encrypted traffic classification methods based on Transformer encoder.

Work	Year	Input	Model	Scenario	Characteristic
He et al. [49]	2020	Raw packets sequence	ALBERT	ETCV/MEAC	The first work to introduce pre-training techniques into the field of encrypted traffic classification
Lin et al. [50]	2021	Sequence of packets of bursts in the session flow	BERT with same-origin burst prediction and masked burst model tasks in pre-training stage	GEAC/EMC/ETCV EATC/EAC-TLS 1.3	The burst concept is introduced into traffic pre-training, and the effectiveness of this work is proved in the downstream multi-scenario and few-shot learning scenario including TLS1.3 traffic classification
Zhao et al. [52]	2023	Traffic representation matrix with hierarchical flow information	MAE-based Transformer	EMC/ETCV/EACT/ GEAC/IoTDTC	The self-supervised learning paradigm MAE is successfully introduced into the field of encrypted traffic classification by modeling the byte-level, packet-level and flow-level of the raw traffic
Lin et al. [53]	2023	Raw traffic bytes+ Packet length sequence	Transformer encoder+LSTM	GEAC	A multimodal pre-training framework combining Transformer encoder and LSTM is proposed to learn the feature contained in the original flow bytes and length sequences respectively
Li et al. [54]	2023	Raw packets sequence+ Statistical features for flow	Transformer encoder+1D-CNN	EMC	The outputs of Transformer encoder with raw packets as input and 1D-CNN with statistical features as input are concatenated for classification
Lei et al. [55]	2022	Sequence of packets of bursts in the session flow	ET-BERT	IoTDTC	To improve the learning strategy of ET-BERT, RNN Drop is adopted
Shi et al. [56]	2023	Raw packets sequence	BERT+CNN	ETCV	Unlike the classic BERT+CNN work in the NLP domain, this work uses hidden layer states associated with [CLS] tokens to represent global features and 1D-CNN without pooling layers to learn local features
Hu et al. [57]	2021	Raw packets sequence	1D-CNN+BERT	ETCV	It is proved that eight-layer BERT can achieve the best results, and the model can achieve 70% accuracy at 20% sample size and 84% at 40%
Lin et al. [51]	2024	Raw datagram and packet length sequences	BERT with contrastive learning	GEAC/EMC/EAC-QUIC	A semi-supervised learning framework combining contrastive pre-training and pseudo-label iteration to solve class imbalance and traffic homogeneity problems in real-world ETC

Table 11
Summary of encrypted traffic classification methods based on Transformer decoder.

Work	Year	Input	Model	Scenario	Characteristic
Meng et al. [58]	2023	Raw packets sequence	GPT-2	ETCV/MATD/EMC/ Five-tuple generation	In the fine-tuning stage, it is set to shuffle header fields, segment packets in flows and incorporating diverse task labels, which is used to make GPT-2 learn traffic patterns better, and design more downstream tasks including simulated header segment generation
Qu et al. [59]	2024	Raw packets sequence	GPT-2 (Linear attention mechanism)	EVTC/IoTDTC/EMC	A GPT-based model with linear attention is proposed, enabling the processing of sequences up to 12032 tokens for long flow classification and generation tasks
Bikmukhamedov et al. [60]	2020	Directional packet length+Arrival timestamp sequence	GPT-2+K-Means	EMC/IoTDTC/GEAC	This work can perform traffic generation tasks, and the quality of generation is comparable to that of traditional Markov-based methods

Table 12
Summary of encrypted traffic classification methods based on Transformer.

Work	Year	Input	Model	Scenario	Characteristic
Cai et al. [61]	2022	Directional packet length sequence	Transformer +MLP	MEAC/EACT/ EDAC	Two pre-training tasks, including traffic window reconstruction model and VAE-based traffic generation, correlations between are proposed to capture potential traffic windows and patterns of behavior
Dai et al. [62]	2022	Header+Payload sequence	Transformer	EACT/EACV/ GEAC	During encoding and decoding, the header and payload of the packet are independently operated to ensure the integrity of the data of different structures in the packet

their sequential correlation to learn the supplementary features of the model from the second mode. The final classification layer combines the hidden features of the sequence layer and the supplementary layer to perform fusion and classification. The author conducted ablation experiments and found that the F1-score of the PEAN without the packet length sequence input decreased by more than 3% compared with the original PEAN, which proved the effectiveness of timing information as a pre-training framework to supplement the input.

TC-Mal, the malicious traffic classification framework proposed by Li et al. [54], consists of two subnets: T-net and C-net. T-net can perform unsupervised pre-training through a Transformer for representation learning of raw packets, while C-net can extract flow-level features through one-dimensional CNN. Specifically, the former uses a PERT [49] like framework that takes raw packet bytes as input. The latter extracts the statistical features of the session flow in the original dataset, including 76 dimensions such as flow duration and

Table 13
Summary of encrypted traffic classification methods based on traditional deep learning model.

Work	Year	Input	Model	Scenario	Characteristic
He et al. [63]	2021	Raw traffic two-dimensional matrix	CNN+AE	MATD/IoTDTC	In the pre-training process, only benign data is used, and the few-shot abnormal traffic samples are detected by comparing the autoencoder reconstruction loss
Liu et al. [64]	2022	Raw session flow bytes sequence	1D-CNN	MATD/EMC/GEAC	Using one sample of each type of newly emerged malicious application traffic, FewFine can achieve a detection accuracy of 0.9765, proving that it can achieve anomaly detection in few-shot scenario

total backward packets, and inputs them into 1D-CNN. Finally, a classifier is placed behind the two subnets to take over the output of both and produce classification results. Compared to subnets alone, the framework achieved between a 2.8% and 11.83% improvement of the F1-score in three scenarios.

In summary, the pre-processing phase is an integral part of the pre-training framework, providing a solid starting point for the model to perform well in various downstream tasks through learning large-scale traffic byte information and capturing feature representations.

5.3. Pre-training stage

The pre-training stage is the core component of the pre-training framework. The data processed in the previous stage is taken as the input. The model can learn the general traffic transmission mode at different levels through specific self-learning pre-training tasks and realize the generalization basis across different downstream scenarios.

According to the selected model, the existing methods can be divided into Transformer encoder-based pre-training, Transformer decoder-based pre-training, and traditional deep learning model-based pre-training. Existing pre-training methods based on Transformer encoders are summarized in Table 10, methods based on Transformer decoders are summarized in Table 11, methods based on the Transformer are summarized in Table 12, and finally, methods based on traditional deep learning are summarized in Table 13.

5.3.1. Pre-training based on transformer encoder

The Transformer model was initially proposed by Vaswani et al. [4] in 2017 and is designed to handle various tasks of sequence data in NLP, such as machine translation, text generation, and sentiment analysis. Unlike traditional RNN and CNN, the Transformer model adopts a new architecture emphasizing parallelization and the ability to model long-distance dependencies.

Fig. 9 shows that the Transformer encoder combines word embedding and positional embedding as model inputs. In the encoder part, the output matrix is obtained by the multi-head attention layer, the add & norm layer and the feed forward layer so as to obtain the encoded information of the input sequence.

The self-attention mechanism is a core component of the Transformer encoder, and its fundamental idea is that when processing input sequences, the model can dynamically assign attention weights to different locations to capture the relationship between different locations better. The self-attention mechanism works in the following ways.

For each position in the input sequence, the self-attention mechanism computes three sets of vectors: Query (Q), Key (K), and Value (V). These vectors are learned from the weight matrix. To be more specific, three weights are assigned $W_Q,\ W_K,\ W_V \in R^{em\_d \times em\_d}$, where em_d is the embedding layer dimension described in Section 5.2.1 and three matrices $Q,\ K,\ V$ are constructed after linear mapping, which is made of Em_{sum} and three weight matrices above. The generation process of three sets of vectors is described as follows:

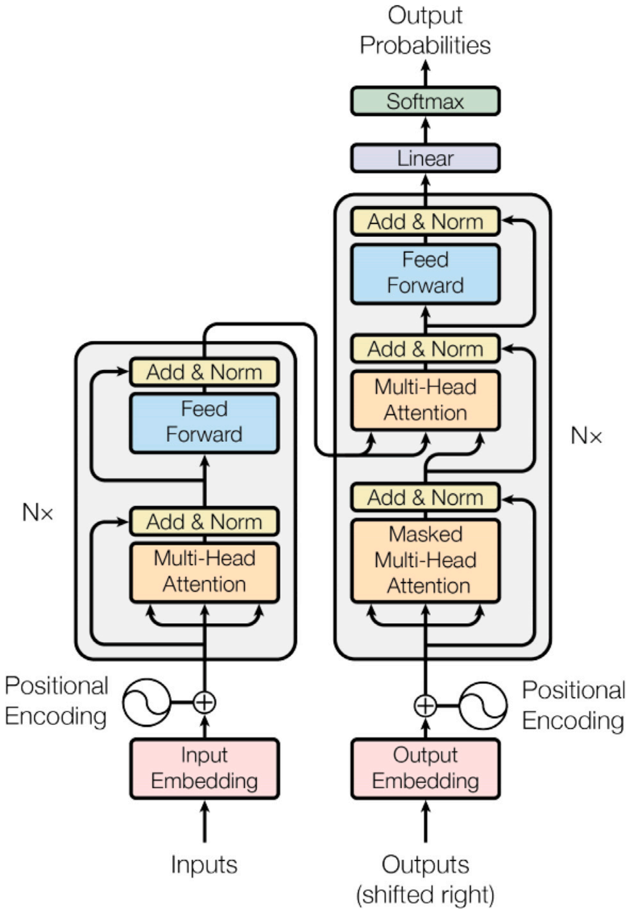


Fig. 9. Transformer architecture.

$$Q = Concat(Q_0, Q_1, \dots, Q_{h-1})$$

$$= Concat(Em_{sum}W_{Q_0}, Em_{sum}W_{Q_1}, \dots, Em_{sum}W_{Q_{h-1}})$$

$$K = Concat(K_0, K_1, \dots, K_{h-1})$$

$$= Concat(Em_{sum}W_{K_0}, Em_{sum}W_{K_1}, \dots, Em_{sum}W_{K_{h-1}})$$

$$V = Concat(V_0, V_1, \dots, V_{h-1})$$

$$= Concat(Em_{sum}W_{V_0}, Em_{sum}W_{V_1}, \dots, Em_{sum}W_{V_{h-1}})$$

where $Q,\ K,\ V \in R^{batch\_size \times h \times N \times em\_d/h}$, h represents the number of heads.

