MVTBA: A Novel Hybrid Deep Learning Model for Encrypted Malicious Traffic Identification

Zuwei Fan[1,2] and Shunliang Zhang[1,2](✉)

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{fanzuwei,zhangshunliang}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

Abstract. Encryption technology protects data security and user privacy, but attackers can misuse it to evade detection techniques. To detect encrypted malicious traffic, deep learning based approaches attract increasing interest due to the manual feature engineering of conventional machine learning based methods. However, existing deep learning based approaches suffer from insufficient traffic representation, especially in fine-grained identification. To this end, this paper proposes a hybrid deep learning model MVTBA that can achieve remarkable traffic representation by automatically extracting spatial-temporal features without decryption. MVTBA consists of two sub-networks: MViT and BiLSTM-Att. The local-global spatial features are extracted by MViT through convolutions and an Unfold-Transformer-Fold structure of the mobile vision transformer block. The temporal features are extracted by BiLSTM with Attention to representing the timing dependence between traffic bytes. Subsequently, the two separated feature vectors are fused with an optimal weight factor to obtain the temporal-spatial features, which are fed into the classifier for encrypted malicious traffic identification. Extensive experimental results show that the accuracy of MVTBA in binary classification is improved to 99.99%. Moreover, MVTBA significantly outperforms other benchmark deep learning methods in fine-grained malicious identification, especially in the context of small data samples.

Keywords: Encrypted malicious traffic · Fine-grained identification · Deep learning

1 Introduction

With the increasing public awareness of privacy protection, encrypted transmission has been gradually replacing the original plaintext transmission. According to the Google Transparency Report [1], 97% of the world's top 100 non-Google sites provide HTTPS by default, and these 100 sites account for approximately 25% of all website traffic worldwide. It is a general trend to move towards

Supported by National Key Research and Development Project (2021YFB2910105).

H. Duan et al. (Eds.): SecureComm 2023, LNICST 568, pp. 60–79, 2025.
https://doi.org/10.1007/978-3-031-64954-7_4

the era of encryption. Encrypted traffic transmission can protect the privacy and security of users, while it may help the attackers evade secret monitoring. ZSCaler Report [4] shows a consistent upward trend of attacks using encrypted channels from 57% in 2020 to 80% in 2021, and more than 85% of attacks were encrypted in 2022. Therefore, how to effectively identify encrypted malicious traffic is of great significance for maintaining cyberspace security and resisting network attacks.

When the traffic is encrypted, the methods of decrypting the payload may consume a lot of resources and violate the privacy of users. It is important to identify encrypted malicious traffic without decryption. Port-based identification and traditional DPI [8] methods are no longer feasible. Therefore, machine learning (ML) is extensively investigated. ML can obtain rules from samples for inference and decision-making. However, traditional ML-based methods highly rely on expert-designed statistical features. In addition, the acquisition of higher-quality statistical features typically requires long-term traffic data collection, which increases storage costs and time consumption.

DL-based methods can automatically extract features from the raw traffic without manual feature engineering and discover non-intuitive connections between different traffic features. However, different DL model structures generate different traffic representations, resulting in different performances. Especially in fine-grained identification of encrypted malicious traffic, many DL-based methods based on a single feature can not fully mine traffic features and there is still room for improvement. How to build a great DL model that can generate effective traffic representation to improve identification accuracy is a problem worthy of research.

In addition, the encrypted malicious traffic in the real network environment is much less than the normal traffic. Due to the dynamic and concealment of malicious traffic, it's challenging to collect and label data samples. The resulting data samples are usually small and present an unbalanced number of encrypted malicious traffic data. Therefore, how to achieve accurate identification with limited samples is another problem to be solved.

In order to solve the above problems, we propose a hybrid deep learning model, MViT-BiLSTM Attention(MVTBA) for encrypted malicious traffic identification. MVTBA can achieve better performance by extracting both spatial and temporal features. The main contributions of this paper can be summarized as follows:

- Proposed a novel hybrid deep learning model MVTBA for encrypted malicious traffic identification, which extracts local-global spatial features by using Mobile Vision Transformer(MViT) block and extracts temporal features by using BiLSTM with an attention mechanism. The fused temporal-spatial feature can better represent the traffic and improve the accuracy of identification.
- The effectiveness of MVTBA is proved by the extensive ablation experiments with MViT and BiLSTM-Att used as variant models on three public datasets. The experimental results indicate that each sub-network contributes to improving the performance of MVTBA.

- The advantage of MVTBA is demonstrated by comparative experiments with LSTM, 1D-CNN [22], 2D-CNN [23], and CNN-LSTM [26] used as baseline models on four public datasets. The experimental results verify the superiority of MVTBA, especially in the context of small data samples.

The remainder of this paper is organized as follows. Related works are illustrated in Sect. 2. A detailed structure of the proposed model MVTBA is presented in Sect. 3. The effectiveness of the proposed mode is evaluated and elaborated via extensive experiments in Sect. 4. Finally, the paper is concluded in Sect. 5.

2 Related Work

In this section, we introduce the related research in encrypted malicious traffic identification and discuss their limitations. Encrypted malicious traffic identification methods mainly include rule-based, ML-based, and DL-based methods.

2.1 Rule-Based Methods

In the early stage, rule-based methods require manual analysis of the remaining plaintext in encrypted traffic to select distinctive components to construct fingerprints for fixed rule matching. Korczynski et al. [15] modeled the sequences of message types observed in single-directional SSL/TLS sessions to construct the fingerprints of different application traffic. However, the application fingerprints may change over time. FlowPrint [21] realizes the real-time construction of a mobile application fingerprint library through clustering, browser isolating, and cross-correlating the plaintext features of traffic, without requiring prior knowledge. However, these fingerprints are vulnerable to tampering and may lose their meaning. In summary, rule-based methods require a significant amount of manual labor and heavily rely on rule libraries, making them easily bypassed and resulting in a high false negative rate. As traffic encryption continues to advance, rule-based methods become more challenging.

2.2 ML-Based Methods

ML-based methods can effectively improve the accuracy of encrypted malicious traffic identification through feature engineering. In supervised learning, the work in [5,9] compared several commonly traditional ML methods, among which random forest(RF), decision tree, and XGBoost algorithms performed better. In particular, the robustness of RF is better [5], while the accuracy of XGBoost is higher [9]. In unsupervised learning, Chen et al. [7] proposed an improved density peaks clustering(IDPC) algorithm based on grid screening, custom center decision value, and mutual neighbor degree to effectively reduce the computational complexity and improve the clustering accuracy. Based on this, their three-stage sampling approach is carried out to improve the accuracy of encrypted malicious traffic identification. In semi-supervised learning, Liu et al. [18] combined the unsupervised Gaussian mixture model(GMM) and supervised XGBoost algorithm to achieve fine-grained identification of malware. ML-based methods have high interpretability, but they involve shallow learning of traffic features and require manual feature selection, resulting in limited generalization ability and the need for continuous updating.

2.3 DL-Based Methods

Deep learning can automatically extract features and generate representation from the raw data and has shown excellent performance in many fields, such as computer vision(CV), natural language processing(NLP), etc. In encrypted malicious traffic identification, DL-based methods take raw traffic or traffic features as input to automatically extract features and generate representations of traffic or find the hidden relationship between the features.

Convolutional Neural Networks(CNN) can effectively extract the spatial features of traffic. Wang et al. [22] proposed an end-to-end method of encrypted traffic classification with 1D-CNN, that directly takes the first 784 bytes of the original traffic as input, extracts spatial features, and outputs classification labels. Subsequently, they preprocessed the bytes into grayscale images as the input for a 2D-CNN [23] similar to LeNet-5 to identify malicious traffic. Based on this, Bazuhair et al. [6] improved the traffic representation by converting traffic features into grayscale images and enhancing them with Perlin noise. The above methods exclusively focus on spatial features, while neglecting the temporal information of traffic. As a result, when dealing with more complex traffic, there may be a decline in the identification performance [20].

Graph neural networks(GNN) is also effective in extracting spatial features of traffic. To detect malicious traffic, GCN-ETA [27] constructs a graph by creating edges when two flows share common IP, which may result in a very dense graph. ST-Graph [12] constructs a heterogeneous graph that correlates all network connections between hosts and servers. However, as the scale of the internal network expands, the graph structure becomes more complex, leading to heavier time costs. Hypervision [11] is a real-time unsupervised malicious traffic detection system. It constructs a flow interaction graph through short flow aggregation and feature distributing fitting for long flows, which reduces the time cost. Prographer [24] combines whole graph embedding and sequence learning to analyze snapshots of a provenance graph, which achieves unsupervised anomaly detection at the graph level. TFE-GNN [25] constructs a byte-level traffic graph by transforming the byte sequence of a flow while it may ignore the temporal information implied in byte sequences. GNN has strong potential in processing unstructured data. However, the construction and calculation cost of the graph is usually complicated and time-consuming.