Then, by calculating the dot product between the query vector and all the key vectors, the attention score for each location is obtained with respect to the others. These scores were normalized by softmax to get how much attention each location paid to the others. Finally, the output for each position is obtained by multiplying the attention score with the value vector and weighted summing all positions.

The output of the multi-head attention layer is defined as follows:

$$MultiHead\left(Q,\ K,\ V\right)$$

$$= \ Concat\left(head_0, head_1, \dots, head_{h-1}\right) W^O$$

where $MultiHead\left(Q,\ K,\ V\right) \in R^{batch\_size \times N \times em\_d}$ and $head_i$ is defined as follows:

$$head_i = Attention\left(Q_i, K_i, V_i\right) = softmax(\frac{Q_i K_i^T}{\sqrt{d_k}})V_i$$

where $d_k \in R^{batch\_size \times \ N \times em\_d/h}$.

As shown in Fig. 9, the sub-layer value is added to the input embedding, and then, the output is normalized to the standard normal distribution. The final output of the multi-head sub-layer is:

$$Output = Norm(MultiHead\left(Q,\ K,\ V\right) + Em_{sum})$$

The self-attention mechanism allows the model to focus on different parts of the input sequence dynamically. It enables it to handle long-distance dependencies and perform well in various downstream traffic classification and identification tasks. In models like BERT, during the pre-training phase, self-attention collaboratively works with pre-training tasks to enhance the understanding of traffic patterns, thereby improving the model's overall effectiveness in contextual comprehension.

In encrypted traffic, Transformer encoders are often used as the base model in the pre-training stage. The pre-training classification framework proposed by Lin et al. [50] consists of multi-layer bidirectional Transformer encoder blocks [5]. Each piece consists of a multi-head self-attention layer containing 12 self-attention heads, which captures the implicit relationship between the encoded traffic units in the datagram. In the pre-training stage, the author adopted two tasks of Masked Burst Model and Same-origin Burst Prediction, namely random masking and replacement of burst marker units: the former will randomly mask each tag in the input sequence with a 15% probability, and randomly mask the tag with an 80% probability of replacing it with $[MASK]$, or choose a random tag to replace it, or leave it unchanged with a 10% probability; the latter uses a binary classifier to predict whether two sub-bursts come from the same source. The final pre-training goal is to reduce the sum of the losses of the above two tasks so that the model learns the context of the packet and the burst level, respectively, and realizes the model's learning of the general communication mode on large-scale unlabeled data.

The YaTC framework [52] consists of embedding modules, packet-level attention modules, and flow-level attention modules. First, the MFR matrix is divided into patches and mapped into embedding vectors. Then, the self-attention mechanism is implemented between the packet sub-matrices in the MFR matrix and the global. During its pre-training stage, YaTC [52] employs the MAE framework and a unique encoder–decoder structure for reconstructing the raw bytes in the MFR matrix. This process involves masking a large portion of MFR patches, with only a minimal subset being fed into the model. The traffic encoder then endeavors to extract substantial valid features, culminating in the generation of an encoder token. This token, along with a mask token, is instrumental in restoring the masked segments of the MFR matrix. The MAE's objective is to minimize reconstruction losses in this process. In the downstream task, the pre-trained parameters are loaded into the packet-level/flow-level attention modules, respectively. For classification, the loss is calculated between the prediction distribution obtained by mean pooling and then flatten of each patch feature and the truth label.

Existing encoder-based methods, including CETP [51], use extra self-supervised tasks during pre-training phase to enhance class-specific feature learning. During pre-training, CETP leverages a unique combination of positive-negative pair contrastive learning and a masked sequence model. The positive-negative pair model is instrumental in enhancing the discriminative power of the feature representations. Then, CETP employs semi-supervised fine-tuning and pseudo-label iterations to address imbalanced encrypted traffic data, achieving superior performance in imbalanced scenarios.

5.3.2. Pre-training based on transformer decoder

The Transformer decoder is a crucial component designed explicitly for generation tasks, processing sequences with a self-attentional mechanism, and generating outputs through a multi-layer structure. This mechanism allows each location to focus on the sequence when decoding, handle long-distance dependencies, and prevent future information leakage through masking. This makes the Transformer decoder suitable for sequence-to-sequence tasks such as traffic generation.

Specifically, the decoder consists of multiple layers, each containing a masked multi-head self-attention mechanism followed by a position-wise feed-forward network. The masking ensures that the model only attends to previous positions in the sequence, maintaining the autoregressive property required for generation tasks. Finally, a softmax layer calculates the probabilities of the next output token.

Based on this architecture, OpenAI developed GPT (Generative Pre-trained Transformer), an autoregressive language model. GPT-2, an upgraded version of the GPT family, is pre-trained with more network parameters and larger datasets to predict the next word based on existing text, resulting in a coherent text sequence. This model is particularly good at capturing semantic associations over long distances, producing more realistic text.

Meng et al. [58] proposed NetGPT, a generative pre-trained model for traffic characterization and generation tasks based on GPT-2. In the pre-processing stage, a unified text input is constructed through multi-mode network traffic modeling. It converts each traffic byte to its corresponding hexadecimal number and then uses a tokenizer to generate tokens. According to the learning mode of GPT-2, the pre-training stage predicts the next token according to the previous token in the input sequence to achieve the goal of a one-directional learning encrypted traffic context pattern.

The TrafficGPT proposed by Qu et al. [59] is equipped with linear attention mechanism, which supports up to 12032 input tokens, far surpassing the traditional 512-token limit of previous models. This enhancement addresses the challenge of processing long sequence data, which is typical in network traffic and beyond the processing capabilities of standard models.

Bikmukhamedov et al. [60] proposed a model based on the generative Transformer decoder, which can perform traffic generation and classification tasks. The model's input is the $[directedpacketlength, timestamp]$ matrix of the packet in the flow. The hard quantization method K-Means, which is suitable for large datasets, is adopted to process the input matrix to generate cluster embedding because the number of parameters in this method is small, and the position embedding and cluster embedding are added as inputs. The pre-training is performed for GPT-2.

5.3.3. Pre-training based on traditional deep learning model

In addition to Transformer and its variant models, some work is pre-trained based on traditional deep learning models such as CNN, LSTM, AE, etc. Liu et al. [64] proposed FewFine, which takes the processed fixed-length byte sequence originating from each session as input and selects 1D-CNN as the pre-trained model. The 1D-CNN consists of a feature extractor of multiple convolutional layers and a classification part including fully connected layers, whereas the feature extraction part consists of four basic blocks, each of which contains a convolutional layer, a batch normalization layer, a pooling layer, and a corresponding activation functions. The classification part consists of fully connected layers. The convolutional operation is used to capture traffic patterns. Then, the prior knowledge in the pre-trained model and a few new class samples are utilized to perform accurate malware detection and classification.

He et al. [63] proposed an anomaly detection model that combines CNN and AE to address few-shot learning challenges using minimal anomaly samples. The model starts by taking a traffic byte matrix as input and implements a channel attention mechanism at the end of the convolutional block through pooling operations [115], including maximum pooling, average pooling, and a combination of both. The processed output then passes through two CNN+attention layers and a fully connected layer for supervised classification. For unsupervised data reconstruction, features are fed into an autoencoder with three fully connected layers in both the encoder and decoder, assessing the reconstruction effects and ensuring discernible differences between normal and abnormal samples through convergence.

Since the current encrypted traffic pre-training classification approaches are still in development, it is reasonable to believe that there will be more architecture suitable for pre-training for researchers to mine in the future.

5.4. Fine-tuning stage

The fine-tuning phase is usually the last phase in the pre-training framework to adapt the already pre-trained model to a specific downstream task or domain. A notable trait of this stage is that rather than training the model from scratch, fine-tuning can be directly adapted from the pre-trained model to meet the needs of a specific scenario with a small amount of labeled data input. This process saves many computing resources and time and sometimes even improves the model's performance. Therefore, fine-tuning enables the model to gradually transition from "general" capabilities to "specialized " capabilities to adjust to the actual application scenario.

In the area of encrypted traffic classification and identification, fine-tuning is used to transform the pre-trained model of upstream traffic type independent into type dependent and output a task-specific representation of traffic for classification, which in turn makes the model perform well on scenarios such as general encrypted application traffic classification, encrypted malicious traffic classification, encrypted VPN traffic classification, and so on. Generally, fine-tuning structure is similar to pre-training, but some work involves setting tasks in the fine-tuning stage for efficient learning of biased traffic data. Meng et al. [58] proposed three significant tasks for NetGPT in the fine-tuning stage. These include shuffling header fields, segmenting packets in flows, and incorporating diverse tasks with different labels, which are used to improve the shortcomings of one-way learning of GPT-2 architecture, realize the learning of packet context, and improve the performance of traffic understanding and generation tasks by adding prompts to the header.