RNN can effectively extract the temporal features of traffic. Gu et al. [13] utilized Multi-headed Attention and BiLSTM to obtain the message-level semantics, utilized LSTM to obtain the stream-level semantics, and then fused these two semantics to obtain the traffic representation. Experimental results showed that BiLSTM outperformed TextCNN and LSTM in encrypted malicious traffic detection. CNN-LSTM [26] is an integration of CNN and LSTM, which can directly extract deep features from the raw bytes for abnormal encrypted proxy traffic identification. HALNet [16] uses convolutional blocks to extract byte-level features, uses BiLSTM with Attention to extract global temporal features, and uses skipLSTM to extract local temporal features. Although the above methods have shown good performance, they do not fully use the spatial features of traffic, resulting in limited traffic representation.

Transformer provides new ideas for traffic identification, such as PERT [14] and ET-BERT [17]. MViT block [19] can extract local-global spatial features that contribute to achieving better classification performance. However, the transformer served as the foundation module for many large models usually requires a long training time, and is commonly used as a pre-training model.

The above literature mainly focuses on encrypted traffic classification, lacking comprehensive research on fine-grained identification of encrypted malicious traffic. Moreover, the single type of traffic feature is generally used for traffic representation, and less work is done on using fusion features. In this paper, both spatial and temporal traffic features can be extracted by MViT and BiLSTM with Attention. The fusion of features can improve the representation of encrypted malicious traffic for fine-grained identification, even in the context of limited data samples.

3 The Proposed Hybrid Deep Learning Model

The framework of the proposed hybrid deep learning model MVTBA is shown in Fig. 1. There are four parts of this framework, data pre-processing, automatic spatial feature extraction, automatic temporal feature extraction, temporal-spatial feature fusion and encrypted malicious traffic identification. In data pre-processing, the raw traffic is split into sessions which will be converted into 28×28 grayscale images or sequences of 784 bytes. In automatic spatial feature extraction, the local-global spatial features are extracted by MViT taking the images as input. In automatic temporal feature extraction, the temporal features are extracted by BiLSTM-Att taking the byte sequences as input. Finally, the above feature vectors will be fused with different weights, where the sum of two weights is equal to 1, resulting in the generation of the fused temporal-spatial features, which will be fed into the classifier for encrypted malicious traffic identification.

3.1 Data Pre-processing

Data pre-processing consists of traffic segmentation, data cleaning, length unification, and format conversion, realizing the transformation from raw traffic to images and byte sequences.

Traffic Segmentation: For each encrypted traffic Pcap file, the five-tuple (source IP, source port, destination IP, destination port, and transport protocol) is used as the basis for traffic split. To obtain more information, a session is selected as the split granularity instead of flow. Each traffic Pcap file is split into several sessions by SplitCap.

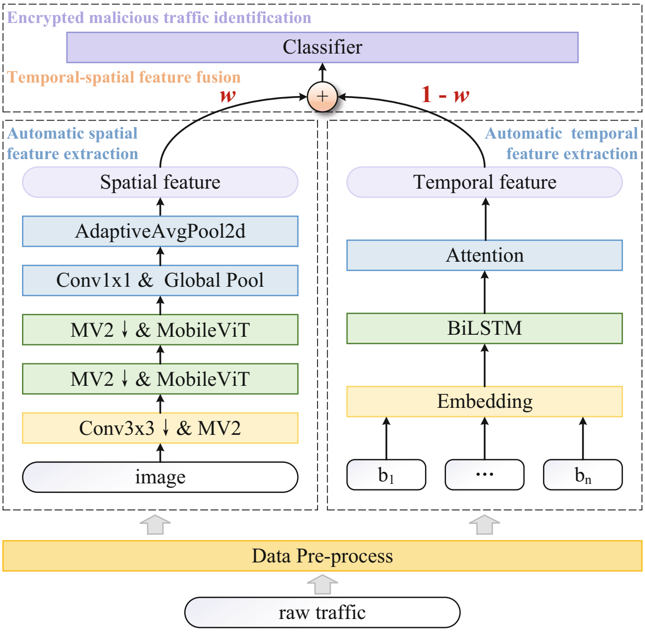


Fig. 1. Framework of MVTBA

Data Cleaning: The information that may bias the model should be randomized, such as MAC address and IP address. Duplicate or empty sessions should be deleted.

Length Unification: The session length is fixed to 784 bytes to meet the model input requirement, and the excess part will be trimmed, the part smaller than the session length will be filled with zero. This results in a byte sequence input for BiLSTM-Att.

Format Conversion: Each byte in the session has a size between 0 and 255, which corresponds to each pixel value of the grayscale image. Therefore, a session can be converted into a $28 \times 28 \times 1$ grayscale image.

3.2 Automatic Spatial Feature Extraction

CNNs have shown excellent capabilities in extracting spatial features, particularly in the fields of image recognition and computer vision. Various studies [6,22,23] have verified the feasibility of converting raw traffic data or traffic features into images and subsequently inputting images into models for spatial feature extraction to achieve accurate traffic classification. MobileViT [19] is a low-latency and lightweight network composed of CNN and vision transformer. It was released by Apple and achieved high accuracy on the ImageNet dataset.

MobileViT incorporates the spatial inductive biases of lightweight CNNs and the global representation learning ability of self-attention-based vision transformers to enhance network training stability and accelerate network convergence. Consequently, MobileViT can achieve better representations with fewer parameters. In image classification, MobileViT outperforms other lightweight CNNs such as MobileNetv1-v3, ShuffleNetv2, and ESPNetv2. Notably, it also delivers better performance than heavy-weight CNNs including ResNet, DenseNet, ResNet-SE, and EfficientNet. For instance, when compared to the best-performing model ResNet-SE, MobileViT improves accuracy by 0.8% while reducing the number of model parameters by 7.8 times. This highlights MobileViT's ability to achieve higher accuracy with fewer parameters.

Given the excellent performance of MobileViT in extracting spatial features from images, it can be transferred for traffic identification to extract the hidden spatial information from traffic images and effectively classify encrypted malicious traffic. In data preprocessing, we convert raw traffic into images. As our input (1×28×28) is much smaller than the input of the MobileViT (3×256×256), we redesigned a new MViT for traffic feature extraction. As shown in Fig. 2, the new MViT consists of four layers while the original MobileViT consists of six layers. They both consist of Convolutional (Conv), MobiletNetV2 (MV2), MobileViT blocks, and global pooling. The "↓" means stride = 2 and MV2 blocks are mainly responsible for down-sampling.

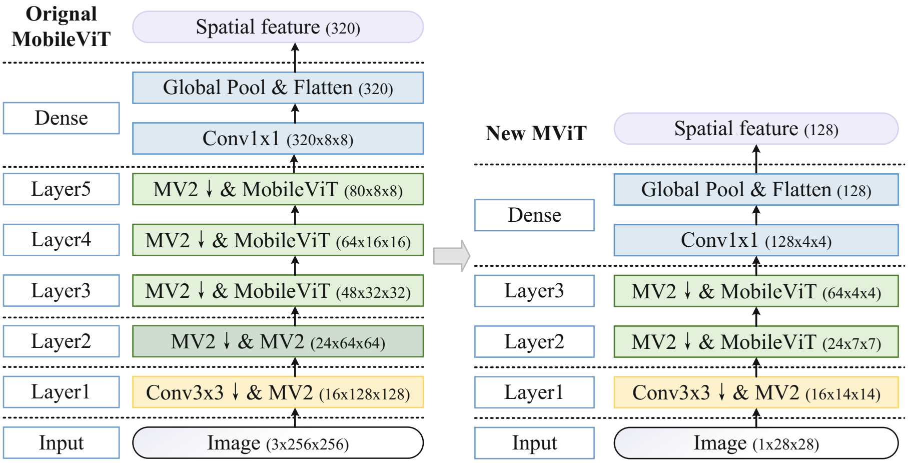

Orignal MobileViT
Spatial feature (320)
Dense
Global Pool & Flatten (320)
Conv1x1 (320x8x8)
Layer5
MV2 ↓ & MobileViT (80x8x8)
Layer4
MV2 ↓ & MobileViT (64x16x16)
Layer3
MV2 ↓ & MobileViT (48x32x32)
Layer2
MV2 ↓ & MV2 (24x64x64)
Layer1
Conv3x3 ↓ & MV2 (16x128x128)
Input
Image (3x256x256)
New MViT
Spatial feature (128)
Dense
Global Pool & Flatten (128)
Conv1x1 (128x4x4)
Layer3
MV2 ↓ & MobileViT (64x4x4)
Layer2
MV2 ↓ & MobileViT (24x7x7)
Layer1
Conv3x3 ↓ & MV2 (16x14x14)
Input
Image (1x28x28)

Fig. 2. Structure of original MobileViT and new MViT

The initial layer in new MViT and original MobileViT is a standard 3×3 convolution with stride = 2 to initially extract important information, followed by MV2 blocks and MobileViT blocks. In the MobileViT block, the convolution kernel size n is set as 3 and the spatial dimensions of the patch(height h and width w) are set as 2. The purpose of patch setting is to reduce the calculation of Self-Attention in Transform. In the new MViT, after two layers of MV2 blocks and MobileViT blocks, it is enough for the model to extract the traffic's local and global spatial information, and the output is $64 \times 4 \times 4$. Then, the output is adjusted to $128 \times 4 \times 4$ by a 1×1 Conv. Finally, the 128-dimensional spatial feature tensor is obtained through adaptive average pooling and flattening. By removing some MV2 and MobileViT blocks from the original model, the training time is effectively reduced, and the number of training parameters is reduced from 953,271 to 117,834.