In order to effectively use a large number of unlabeled encrypted traffic to solve the problem of class imbalance, Lin et al. [55] used a pre-trained model to fine-tune the unbalanced traffic data and label the unlabeled traffic as pseudo label data. The availability of the traffic sample is then determined based on the appropriate threshold obtained from cross-validation, and continuous fine-tuning is performed to better identify unbalanced traffic.

Part of the work enables the pre-training framework to learn from various datasets by modifying hyperparameters during the fine-tuning phase. Lei et al. [55] proposed RP-BERT, a network intrusion detection and classification method combining transfer learning and rules. They improved ET-BERT in the fine-tuning stage with an approach named RNN Drop, which applies Bernoulli masks to LSTM hidden states and maintains the same mask across different data sequences. This technique helps prevent overfitting and enhances accuracy. The model, after fine-tuning, shows about an increase in accuracy over the original ET-BERT, effectively improving the performance of network intrusion detection.

5.5. Comparison of existing methods

The pre-trained model proposed by researchers shows excellent generalization performance in various scenarios in encrypted traffic. The results of several mainstream methods on commonly used public datasets are illustrated in Tables 14[2] and 15 to showcase the performance of existing pre-training methods more effectively.

In the work based on the Transformer encoder, PERT [49], as one of the earliest pre-trained encrypted traffic classification approaches, has proved the effectiveness of the pre-trained framework in the field of encrypted traffic on many datasets. However, this method also needs to improve. In other words, it lacks the specific design of encrypted traffic representation, which limits its generalization ability on the traffic of new encryption protocols, such as TLS 1.3, QUIC, etc. At the same time, it is easy to find that PERT performs poorly on the ISCX-Tor dataset, possibly because the number of flow samples in ISCX-Tor is too tiny, and PERT lacks specific pre-training tasks, which makes it challenging to learn diverse inter-packet associations from multi-layer encryption and adversarial confusion.

Compared with PERT, ET-BERT [50] not only adopts the traffic representation that can reflect the general communication pattern but also designs more pre-training tasks that are more conducive to learning traffic context structure. Therefore, ET-BERT achieves better results in five scenarios, including the application classification of new protocols on the low resource imbalance dataset. In addition to USTC-TFC, the classification effect of ET-BERT(packet) is better than that of ET-BERT(flow), because the task of identifying USTC-TFC datasets is relatively simple: the number of packets in the flow is large, and the encryption degree is not high so that ET-BERT can capture more information from the flow-level. In the remaining scenarios, the ET-BERT(packet) result is generally higher or closer to the ET-BERT(flow) result because ET-BERT can adapt to more fine-grained inputs. However, ET-BERT still lacks the ability to predict unknown class samples and toxic embedding attacks, which limits its application in broader scenarios such as open set identification and needs further improvement.

At the same time, compared with ET-BERT, CETP [51] with additional side channel features performs better on the dataset of ISCX-VPN APP. In addition to the improved pre-training task and semi-supervised settings in the fine-tuning stage, it also inspires us that the reasonable introduction of more feature information can provide more stable and abundant discriminant factors for the pre-training model.

YaTC [52] using MAE paradigm reaches F1-score above 0.965 in all four types of scenarios, which proves that it is feasible to apply a visual pre-training module to encrypted traffic classification. At the same time, the amount of parameters of this method is much smaller than that of the pre-training framework based on NLP mode (1.8M v.s. 132M compared to ET-BERT). However, unlike ET-BERT, YaTC's training set for fine-tuning and the data used for pre-training exactly coincide, which can lead to overfitting.

In the work based on the Transformer decoder, NetGPT is the first classification framework designed based on the generative pre-trained model. NetGPT can achieve 100% accuracy on ISCX-VPN-App datasets, proving that the generative pre-trained model has a good prospect in the task of encrypted traffic classification. However, some things could still be improved in this work. For instance, NetGPT achieved better performance with fewer training rounds, and the model performance decreased with increased training rounds. At the same time, the ablation study also proved that the performance of the complete model would be improved by 0.4% on average if the header field shuffling was not carried out in the fine-tuning stage. It is reasonable to speculate that

[2] According to their model requirements, the experimental settings of some methods in the table are slightly different, and the purpose of this table is to show the performance of mainstream pre-training methods for readers visually.

Table 14
Comparison results on ISCX-VPN-Service, ISCX-VPN-App, ISCX-Tor and USTC-TFC datasets.

Dataset	ISCX-VPN-Service				ISCX-VPN-App				ISCX-Tor				USTC-TFC			
Method	AC	PR	RC	F1	AC	PR	RC	F1	AC	PR	RC	F1	AC	PR	RC	F1
PERT [49]	0.9352	0.9400	0.9349	0.9368	0.8229	0.7092	0.7173	0.6992	0.7682	0.4424	0.4446	0.4345	0.9909	0.9911	0.9910	0.9911
ET−BERT(flow) [50]	0.9729	0.9756	0.9731	0.9733	0.8519	0.7508	0.7294	0.7306	0.8311	0.5564	0.6448	0.5886	0.9929	0.9930	0.9930	0.9930
ET−BERT(packet)	0.9890	0.9891	0.9890	0.9890	0.9962	0.9936	0.9938	0.9937	0.9921	0.9923	0.9921	0.9921	0.9915	0.9915	0.9916	0.9916
CETP [51]	–	–	–	–	0.8950	0.8402	0.8531	0.8416	–	–	–	–	–	–	–	–
YaTC [52]	0.9807	–	–	0.9804	–	–	–	–	0.9972	–	–	0.9972	0.9786	–	–	0.9786
NetGPT(flow) [58]	–	–	–	–	1.0000	–	–	1.0000	–	–	–	–	0.9575	–	–	0.9567
NetGPT(packet)	–	–	–	–	1.0000	–	–	1.0000	–	–	–	–	0.9563	–	–	0.9463
TrafficGPT(3K) [59]	–	–	–	–	0.9912	–	–	0.9912	–	–	–	–	0.9856	–	–	0.9854
TrafficGPT(12K)	–	–	–	–	1.0000	–	–	1.0000	–	–	–	–	0.9900	–	–	0.9877
BFCN [56]	0.9912	0.9913	0.9911	0.9911	0.9965	0.9936	0.9947	0.9941	–	–	–	–	–	–	–	–
TCMal [54]	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	0.9192
FewFine [64]	–	–	–	–	–	–	–	–	–	–	–	–	0.9711	–	–	–
SHAPE [62]	0.8607	–	–	0.8457	0.8196	–	–	0.6547	–	–	–	–	–	–	–	–
METC-MVAE [61]	–	–	–	–	–	–	–	–	0.9749	0.9145	0.9021	0.9082	–	–	–	–
DFBAN(known) [63]	–	–	–	–	–	–	–	–	–	–	–	–	–	0.9965	0.9983	0.9973
DFBAN(unknown)	–	–	–	–	–	–	–	–	–	–	–	–	–	0.9675	0.9552	0.9613

Table 15
Comparison results on CSTNET-TLS 1.3, Cross-Platform and CIC-IoT datasets.

Dataset	CSTNET-TLS 1.3				Cross-Platform(iOS)				Cross-Platform(Android)				CIC-IoT			
Method	AC	PR	RC	F1	AC	PR	RC	F1	AC	PR	RC	F1	AC	PR	RC	F1
PERT [49]	0.8915	0.8846	0.8719	0.8741	0.9789	0.9621	0.9611	0.9584	0.9772	0.8628	0.8591	0.8550	–	–	–	–
ET−BERT(flow) [50]	0.9510	0.9460	0.9419	0.9426	0.9844	0.9701	0.9632	0.9643	0.9865	0.9324	0.9266	0.9246	–	–	–	–
ET−BERT(packet)	0.9737	0.9742	0.9742	0.9741	0.9810	0.9757	0.9772	0.9754	0.9728	0.9439	0.9119	0.9206	0.9621	0.9427	0.9296	0.9321
CETP [51]	0.9618	0.9583	0.9501	0.9511	0.9813	0.9765	0.9727	0.9745	0.9882	0.9695	0.9635	0.9631	–	–	–	–
YaTC [52]	–	–	–	–	–	–	–	–	–	–	–	–	0.9658	–	–	0.9658
TrafficGPT(3K) [59]	–	–	–	–	0.9844	–	–	0.9829	0.9540	–	–	0.9483	0.9856	–	–	0.9854
TrafficGPT(12K)	–	–	–	–	0.9839	–	–	0.9863	0.9444	–	–	0.9498	0.9900	–	–	0.9877
METC-MVAE [61]	0.9514	0.9497	0.9324	0.9410	–	–	–	–	–	–	–	–	–	–	–	–

the header shuffling may cause the model to have difficulties processing the actual traffic data because the order and structure of the packet header fields in the actual traffic data are meaningful and relatively fixed. Furthermore, shuffling and segmentation increase the model's complexity, leading to overfitting and indicating that there is still room for improvement in the model design.

In the pre-training framework based on the traditional deep learning model, Liu et al. proposed FewFine [64], inspired by the fact that the features learned from lower-level convolutional layers in deep learning combine special and general features. They applied the pre-trained model to the new classification task. Specifically, the first K blocks are frozen to use prior knowledge. The remaining blocks are unfrozen and updated with a few samples to improve the accuracy of new tasks. When detecting new malware traffic, only one sample per class is fine-tuned and can achieve an accuracy 0.97. In the new classification, with ten samples per class fine-tuned, it can get an accuracy of 0.95. This result proves that the pre-trained model is a natural small sample learner, which is especially valuable in zero-day application identification. However, due to the hidden variability of malicious traffic, further correlations between different malicious flows need to be captured and integrated into the representation of traffic to identify its possible future variants accurately.

The model proposed by He et al. [63] is a few-shot detection architecture in the context of malicious traffic identification. The approach consists of two main modules: a CNN-based supervised pre-training module that utilizes a few abnormal samples for building deep feature extraction structures and an AE-based data reconstruction module that employs normal samples' deep features as training data. The model also incorporates effective attention mechanisms within the pre-training module to enhance the model's focus on relevant features. As shown in Table 14, the model can identify up to 99.7% of F1-score in known classes and more than 96% of F1-score in unknown classes.

The current pre-trained framework for encrypted traffic classification has demonstrated strong performance across various scenarios. It is evident that improving the adaptability of these models to real-world scenarios should be a priority in future research. Additionally, there is significant room for advancement in data characterization capabilities and the design of pre-training tasks.

6. Problems and opportunities

With the introduction of new protocols, the upgrading of obfuscation technology, and the increasing complexity of the network environment, the existing encrypted traffic classification technology has exposed many limitations. At the same time, as traffic classification methods based on pre-training represent an emerging technology, there remains significant room for exploration in the future.

In this section, we first introduce some defects and problems in the existing traffic classification work and then summarize the possible development direction and opportunities based on pre-training classification in the future.

6.1. Problems

In recent years, encrypted traffic classification in specific scenarios and tasks has achieved excellent performance. However, the existing work still needs to be revised in the real-world environment, for example, zero-day application identification, large-scale real-time traffic classification, etc. The problems that remain unresolved are given below.

6.1.1. Lack of convincing public datasets

In order to meet the diversified traffic types and multi-level analysis requirements caused by complex, realistic scenarios, researchers have proposed various types of datasets in the field of encrypted traffic analysis. Among them, the influential datasets include but are not limited to ISCXVPN2016 [65] for traffic classification that uses VPN, ISCXTOR2016 [66] for traffic classification on Onion Router (Tor), KDD Cup 99 [116] for intrusion detection, CIC MalMem [117] for malware

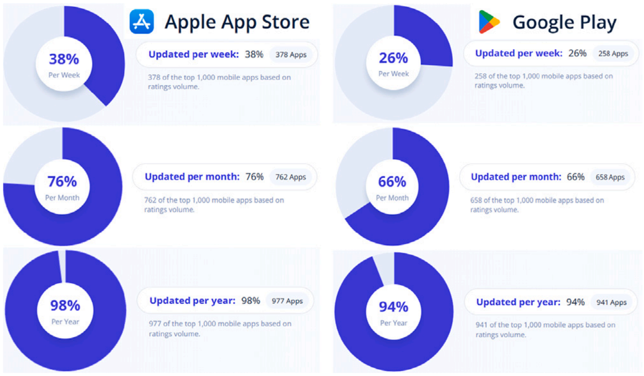


Fig. 10. Google Play and AppStore Top 1K app update frequency statistics.

classification, and Darknet 2020 [118] for dark web traffic classification, etc. However, the field still needs a comprehensive public dataset to uniformly measure the performance of models for professional use, such as ImageNet [119] in computer vision.

In addition, traffic generation and transmission modes vary in different scenarios; for instance, the frequency and rate of Voice over Internet Protocol (VoIP) communication data flows are high. As a result, the VoIP traffic in the ISCXVPN2016 dataset is much more than that in other categories [29]. If these unbalanced datasets are not adjusted appropriately, the model's performance will be significantly affected. Therefore, when using specific datasets, it is often necessary to solve the imbalance problem by data resampling (oversampling or undersampling), using unbalanced classification algorithms (such as integration methods), introducing cost-sensitive learning, sampling or modifying loss functions, and so on to improve the performance of the model.

6.1.2. Domain generalization under concept drift

Concept drift refers to the phenomenon that the statistical distribution of traffic data changes with changes in time, version, behavior, service [120], etc. Concept drift is one of the major problems facing the field of artificial intelligence today because traditional machine learning and deep learning models are trained based on the assumption of IID of training and test data. However, due to the complexity of the real environment, the training sample distribution is usually different from the test sample distribution, so the model often has high accuracy in the experimental scenario, and the effect is poor in practical application [121]. The statistical distribution of a specific type of traffic data changes frequently due to factors such as scenario requirements and protocol updates. As shown in Fig. 10, Google, for example, updates its existing service versions frequently in order to meet the new demands constantly put forward by users, among which the average update cycle of Google Play Top 100 applications is about 1–2 weeks [122]. Similarly, Jan et al. [123] found that the performance of their proposed deep learning classifier dropped dramatically over a one-month experimental period. According to the Certificate Transparency (CT) log, the size of the *.google.com certificate changed eight times from November 1, 2022, to February 28, 2023, which is speculated to be the reason for the above concept drift. Taking malicious traffic detection as an example, because malicious attacks are highly covert, attackers must constantly change the version of malicious software. As of 2022, more than 270,228 malware variants have been detected [124]. Traditional traffic classification methods may fail to identify traffic with concept drift accurately.

Domain generalization refers to extending a model from the trained domain (or data distribution) to an unknown new domain or data distribution when applying it [125]. In the context of encrypted traffic classification, domain generalization refers to applying a trained model to untrained new versions or dynamically changing network traffic while ensuring that the model achieves the desired generalization performance. Because the encrypted traffic generated by different platforms, regions and time shows great differences, and it is difficult to capture the invariant characteristics of these changing traffic, there are few domain generalization work in this field. The feasible strategy at present is to design an adaptive model [126], which iteratively updates training data and classifiers to adapt to dynamic changes in traffic. For example, Jorgensen et al. [127] proposed a traffic classification framework of a Prototypical Network based on wavelet and statistical features discrete sampled time windows. The framework is designed to detect OOD data by constructing an OOD score based on the p-value of the relative Mahalanobis distance of the in-distribution data. It can be trained quickly by the small data volume requirements of the Prototype Network to merge new traffic data into known categories or assign them to new categories. However, this approach is susceptible to adversarial machine learning [128,129].

In the future, researchers can consider the path of exploiting the general learning strategy, including ensemble learning, gradient operations, distributionally robust optimization [130–132], etc., to further explore the generalization of OOD traffic data.

6.1.3. Open-set problem

The open-set problem is the problem that must be solved by the encrypted traffic classification method in practice. That is, a classifier must be able to separate the traffic it has seen during the training phase (known known classes, KKCs) from all other traffic (unknown classes) [133]. Unlike domain adaptation and generalization issues, which assume that the training and testing domains share the same set of labels, the challenge of open-set recognition arises from a discrepancy in the label spaces: the label space in open-set scenarios extends beyond those seen during training.

Although there are partial approaches to solving the open-set problem, these approaches basically solve the problem either by adding background classes or binary classifiers to separate the known from the unknown, or by using classifier confidence thresholds to filter unknown traffic. These methods either divide large and diverse background traffic into one single unknown class, or ignore the fact that current deep learning models have high confidence even when they make mistakes [134].

In the future, the challenge of open-set can be studied from two distinct perspectives: firstly, traffic can be categorized based on its degree of unknown, ranging from unknown types but with known attribute (unknown known classes, UKCs), to unknown types with unknown attribute (unknown unknown classes, UUCs) among other types [135]. This categorization allows for a more nuanced approach, including the extraction of invariant features from known traffic to accurately identify the "known unknowns"; secondly, by examining the diverse characteristics of unknown traffic, it is possible to further subdivide background class, enabling more precise and detailed management of background traffic. The research can combine the hierarchical structure of data [119] or the hierarchical structure of the model [136] to further process UUCs by the way of hierarchical search to adapt to specific problems. This approach promises to enhance the granularity and accuracy of traffic classification systems.

6.1.4. More complex encryption protocols

Due to the further demand for security and efficiency of network encrypted communication, more robust encryption protocols such as TLS 1.3 and QUIC have emerged. Compared with TLS 1.2, TLS 1.3 supports 0-RTT data transmission, which makes the classification model based on the plaintext information of the TLS handshake phase significantly less effective. Since the pre-training-based approach does not require plaintext features for discrimination, it performs well on TLS 1.3. The model proposed by Lin et al. [50] realized the accurate classification of 120 types of mobile applications based on TLS 1.3

protocol by designing two tasks: Masked Burst Model and Same-origin Burst Prediction.

Another popular new encryption protocol is called QUIC. In contrast to the Transmission Control Protocol (TCP), QUIC multiplexes multiple requests/responses over a single connection by using a lightweight data structure called stream, where each request/response provides its stream ID so that the loss of a single packet blocks only the data flow in that packet and not other data flows in the same QUIC connection. This eliminates head-of-line blocking [137]. At present, some work [123, 138] has shown excellent results for the traffic classification under QUIC protocol. However, the fine-grained behavior identification in the same stream under a real environment is still to be solved.

6.1.5. Real-time and scalability

In real-world scenarios, encrypted traffic classification needs to be implemented in real-time or near real-time to classify and respond accurately, such as real-time threat detection, immediate response to critical events, QoS guarantee [139–141], network performance optimization, compliance, and supervision. The current research shows high recognition accuracy can be achieved in a small-scale network environment. However, with the continuous expansion of network traffic scale, the trend of full traffic encryption, and the wide application of private protocols and new encryption protocols, ensuring the efficiency, scalability, and low latency of classification methods is a significant challenge.

6.1.6. Homogenization in mobile network traffic

In the real world, because many mobile applications use the same third-party libraries and the same application-level protocol [142,143], and some content is hosted over CDNs or cloud providers, different apps share many network traffic characteristics [20], which makes it more difficult to accurately identify a specific application in the case of homogeneous traffic. Li et al. [25] propose a new app fingerprinting attack method called PacketPrint to identify user activity associated with applications of interest from encrypted wireless traffic. The sequential XGBoost and hierarchical bag of words models are proposed for application and behavior identification in homogeneous traffic. In the future, researchers can use the powerful learning ability of the pre-trained model to solve homogeneous traffic problems in more complex scenarios through data processing strategies suitable for real needs.