The MobileViT block is the core and its specific structure is presented in Fig. 3. It consists of three parts: local representations, global representations, and local-global representations.

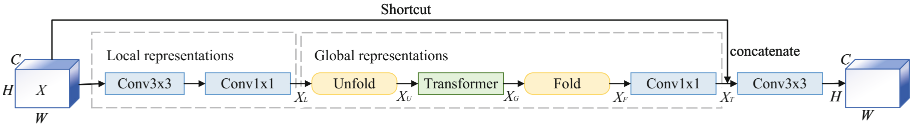


Fig. 3. MobileViT block

Local Representations. The MobileViT block first encodes local spatial information of the input tensor $X \in R^{H \times W \times C}$ by using a 3×3 depth-wise separable Conv while a 1×1 Conv adjusts channel numbers to produce $X_L \in R^{H \times W \times d}$ and prepare for subsequent global feature extraction.

Global Representations. Global spatial information is encoded through the Unfold-Transformer-Fold structure. Firstly, X_L is unfold into N non-overlapping flattened patches $X_U \in R^{P \times N \times d}$. Here, $P = wh$, $N = \frac{HW}{P}$ is the number of patches, $h \leq n$ is the height of a patch, and $w \leq n$ is the width of a patch. The inter-patch relationship $X_G \in R^{P \times N \times d}$ are obtained by encoding each $p \in \{1, \cdots, P\}$ through transformers:

$$X_G(p) = \text{Transformer}\left(X_U(p)\right), 1 \leq p \leq P \tag{1}$$

Dot products in the self-attention mechanism of the Transformer are performed only between pixels at the same position to reduce the computational cost. When the patch size is $2 \times 2 (h = 2, w = 2)$, the computational cost $= O(WHC/4)$, which is only 1/4 of the original cost. Then, $X_G \in R^{P \times N \times d}$ is fold to obtain $X_F \in R^{H \times W \times d}$. Channel numbers are restored through a 1×1 Conv to obtain the output $X_T \in R^{H \times W \times C}$ which has the same size of X.

Local-Global Spatial Representations. A shortcut branch concatenates the original input $X \in R^{H \times W \times C}$ with the current output $X_T \in R^{H \times W \times C}$ for increased richness and diversity of information. Finally, a 3×3 Conv is used to fuse these concatenated features, leading to the comprehensive local-global spatial features representation. Therefore, MViT can effectively extract the local-global spatial traffic features in the automatic spatial feature extraction.

3.3 Automatic Temporal Feature Extraction

The byte-packet-flow structure of the traffic is similar to the character-word-sentence structure of the text. Network traffic is a time-series sequence and its temporal features can be extracted by RNN. The effectiveness of BiLSTM in encrypted malicious traffic identification has been demonstrated in [13]. Unlike standard LSTM, which only considers current input and past information, BiLSTM is connected by forward and backward LSTMs to construct a better temporal traffic representation by considering both past and future states.

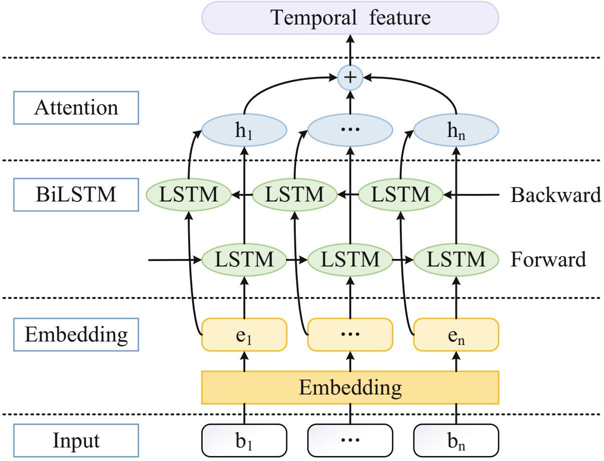


Fig. 4. Structure of temporal feature extraction

In this paper, the first 784 bytes of each session are used as the input sequence of BiLSTM-Att. As shown in Fig. 4, BiLSTM-Att comprises three components: embedding layer, BiLSTM layer, and attention layer. In the embedding layer, each byte is word-embedded into a 64-dimensional vector, so the first 784 bytes of the session are encoded into a dense vector of 784 × 64. After this, the temporal information can be extracted by BiLSTM, and then the more important part is weighted by the attention mechanism to obtain the temporal features. The size of the hidden layers is set as 64, so the 128-dimensional temporal feature tensor will be obtained.