6.2. Opportunities

Nowadays, there are a certain number of encrypted traffic classification methods based on pre-training, but many areas still can be improved and unexplored. Some future opportunities for encrypted traffic classification methods based on pre-training are proposed in the following.

6.2.1. Pre-training security

Although the pre-trained model shows a strong generalization ability and wide versatility in traffic classification, it has high requirements for data purity. Attackers can achieve attacks by adding low-frequency sub-words to large-scale unlabeled training data for biased pre-trained model generation and directing classification results to specific categories in the downstream fine-tuning application stage [144,145]. Future researchers can study the robustness of models under biased data based on existing pre-trained models.

6.2.2. Pre-trained model computing resource requirements

Although pre-trained models have demonstrated their powerful capabilities in the field of encrypted traffic classification, they also have the problem of large resource requirements: they usually require a lot of computing resources and storage space to train and deploy. This may limit the broad application of the model, especially in devices or environments with limited computing power. How to maintain and improve the performance of the original model while reducing the resources required by the existing pre-trained model is a subject for future researchers to explore.

6.2.3. More diverse pre-training tasks

The task involved in existing traffic classification methods based on pre-training technology focuses on learning contextual pattern relationships related to masked byte information in large-scale traffic data. However, such tasks often require high hardware and software resources, so it is necessary to explore more efficient pre-training task. To this end, a feasible strategy is to transform the generating task into the discriminating task. The effectiveness of this method has been demonstrated in the literature [146], which can effectively reduce the amount of resources required in the pre-training stage.

6.2.4. Privacy-first pre-trained federated learning model

Since the pre-trained traffic classification model in a specific scenario needs distributed training, and the security of this part of traffic data must be guaranteed higher, such needs can be met by designing a privacy protection pre-trained model based on Federated Learning (FL). Federated Learning is a method that can train machine learning models collaboratively without disclosing privacy and keeping data dispersed [147]. The training and iteration of the distributed traffic pre-trained model are realized through global model building, local model training, parameter update sharing, global model aggregation, and cycle training and updating. At the same time, technologies such as differential privacy and secure multi-party computing can prevent unauthorized data access [148] in the entire process to ensure that traffic data privacy is protected.

6.2.5. Lower traffic data resources requirement

By inputting large-scale unlabeled traffic data, the pre-trained model can learn the implicit general pattern of encrypted traffic. Lin et al. [50] verified that the pre-trained model can perform well in generalization ability when only 10% of the training data is retained. In the future, researchers can further improve the generalization ability of the pre-trained model in a scenario with small sample size and realize the detection and identification of zero-day application traffic by grasping small samples and rapidly updating models.

6.2.6. Emulation traffic generation based on transformer decoder

Most existing pre-trained models in encrypted traffic use a Transformer encoder-based classification framework. However, in terms of traffic generation tasks, there is still some work [58–60], which uses a Transformer decoder structure to generate emulation traffic fragments. In the future, researchers can further improve the reality of generated traffic based on the excellent generation capability of the Transformer decoder according to the needs of realistic scenarios.

6.2.7. Traffic analysis based on generative large model

It has been shown that when the parameter scale of a language model is increased to a critical size (such as 10B), the model will emerge with abilities such as context learning, instruction following, and stepwise reasoning [149]. Recent generative large language models with billions of parameters, such as ChatGPT and LLaMa, have demonstrated surprising emergent capabilities. In addition to the field of natural language processing, the current generative large model has also profoundly changed the landscape of computer vision, law, autonomous driving, and other fields.

For the field of encrypted traffic classification and identification, the future feasible research direction covers the following three main aspects: first, through the super-large scale training, to propose a generative large model designed for the field of encrypted traffic in order to understand and analyze various types of encrypted communication modes more deeply; second, in the case of limited computing resources, how to fine-tune a general-purpose large-scale model with limited data to make it efficient while accurately performing traffic analysis. One possible solution is to bridge the gap between language representation and traffic representation by prompt tuning [150] to design specific prompts for different downstream tasks, such as malicious traffic identification, anomaly detection, etc.; finally, based on the excellent semantic understanding and generation ability of the large language model, the interpretable classification of encrypted traffic can be realized, so as to overcome the poor interpretation of the existing deep learning model, and the information provided by the large language model can be an important basis for further reliable classification and analysis. These three directions contain rich development opportunities for future encrypted traffic classification and identification research and are expected to drive cutting-edge innovation in the field.

6.2.8. Graph-based pre-trained and generative large model

As mentioned above, graph-based representation learning for encrypted traffic classification has performed excellently. However, the combination of this approach with pre-training techniques in the field is still limited. As proposed by Liu et al. [151], the graph foundation model is envisioned as a model that can be pre-trained on a wide range of graph data and used for various downstream graph tasks. Graph representation learning and pre-training in this field can be combined from three aspects: introducing self-attention thought into GNN, post-processing graph data input into LLM, and combining GNN with LLM. The first scheme is simple, and the parameter scale is small, yet the emergence ability is poor. The second scheme can realize the unified processing flow of traffic data in various scenes. However, finding the alignment strategy of natural language and encrypted traffic graph structure data is a severe problem. The last option requires researchers to consider balancing GNN and LLM to achieve efficient processing and low resource consumption. How to weigh the advantages and disadvantages of these three approaches and the possibility of exploring more graphs combined with pre-training and large models is a question that needs to be explored in the future.

7. Conclusion

Deep learning and pre-training technology offer great potential for revolutionizing the field of encrypted traffic classification. In this comprehensive review, we explore the evolving landscape in this area and highlight the pivotal role that deep learning and pre-training methodologies play in advancing traffic classification strategies. Our study begins with a detailed analysis of the current network environment and the critical need for advanced traffic classification strategies. We then delve into the mechanisms of various methods, analyzing various deep learning methods, such as graph representation learning, and exploring their model architecture, applicable scenarios, and characteristics. Furthermore, our discussion offers insightful comparisons of cutting-edge pre-training techniques and their implementation. We investigate the existing work from the three stages of the pre-training framework, namely pre-processing, pre-training, and fine-tuning, and reveal their unique advantages in the field. Finally, we put forward some existing problems and new research and exploration directions based on the needs of the current and future network environment.

In conclusion, our findings underscore the pivotal role of deep learning and pre-training techniques in revolutionizing encrypted traffic analysis, anticipating their growing influence in the dynamic landscape of network technology, while also highlighting the emergent trend of large models as a future trajectory. We anticipate that the synergy of these approaches — starting from the foundational deep learning techniques, advancing through the strategic use of pre-training, and culminating in the adoption of generative large models — will significantly influence the dynamic landscape of network technology. This integrated technological route, encompassing deep learning, pre-training, and generative large models, promises to address current challenges and propel the field forward. Therefore, we call for future research and innovation to explore this pathway further, recognizing its potential for groundbreaking developments in encrypted traffic analysis.

CRediT authorship contribution statement

Wenqi Dong: Writing – original draft, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Jing Yu:** Writing – review & editing, Supervision. **Xinjie Lin:** Writing – review & editing, Supervision, Investigation. **Gaopeng Gou:** Writing – review & editing, Supervision, Resources, Data curation. **Gang Xiong:** Supervision, Resources, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] Ericsson, Ericsson mobility report, 2023, [Online]. Available: https://www.ericsson.com/4ae12c/assets/local/reports-papers/mobility-report/documents/2023/ericsson-mobility-report-november-2023.pdf. (Accessed Novemeber 2023).

[2] F. Bi, T. He, X. Luo, A two-stream light graph convolution network-based latent factor model for accurate cloud service QoS estimation, in: Proceedings of the IEEE International Conference on Data Mining, ICDM, 2022.

[3] Google, HTTPS encryption on the web, 2024, [Online]. Available: https://transparencyreport.google.com/https/overview. (Accessed May 2024).

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the Conference on Neural Information Processing Systems, NeurIPS, 2017.

[5] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.

[6] A. Radford, K. Narasimhan, Improving language understanding by generative pre-training, 2018.

[7] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, 2019, OpenAI blog.

[8] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, in: Proceedings of the Conference on Neural Information Processing Systems, NeurIPS, 2020.

[9] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, Q.V. Le, XLNet: Generalized autoregressive pretraining for language understanding, in: Proceedings of the Conference on Neural Information Processing Systems, NeurIPS, 2019.

[10] H. Wang, J. Li, H. Wu, E. Hovy, Y. Sun, Pre-trained language models and their applications, Engineering 25 (2022) 51–65.

[11] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P.F. Christiano, J. Leike, R. Lowe, Training language models to follow instructions with human feedback, in: Proceedings of the Conference on Neural Information Processing Systems, NeurIPS, 2022.

[12] S. Rezaei, X. Liu, Deep learning for encrypted traffic classification: An overview, IEEE Commun. Mag. 57 (5) (2019) 76–81.

[13] P. Wang, X. Chen, F. Ye, Z. Sun, A survey of techniques for mobile service encrypted traffic classification using deep learning, IEEE Access 7 (2019) 54024–54033.

[14] X. Hu, C. Gu, Y. Chen, F. Wei, CBD: A deep-learning-based scheme for encrypted traffic classification with a general pre-training method, Sensors (Basel) 21 (24) (2021) 8231.

[15] D. Wu, X. Luo, M. Shang, Y. He, G. Wang, X. Wu, A data-characteristic-aware latent factor model for web services QoS prediction, IEEE Trans. Knowl. Data Eng. (TKDE) 34 (6) (2020) 2525–2538.

[16] Y. Guo, H. Li, J. Ding, Review and perspective on encrypted traffic identification using deep learning, Commun. Technol. 54 (9) (2021) 2074–2079.

[17] L. Chen, S. Gao, B. Liu, Z. Lu, Z. Jiang, THS-IDPC: A three-stage hierarchical sampling method based on improved density peaks clustering algorithm for encrypted malicious traffic detection, J. Supercomput. 76 (2020) 7489–7518.

[18] C. Fu, Q. Li, K. Xu, Detecting unknown encrypted malicious traffic in real time via flow interaction graph analysis, in: Proceedings of the Network and Distributed System Security Symposium, NDSS, 2023.

[19] M. Conti, L.V. Mancini, R. Spolaor, N.V. Verde, Analyzing android encrypted network traffic to identify user actions, IEEE Trans. Inform. Forensics Secur. (TIFS) 11 (1) (2015) 114–125.

[20] T. van Ede, R. Bortolameotti, A. Continella, J. Ren, D.J. Dubois, M. Lindorfer, D. Choffnes, M. van Steen, A. Peter, FlowPrint: Semi-supervised mobile-app fingerprinting on encrypted network traffic, in: Proceedings of the Network and Distributed System Security Symposium, NDSS, 2020.

[21] V.F. Taylor, R. Spolaor, M. Conti, I. Martinovic, AppScanner: Automatic fingerprinting of smartphone apps from encrypted network traffic, in: Proceedings of the IEEE European Symposium on Security and Privacy, EuroS&P, 2016.

[22] K. Al-Naami, S. Chandra, A. Mustafa, L. Khan, Z. Lin, K. Hamlen, B. Thuraisingham, Adaptive encrypted traffic fingerprinting with bi-directional dependence, in: Proceedings of the Annual Conference on Computer Security Applications, ACSAC, 2016.

[23] A. Panchenko, F. Lanze, J. Pennekamp, T. Engel, A. Zinnen, M. Henze, K. Wehrle, Website fingerprinting at internet scale, in: Proceedings of the Network and Distributed System Security Symposium, NDSS, 2016.

[24] J. Hayes, G. Danezis, k-fingerprinting: A robust scalable website fingerprinting technique, in: Proceedings of the USENIX Security Symposium, Security, 2016.

[25] J. Li, S. Wu, H. Zhou, X. Luo, T. Wang, Y. Liu, X. Ma, Packet-level open-world app fingerprinting on wireless traffic, in: Proceedings of the Network and Distributed System Security Symposium, NDSS, 2022.

[26] J. Li, H. Zhou, S. Wu, X. Luo, T. Wang, X. Zhan, X. Ma, FOAP:Fine-grained open-world android app fingerprinting, in: Proceedings of the USENIX Security Symposium, Security, 2022.

[27] Z. Tang, J. Wang, B. Yuan, H. Li, J. Zhang, H. Wang, Markov-GAN: Markov image enhancement method for malicious encrypted traffic classification, IET Inf. Secur. 16 (6) (2022) 442–458.

[28] L. Vu, C.T. Bui, Q.U. Nguyen, A deep learning based method for handling imbalanced problem in network traffic classification, in: Proceedings of the International Symposium on Information and Communication Technology, SoICT, 2017.

[29] W. Wang, M. Zhu, J. Wang, X. Zeng, Z. Yang, End-to-end encrypted traffic classification with one-dimensional convolution neural networks, in: Proceedings of the IEEE International Conference on Intelligence and Security Informatics, ISI, 2017.

[30] P. Sirinam, M. Imani, M. Juarez, M. Wright, Deep fingerprinting: Undermining website fingerprinting defenses with deep learning, in: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS, 2018.

[31] M. Lotfollahi, M. Jafari Siavoshani, R. Shirali Hossein Zade, M. Saberian, Deep packet: A novel approach for encrypted traffic classification using deep learning, Soft Comput. 24 (3) (2020) 1999–2012.

[32] P. Wang, F. Ye, X. Chen, Y. Qian, Datanet: Deep learning based encrypted network traffic classification in sdn home gateway, IEEE Access 6 (2018) 55380–55391.

[33] C. Liu, L. He, G. Xiong, Z. Cao, Z. Li, Fs-net: A flow sequence network for encrypted traffic classification, in: Proceedings of the IEEE International Conference on Computer Communications, INFOCOM, 2019.

[34] Z. Zhang, C. Kang, G. Xiong, Z. Li, Deep forest with LRRS feature for fine-grained website fingerprinting with encrypted SSL/TLS, in: Proceedings of the ACM International Conference on Information and Knowledge Management, CIKM, 2019.

[35] C. Rong, G. Gou, M. Cui, G. Xiong, Z. Li, L. Guo, TransNet: Unseen malware variants detection using deep transfer learning, in: Proceedings of the International Conference on Security and Privacy in Communication Systems, SecureComm, 2020.

[36] K. Lin, X. Xu, H. Gao, TSCRNN: A novel classification scheme of encrypted traffic based on flow spatiotemporal features for efficient management of IIoT, Comput. Netw. 190 (2021) 107974.

[37] M. Shen, Z. Gao, L. Zhu, K. Xu, Efficient fine-grained website fingerprinting via encrypted traffic analysis with deep learning, in: Proceedings of the IEEE/ACM International Symposium on Quality of Service, IWQoS, 2021.

[38] E. Horowicz, T. Shapira, Y. Shavitt, A few shots traffic classification with mini-FlowPic augmentations, in: Proceedings of the ACM Internet Measurement Conference, IMC, 2022.

[39] J. Guo, M. Cui, C. Hou, G. Gou, Z. Li, G. Xiong, C. Liu, Global-aware prototypical network for few-shot encrypted traffic classification, in: Proceedings of the IFIP International Conferences on Networking, Networking, 2022.

[40] N. Malekghaini, E. Akbari, M.A. Salahuddin, N. Limam, R. Boutaba, B. Mathieu, S. Moteau, S. Tuffin, Deep learning for encrypted traffic classification in the face of data drift: An empirical study, Comput. Netw. 225 (2023) 109648.

[41] M. Jiang, M. Cui, C. Liu, G. Gou, G. Xiong, Z. Li, Zero-relabelling mobile-app identification over drifted encrypted network traffic, Comput. Netw. 228 (2023) 109728.

[42] M. Jiang, Z. Li, P. Fu, W. Cai, M. Cui, G. Xiong, G. Gou, Accurate mobile-app fingerprinting using flow-level relationship with graph neural networks, Comput. Netw. 217 (2022) 109309.

[43] B. Sun, W. Yang, M. Yan, D. Wu, Y. Zhu, Z. Bai, An encrypted traffic classification method combining graph convolutional network and autoencoder, in: Proceedings of the IEEE International Performance Computing and Communications Conference, IPCCC, 2020.

[44] M. Shen, Y. Liu, L. Zhu, X. Du, J. Hu, Fine-grained webpage fingerprinting using only packet length information of encrypted traffic, IEEE Trans. Inform. Forensics Secur. (TIFS) 16 (2021) 2046–2059.

[45] T.-D. Pham, T.-L. Ho, T. Truong-Huu, T.-D. Cao, H.-L. Truong, Mappgraph: Mobile-app classification on encrypted network traffic using deep graph convolution neural networks, in: Proceedings of the Annual Conference on Computer Security Applications, ACSAC, 2021.

[46] R. Zhao, X. Deng, Y. Wang, L. Chen, M. Liu, Z. Xue, Y. Wang, Flow sequence-based anonymity network traffic identification with residual graph convolutional networks, in: Proceedings of the IEEE/ACM International Symposium on Quality of Service, IWQoS, 2022.

[47] F. Zola, L. Segurola-Gil, J.L. Bruse, M. Galar, R. Orduna-Urrutia, Network traffic analysis through node behaviour classification: a graph-based approach with temporal dissection and data-level preprocessing, Comput. Secur. 115 (2022) 102632.

[48] Z. Diao, G. Xie, X. Wang, R. Ren, X. Meng, G. Zhang, K. Xie, M. Qiao, EC-GCN: A encrypted traffic classification framework based on multi-scale graph convolution networks, Comput. Netw. 224 (2023) 109614.

[49] H. He, Z. Yang, X. Chen, PERT: Payload encoding representation from transformer for encrypted traffic classification, in: Proceedings of the ITU Kaleidoscope: Industry-Driven Digital Transformation, ITU K, 2020.

[50] X. Lin, G. Xiong, G. Gou, Z. Li, J. Shi, J. Yu, ET-BERT: A contextualized datagram representation with pre-training transformers for encrypted traffic classification, in: Proceedings of the International Conference on World Wide Web, WWW, 2022.

[51] X. Lin, L. He, G. Gou, J. Yu, Z. Guan, X. Li, J. Guo, G. Xiong, CETP: A novel semi-supervised framework based on contrastive pre-training for imbalanced encrypted traffic classification, Comput. Secur. 143 (2024) 103892.

[52] R. Zhao, M. Zhan, X. Deng, Y. Wang, Y. Wang, G. Gui, Z. Xue, Yet another traffic classifier: A masked autoencoder based traffic transformer with multi-level flow representation, in: Proceedings of the AAAI Conference on Artificial Intelligence, AAAI, 2023.

[53] P. Lin, K. Ye, Y. Hu, Y. Lin, C.-Z. Xu, A novel multimodal deep learning framework for encrypted traffic classification, IEEE/ACM Trans. Netw. 31 (2022) 1369–1384.

[54] M. Li, X. Song, J. Zhao, B. Cui, TCMal: A hybrid deep learning model for encrypted malicious traffic classification, in: Proceedings of the IEEE International Conference on Computer and Communications, ICCC, 2022.

[55] S. Lei, X. Zhang, J. Yi, RP-BERT: An approach to detect and classify network intrusions based on a combination of transfer learning and rules, in: Proceedings of the International Conference on Computer, Big Data and Artificial Intelligence, ICCBDAI, 2023.

[56] Z. Shi, N. Luktarhan, Y. Song, G. Tian, BFCN: A novel classification method of encrypted traffic based on BERT and CNN, Electronics 12 (3) (2023) 516.