3.4 Temporal-Spatial Feature Fusion and Traffic Identification

Local-global spatial features and temporal features have different impacts on identification. In this paper, the feature fusion formula is defined as follows:

$$F = w \times f_s + (1 - w) \times f_t \tag{2}$$

where F is the fused feature vector that will be fed into the classifier, f_s is the local-global spatial feature vector, f_t is the temporal feature vector, and w is a hyper parameter ranging from 0 to 1, which is used to adjust the influence of two features. The value of hyper parameter w is determined through experiments.

The classifier consists of a fully connected layer and a softmax layer. The fully connected layer takes over the fused features and converts the high-dimensional vector into a low-dimensional one(usually equal to the number of classes). After that, the probability of each class can be calculated by the softmax when the sum of the probabilities of all classes is 1. The softmax formula is defined as follows:

$$p_i = \frac{e^{z_i}}{\sum_{i=0}^{k} e^{z_i}} \tag{3}$$

where p_i is the probability of the input belonging to class i, z_i is the score, and k is the total number of classes.

4 Performance Evaluation

4.1 Datasets

In this paper, the following four public datasets are selected as the experimental datasets:

ISCX VPN-nonVPN [10] dataset is designed for encrypted application and service traffic classification, including 7 regular and 7 VPN encrypted traffic. Traffic types include Email, Chat, Streaming, File Transfer, VoIP, P2P, and Browsing. In this paper, 9 kinds of traffic are selected to form a normal encrypted traffic dataset, including VPN and nonVPN traffic, as shown in Table 1.

Table 1. ISCX VPN-nonVPN dataset

Type	Encryption	Content	Train	Test
Chat	VPN	facebook_chat	927	232
File transfer	VPN	ftps, sftp, skype_file	805	201
Streaming	VPN	hangouts_audio	2538	634
VoIP	VPN	voipbuster	1294	324
Email	nonVPN	email	2798	699
Streaming	nonVPN	hangouts_audio	1384	346
Chat	nonVPN	skype_chat	3542	886
P2P	nonVPN	Torrent	836	209
VoIP	nonVPN	voipbuster	1420	355

Malware-traffic-analysis (MTA) [3] dataset consists of encrypted malicious traffic provided by malware-traffic-analysis.net since 2013. We chose traffic files from February 2020 to February 2023 and selected seven popular malware families, including Dridex, Emotet, Hancitor, Icedid, Qakbot, Trickbot, and Ursnif, as shown in Table 2.

Table 2. MTA dataset

Type	Dridex	Emotet	Hancitor	Icedid	Qakbot	Trickbot	Ursnif
Train	492	3368	13452	1454	3350	1794	506
Test	123	842	3363	364	838	448	127

Malware Capture Facility Project (MFCP) [2] dataset consists of encrypted malicious traffic collected by the Malware Capture Facility Project conducted by Stratosphere IPS. We selected 6 popular malware families including Artemis, Cobalt, Dridex, PUA, TrickBot, and Ursnif, as shown in Table 3. Due to an excessive number of sessions contained in certain raw Pcap files, we trimmed some of the traffic to expedite the experimental process.

Table 3. MFCP dataset

Type	Artemis	Cobalt	Dridex	PUA	TrickBot	Ursnif
CTU_Num	316-1	345-1	326-1	335-1,337-1	327-1	313-1
Train	6000	1501	6000	5614	6000	6000
Test	1500	375	1500	1403	1500	1500

USTC [23] dataset collected by the University of Science and Technology of China includes 10 normal and 10 malicious encrypted traffic. We conducted experiments using only the malicious traffic subset, as shown in Table 4. Due to the large scale and diverse categories of the USTC dataset, we chose to evaluate the identification performance of models with reduced training sample sizes on it for better comparative results.