[57] X. Hu, C. Gu, Y. Chen, F. Wei, CBD: A deep-learning-based scheme for encrypted traffic classification with a general pre-training method, Sensors 21 (24) (2021) 8231.

[58] X. Meng, C. Lin, Y. Wang, Y. Zhang, NetGPT: Generative pretrained transformer for network traffic, 2023, arXiv preprint arXiv:2304.09513.

[59] J. Qu, X. Ma, J. Li, Trafficgpt: Breaking the token barrier for efficient long traffic analysis and generation, 2024, arXiv preprint arXiv:2403.05822.

[60] R.F. Bikmukhamedov, A.F. Nadeev, Generative transformer framework for network traffic generation and classification, IEEE Trans. Commun. (T-Comm) 14 (11) (2020) 64–71.

[61] W. Cai, Z. Li, P. Fu, C. Hou, G. Xiong, G. Gou, METC-MVAE: Mobile encrypted traffic classification with masked variational autoencoders, in: Proceedings of the IEEE Int Conf on High Performance Computing & Communications; Int Conf on Data Science & Systems; Int Conf on Smart City; Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application, HPCC/DSS/SmartCity/DependSys, 2022.

[62] J. Dai, X. Xu, H. Gao, X. Wang, F. Xiao, SHAPE: A simultaneous header and payload encoding model for encrypted traffic classification, IEEE Trans. Netw. Serv. Manag. (TNSM) 20 (2) (2022) 1993–2012.

[63] M. He, X. Wang, J. Zhou, Y. Xi, L. Jin, X. Wang, Deep-feature-based autoencoder network for few-shot malicious traffic detection, Secur. Commun. Netw. 2021 (2021) 1–13.

[64] X. Liu, M. Shen, L. Cui, K. Ye, J. Jia, G. Yue, FewFine: Few-shot malware traffic classification via transfer learning based on fine-tuning strategy, in: Proceedings of the IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles, SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta, 2022, pp. 425–432.

[65] G. Draper-Gil, A.H. Lashkari, M.S.I. Mamun, A.A. Ghorbani, Characterization of encrypted and VPN traffic using time-related, in: Proceedings of the International Conference on Information Systems Security and Privacy, ICISSP, 2016.

[66] A.H. Lashkari, G.D. Gil, M.S.I. Mamun, A.A. Ghorbani, Characterization of tor traffic using time based features, in: Proceedings of the International Conference on Information Systems Security and Privacy, ICISSP, 2017.

[67] W. Wang, M. Zhu, X. Zeng, X. Ye, Y. Sheng, Malware traffic classification using convolutional neural network for representation learning, in: Proceedings of the IEEE International Conference on Information Networking, ICOIN, 2017.

[68] I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, et al., Toward generating a new intrusion detection dataset and intrusion traffic characterization, in: Proceedings of the International Conference on Information Systems Security and Privacy, ICISSP, 2018.

[69] G. Sebastian, P. Agustin, J.E. Maria, IoT-23: A labeled dataset with malicious and benign IoT network traffic (version 1.0.0) [Data set], 2020, Zenodo.

[70] J. Ren, M. Lindorfer, D.J. Dubois, A. Rao, D. Choffnes, N. Vallina-Rodriguez, A longitudinal study of PII leaks across android app versions, in: Proceedings of the Network and Distributed System Security Symposium, NDSS, 2018.

[71] Y. Wang, G. Xiong, C. Liu, Z. Li, M. Cui, G. Gou, CQNet: A clustering-based quadruplet network for decentralized application classification via encrypted traffic, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, ECML PKDD, 2021.

[72] T. Carrier, P. Victor, A. Tekeoglu, A.H. Lashkari, Detecting obfuscated malware using memory feature engineering, in: Proceedings of the International Conference on Information Systems Security and Privacy, ICISSP, 2022.

[73] S. Cui, J. Liu, C. Dong, Z. Lu, D. Du, Only header: A reliable encrypted traffic classification framework without privacy risk, Soft Comput. 26 (24) (2022) 13391–13403.

[74] D. Wu, Z. Li, Z. Yu, Y. He, X. Luo, Robust low-rank latent feature analysis for spatiotemporal signal recovery, IEEE Trans. Neural Netw. Learn. Syst. (TNNLS) (2023).

[75] M. Finsterbusch, C. Richter, E. Rocha, J.-A. Muller, K. Hanssgen, A survey of payload-based traffic classification approaches, IEEE Commun. Surv. Tutor. 16 (2) (2013) 1135–1156.

[76] W. Pan, G. Cheng, X. Guo, S. Huang, Review and perspective on encrypted traffic identification research, J. Commun. 37 (9) (2016) 154–167.

[77] Z. Cao, G. Xiong, Y. Zhao, Z. Li, L. Guo, A survey on encrypted traffic classification, in: Proceedings of the Applications and Techniques in Information Security, AITS, 2014.

[78] P. Velan, M. Cermák, P. Čeleda, M. Drasar, A survey of methods for encrypted traffic classification and analysis, Int. J. Netw. Manage. 25 (5) (2015) 355–374.

[79] A. Ankit, B. Ashutosh, B. Ayush, T. Kamlesh, H. K., V. Deepak, K. Rekha, A survey on analyzing encrypted network traffic of mobile devices, Int. J. Inf. Secur. 21 (4) (2022) 873–915.

[80] N. Alqudah, Q. Yaseen, Machine learning for traffic analysis: A review, Procedia Comput. Sci. 170 (2020) 911–916.

[81] E. Rodriguez, B. Otero, N. Gutierrez, R. Canal, A survey of deep learning techniques for cybersecurity in mobile networks, IEEE Commun. Surv. Tutor. 23 (3) (2021) 1920–1955.

[82] E. Papadogiannaki, S. Ioannidis, A survey on encrypted network traffic analysis applications, techniques, and countermeasures, ACM Comput. Surv. 54 (6) (2021) 1–35.

[83] M. Shen, K. Ye, X. Liu, L. Zhu, J. Kang, S. Yu, Q. Li, K. Xu, Machine learning-powered encrypted network traffic analysis: A comprehensive survey, IEEE Commun. Surv. Tutor. 25 (1) (2023) 791–824.

[84] Z. Wang, K.W. Fok, V.L. Thing, Machine learning for encrypted malicious traffic detection: Approaches, datasets and comparative study, Comput. Secur. 113 (2022) 102542.

[85] IANA, Service name and transport protocol port number registry, 2023, [Online]. Available: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml. (Accessed 20 Decemeber 2023).

[86] W. Niu, Z. Zhuo, X. Zhang, X. Du, G. Yang, M. Guizani, A heuristic statistical testing based approach for encrypted network traffic identification, IEEE Trans. Veh. Technol. (TVT) 68 (4) (2019) 3843–3853.

[87] S. Sen, O. Spatscheck, D. Wang, Accurate, scalable in-network identification of P2P traffic using application signatures, in: Proceedings of the International Conference on World Wide Web, WWW, 2004.

[88] A.W. Moore, K. Papagiannaki, Toward the accurate identification of network applications, in: Proceedings of the International Workshop on Passive and Active Network Measurement, PAM, 2005.

[89] A. Madhukar, C. Williamson, A longitudinal study of P2P traffic classification, in: Proceedings of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, MASCOTS, 2006.

[90] Zenarmor, How Deep Packet Inspection is changing the online world: DPI explained, 2023, [Online]. Available: https://www.zenarmor.com/docs/network-security-tutorials/what-is-deep-packet-inspection-dpi#do-firewalls-use-dpi-technology. (Accessed 21 Decemeber 2023).

[91] HongKe, Hongke sharing | what is Deep Packet Inspection (DPI)? (Chinese), 2022, [Online]. Available: https://zhuanlan.zhihu.com/p/572823255. (Accessed 21 Decemeber 2023).

[92] L. Hu, P. Hu, X. Luo, X. Yuan, Z.-H. You, Incorporating the coevolving information of substrates in predicting HIV-1 protease cleavage sites, IEEE/ACM Trans. Comput. Biol. Bioinform. 17 (6) (2019) 2017–2028.

[93] L. Hu, X. Pan, Z. Tang, X. Luo, A fast fuzzy clustering algorithm for complex networks via a generalized momentum method, IEEE Trans. Fuzzy Syst. 30 (9) (2021) 3473–3485.

[94] L. Hu, J. Zhang, X. Pan, X. Luo, H. Yuan, An effective link-based clustering algorithm for detecting overlapping protein complexes in protein-protein interaction networks, IEEE Trans. Netw. Sci. Eng. 8 (4) (2021) 3275–3289.

[95] X. Luo, H. Wu, Z. Li, NeuLFT: A novel approach to nonlinear canonical polyadic decomposition on high-dimensional incomplete tensors, IEEE Trans. Knowl. Data Eng. (TKDE) (2022).

[96] F. Bi, T. He, X. Luo, A fast nonnegative autoencoder-based approach to latent feature analysis on high-dimensional and incomplete data, IEEE Trans. Serv. Comput. (2023).

[97] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Proceedings of the Conference on Neural Information Processing Systems, NeurIPS, 2014.

[98] I.O. Lopes, D. Zou, I.H. Abdulqadder, S. Akbar, Z. Li, F. Ruambo, W. Pereira, Network intrusion detection based on the temporal convolutional model, Comput. Secur. 135 (2023) 103465.

[99] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2016, arXiv preprint arXiv:1609.02907.

[100] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: Proceedings of the International Conference on Learning Representations, ICLR, 2018.

[101] Y. Yuan, Q. He, X. Luo, M. Shang, A multilayered-and-randomized latent factor model for high-dimensional and sparse matrices, IEEE Trans. Big Data 8 (3) (2020) 784–794.