Table 4. USTC dataset

Type	Cridex	Geodo	Htbot	Miuref	Neris	Nsis-ay	Shifu	Tinba	Virut	Zeus
CTU_Num	108-1	119-2	110-1	127-1	42,43	53	142-1	150-1	54	116-2
Train	13109	32758	5094	10785	27033	4855	7707	6803	26482	8776
Test	3277	8189	1273	2696	6758	1214	1927	1701	6621	2194

4.2 Evaluation Metrics

To evaluate the performance of MVTBA, we use accuracy, precision, recall, and F1-score as evaluation metrics. The metrics are defined as follows:

$$Accuarcy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$Precision_i = \frac{TP_i}{TP_i + FP_i}, macroP = \frac{1}{n}\sum_{i=1}^{n} Precision_i \tag{5}$$

$$Recall_i = \frac{TP_i}{TP_i + FN_i}, macroR = \frac{1}{n}\sum_{i=1}^{n} Rrecall_i \tag{6}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}, macroF1 = \frac{2 \times macroP \times macroR}{macroP + macroR} \tag{7}$$

In (4)–(5), TP_i is the number of correctly identified i-class traffic, FP_i is the number of i-class traffic identified as non-i-class traffic, TN_i is the number of correctly identified non-i-class traffic and FN_i is the number of non-i-class traffic identified as i-class traffic.

4.3 Experiment Design

The paper designs three groups of experiments, namely:

Exp. I: Mix ISCX VPN-nonVPN, MTA, and MCFP to generate an encrypted abnormal traffic identification dataset with 9 normal traffic and 9 abnormal traffic. Conduct abnormal traffic identification experiments on the mixed dataset. The main purpose is to evaluate the performance of MVTBA on abnormal encrypted traffic identification. LSTM, 1D-CNN [22], 2D-CNN [23], CNN-LSTM [26], MViT, and BiLSTM-Att are selected as baseline models.

Exp. II: We conducted 7-class identification experiments on the MTA dataset and 6-class identification experiments on the MCFP dataset. The max sample size in MTA is 16815 for hancitor while the min size is 615 for dridex, which can represent the case of data imbalance. The main purpose is to evaluate the performance of models on fine-grained encrypted malicious traffic identification. LSTM, 1D-CNN [22], 2D-CNN [23], CNN-LSTM [26], MViT, and BiLSTM-Att are selected as baseline models.

Exp. III: Malicious traffic in the real network environment is rare. Therefore, in addition to conducting experiments of fine-grained malicious traffic identification on the original USTC dataset, we also limited the number of samples in the training set to 4000, 3000, and 2000 respectively, while keeping the testing set unchanged. The main purpose is to evaluate the performance of models in the context of small data samples. LSTM, 1D-CNN [22], and 2D-CNN [23] are selected as baseline models.

The experiments above are based on Pytorch, the training epoch is 30, the batch size is 64, and the optimizer is Adam with a learning rate of 0.001. The experimental environment is under the Windows system, with Intel(R) Core(TM) i5-7300HQ CPU @2.50 GHz, 8G RAM, and GeForce RTX 1050.

4.4 Hyper Parameter Selection

The hyper parameter w we mentioned above represents the weight of spatial features in the fused temporal-spatial features. We consider that if w is too close to 1, the temporal information learned by the model will be not enough, while if w is too close to 0, the spatial information learned by the model will be not enough. In this subsection,w is set to 0, 0.2, 0.4, 0.6, 0.8, and 1 to train the model on the sample imbalanced MTA dataset, which is close to the real network environment. When $w = 1$, the model is MViT, and when $w = 0$, it is BiLSTM-Att. As shown in Fig. 5, MVTBA has the best performance when $w = 0.6$. The reason why MViT performs slightly worse than BiLSTM-Att is the limited data size of the MTA dataset, which hampers the potential advantage of ViT. When $w = 0.2$ or $w = 0.8$, the imbalanced fusion of two features may destroy the balance of the original model, leading to a decrease in accuracy. When the fusion of both types of information is relatively balanced, MVTBA performs well in extracting temporal-spatial features, especially with $w = 0.6$. Therefore, we choose $w = 0.6$ to guarantee the performance of MVTBA.