[102] H. Wu, X. Luo, M. Zhou, M.J. Rawa, K. Sedraoui, A. Albeshri, A PID-incorporated latent factorization of tensors approach to dynamically weighted directed network analysis, IEEE/CAA J. Autom. Sin. 9 (3) (2021) 533–546.

[103] D. Wu, P. Zhang, Y. He, X. Luo, MMLF: Multi-metric latent feature analysis for high-dimensional and incomplete data, IEEE Trans. Serv. Comput. (2023).

[104] X. Luo, H. Wu, Z. Wang, J. Wang, D. Meng, A novel approach to large-scale dynamically weighted directed network representation, IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) 44 (12) (2021) 9756–9773.

[105] Y. Xie, S. Li, C. Yang, R.C.-W. Wong, J. Han, When do GNNS work: Understanding and improving neighborhood aggregation, in: Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI, 2020.

[106] F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Trans. Neural Netw. (TNNLS) 20 (1) (2008) 61–80.

[107] G. Li, M. Muller, A. Thabet, B. Ghanem, Deepgcns: Can GCNS go as deep as CNNS? in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2019.

[108] M. Zhang, Z. Cui, M. Neumann, Y. Chen, An end-to-end deep learning architecture for graph classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, AAAI, 2018.

[109] H. Xu, S. Li, Z. Cheng, R. Qin, J. Xie, P. Sun, VT-GAT: A novel VPN encrypted traffic classification model based on graph attention neural network, in: Proceedings of the International Conference on Collaborative Computing: Networking, Applications and Worksharing, CollaborateCom, 2022.

[110] M. Shen, J. Zhang, L. Zhu, K. Xu, X. Du, Accurate decentralized application identification via encrypted traffic analysis using graph neural networks, IEEE Trans. Inform. Forensics Secur. (TIFS) 16 (2021) 2367–2380.

[111] R. Zhao, X. Deng, Y. Wang, L. Chen, M. Liu, Z. Xue, Y. Wang, Flow sequence-based anonymity network traffic identification with residual graph convolutional networks, in: Proceedings of the IEEE/ACM International Symposium on Quality of Service, IWQoS, 2022.

[112] L. Hu, S. Yang, X. Luo, M. Zhou, An algorithm of inductively identifying clusters from attributed graphs, IEEE Trans. Big Data 8 (2) (2020) 523–534.

[113] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R. Girshick, Masked autoencoders are scalable vision learners, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2022.

[114] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, in: Proceedings of the International Conference on Learning Representations, ICLR, 2020.

[115] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2018, pp. 7132–7141.

[116] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the KDD CUP 99 data set, in: Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA, 2009.

[117] T. Carrier, P. Victor, A. Tekeoglu, A.H. Lashkari, Detecting obfuscated malware using memory feature engineering, in: Proceedings of the International Conference on Information Systems Security and Privacy, ICISSP, 2022.

[118] A. Habibi Lashkari, G. Kaur, A. Rahali, DIDarknet: A contemporary approach to detect and characterize the darknet traffic using deep image learning, in: Proceedings of the International Conference on Communication and Network Security, ICCNS, 2021.

[119] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2009.

[120] M. Juarez, S. Afroz, G. Acar, C. Diaz, R. Greenstadt, A critical evaluation of website fingerprinting attacks, in: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS, 2014.

[121] J. Wang, C. Lan, C. Liu, Y. Ouyang, T. Qin, W. Lu, Y. Chen, W. Zeng, P.S. Yu, Generalizing to unseen domains: A survey on domain generalization, IEEE Trans. Knowl. Data Eng. (TKDE) 35 (8) (2023) 8052–8072.

[122] 42matters, App update frequency statistics 2023: Google Play Store ASO, 2023, [Online]. Available: https://42matters.com/google-play-aso-with-app-update-frequency-statistics. (Accessed 21 Decemeber 2023).

[123] J. Luxemburk, K. Hynek, T. Čejka, Encrypted traffic classification: The QUIC case, in: Proceedings of the Network Traffic Measurement and Analysis Conference, TMA, 2023.

[124] SonicWall, Latest SonicWall threat report uncovers seismic shift in cyber arms race due to geopolitical unrest as cyberattacks climb, 2022, https://www.sonicwall.com/news/latest-sonicwall-threat-report-uncovers-seismic-shift-in-cyber-arms-race-due-to-geopolitical-unrest-as-cyberattacks-climb/. (Accessed 21 Decemeber 2023).

[125] Z. Shen, J. Liu, Y. He, X. Zhang, R. Xu, H. Yu, P. Cui, Towards out-of-distribution generalization: A survey, 2021, arXiv preprint arXiv:2108.13624.

[126] R. Attarian, L. Abdi, S. Hashemi, AdaWFPA: Adaptive online website fingerprinting attack for tor anonymous network: A stream-wise paradigm, Comput. Commun. 148 (C) (2019) 74–85.

[127] S. Jorgensen, J. Holodnak, J. Dempsey, K.D. Souza, A. Raghunath, V. Rivet, N. DeMoes, A. Alejos, A. Wollaber, Extensible machine learning for encrypted network traffic application labeling via uncertainty quantification, IEEE Trans. Artif. Intell. (TAI) (2023) 1–15.

[128] M. Nasr, A. Bahramali, A. Houmansadr, Defeating DNN-Based traffic analysis systems in Real-Time with blind adversarial perturbations, in: Proceedings of the USENIX Security Symposium, Security, 2021.

[129] K. Bock, G. Hughey, X. Qiang, D. Levin, Geneva: Evolving censorship evasion strategies, in: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS, 2019.

[130] K. Zhou, Y. Yang, Y. Qiao, T. Xiang, Domain adaptive ensemble learning, IEEE Trans. Image Process. (TIP) 30 (2021) 8008–8018.

[131] Z. Huang, H. Wang, E.P. Xing, D. Huang, Self-challenging improves cross-domain generalization, in: Proceedings of the European Conference on Computer Vision, ECCV, 2020.

[132] S. Sagawa, P.W. Koh, T.B. Hashimoto, P. Liang, Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization, in: Proceedings of the International Conference on Learning Representations, ICLR, 2019.

[133] T. Dahanayaka, Y. Ginige, Y. Huang, G. Jourjon, S. Seneviratne, Robust open-set classification for encrypted traffic fingerprinting, Comput. Netw. 236 (2023) 109991.

[134] A. Bendale, T.E. Boult, Towards open set deep networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016.

[135] C. Geng, S.-j. Huang, S. Chen, Recent advances in open set recognition: A survey, IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) 43 (10) (2020) 3614–3631.

[136] Y. Qu, L. Lin, F. Shen, C. Lu, Y. Wu, Y. Xie, D. Tao, Joint hierarchical category structure learning and large-scale image classification, IEEE Trans. Image Process. (TIP) 26 (9) (2016) 4331–4346.

[137] Q. Zhang, C.-J. Su, Application-layer characterization and traffic analysis for encrypted QUIC transport protocol, in: Proceedings of the IEEE Conference on Communications and Network Security, CNS, 2023.

[138] V. Tong, H.A. Tran, S. Souihi, A. Mellouk, A novel QUIC traffic classifier based on convolutional neural networks, in: Proceedings of the IEEE Global Communications Conference, GLOBECOM, 2018.

[139] D. Wu, P. Zhang, Y. He, X. Luo, A double-space and double-norm ensembled latent factor model for highly accurate web service QoS prediction, IEEE Trans. Serv. Comput. 16 (2) (2022) 802–814.

[140] Y. Yuan, X. Luo, M. Shang, Z. Wang, A Kalman-filter-incorporated latent factor analysis model for temporally dynamic sparse data, IEEE Trans. Cybern. (2022).

[141] X. Luo, M. Chen, H. Wu, Z. Liu, H. Yuan, M. Zhou, Adjusting learning depth in nonnegative latent factorization of tensors for accurately modeling temporal patterns in dynamic QoS data, IEEE Trans. Autom. Sci. Eng. (TASE) 18 (4) (2021) 2142–2155.

[142] M. Backes, S. Bugiel, E. Derr, Reliable third-party library detection in android and its security applications, in: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS, 2016.

[143] A. Razaghpanah, A.A. Niaki, N. Vallina-Rodriguez, S. Sundaresan, J. Amann, P. Gill, Studying TLS usage in Android apps, in: Proceedings of the ACM International Conference on Emerging Networking EXperiments and Technologies, CoNEXT, 2017.

[144] N. Kassner, H. Schütze, Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, ACL, 2019.

[145] K. Kurita, P. Michel, G. Neubig, Weight poisoning attacks on pre-trained models, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, ACL, 2020.

[146] X. Han, Z. Zhang, N. Ding, Y. Gu, X. Liu, Y. Huo, J. Qiu, Y. Yao, A. Zhang, L. Zhang, et al., Pre-trained models: Past, present and future, AI Open 2 (2021) 225–250.

[147] Q. Zhang, L. Cheng, R. Boutaba, Cloud computing: State-of-the-art and research challenges, J. Int. Serv. Appl. (JISA) 1 (2010) 7–18.

[148] C. Dwork, A. Roth, et al., The algorithmic foundations of differential privacy, Found. Trends® Theoret. Comput. Sci. 9 (3–4) (2014) 211–407.

[149] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E.H.-h. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, W. Fedus, Emergent abilities of large language models, Trans. Mach. Learn. Res. (TMLR) 2022 (2022).

[150] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, J. Tang, GPT understands, too, 2021, arxiv prprint arXiv:2103.10385.

[151] J. Liu, C. Yang, Z. Lu, J. Chen, Y. Li, M. Zhang, T. Bai, Y. Fang, L. Sun, P.S. Yu, et al., Towards graph foundation models: A survey and beyond, 2023, arXiv preprint arXiv:2310.11829.