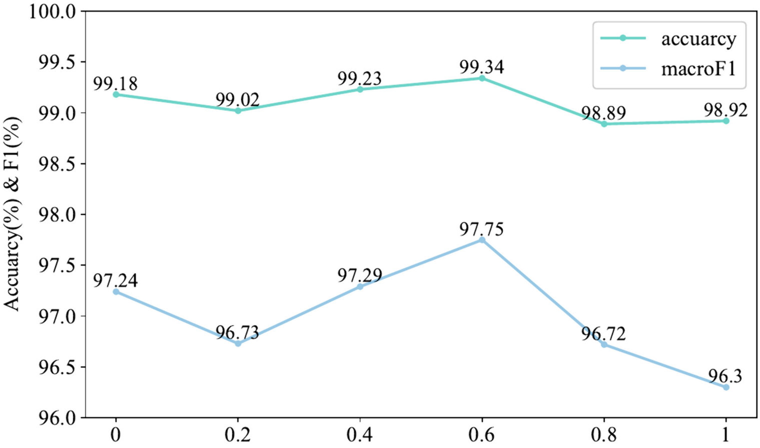


Fig. 5. Accuracy and macroF1 of MVTBA with different values of w

4.5 Experiment Result Analysis and Discussion

(1) Performance on Abnormal Encrypted Traffic Identification: Table 5 shows the identification accuracy of Exp.I is quite high. The accuracy rate of deep learning models has reached more than 99% and the highest accuracy of MVTBA reaches 99.99%. The highest recall of MVTBA reaches 100%. Exp. I proves that DL-based methods can effectively identify normal and abnormal encrypted traffic. However, real-world scenarios also require fine-grained identification of malicious traffic to provide accurate defense measures. Next, we conduct a fine-grained identification experiment of encrypted malicious traffic.

Table 5. Metrics on EXP.I

Model	Accuracy(%)	Precision (%)	Recall (%)	F1 (%)
1D-CNN	99.96	99.98	99.94	99.96
2D-CNN	99.92	99.96	99.91	99.94
LSTM	99.98	99.98	99.98	99.98
CNN-LSTM	99.94	99.96	99.93	99.94
BiLSTM-Att	99.95	99.94	99.96	99.95
MViT	99.96	99.98	99.94	99.96
MVTBA	**99.99**	**99.98**	**1**	**99.99**

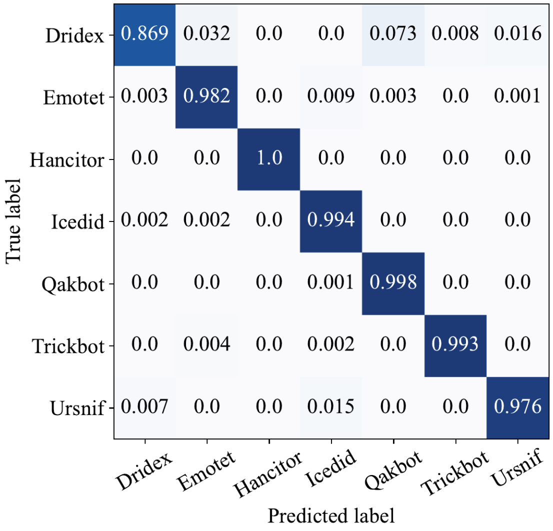
(2) Performance on Fine-Grained Encrypted Malicious Traffic Identification: The evaluation results on the MTA dataset are shown in Table 6. On this dataset, MVTBA is the best-performing model. Taking macroF1 as an example, MVTBA achieves a score of 97.75%, which is 2.17%, 2.72%, and 13.36 % higher than that of 1D-CNN, 2D-CNN, and LSTM respectively. It can be stated that the performance of MVTBA is far superior to 1D-CNN, 2D-CNN, and LSTM. The macroF1 of 1D-CNN is 0.55% higher than that of 2D-CNN, suggesting that changing 1D-CNN to 2D-CNN does not improve the performance of encrypted malicious traffic identification. The macroF1 of BiLSTM-Att is 12.85% higher than that of LSTM, which proves the effectiveness of BiLSTM with Attention in extracting comprehensive temporal features. The accuracy of MViT is better than that of 1D-CNN and 2D-CNN, which proves the effectiveness of Vision Transformer in extracting local-global spatial features.

Table 6. Metrics on MTA of EXP.II

Model	Accuarcy (%)	macroP (%)	macroR (%)	macroF1 (%)
1D-CNN	98.48	95.8	95.46	95.58
2D-CNN	98.39	94.88	95.2	95.03
LSTM	95.14	88.84	82.00	84.39
CNN-LSTM	95.69	87.33	84.92	86.01
BiLSTMAtt	99.18	97.75	96.77	97.24
MViT	98.92	96.98	95.75	96.3
MVTBA	**99.34**	**98.21**	**97.36**	**97.75**

In addition, Fig. 6 presents the confusion matrix of MVTBA after 30-round finetuning on the MTA dataset. Meanwhile, Fig. 7 illustrates the confusion matrix of baseline models with identical settings for comparison.

It is evident that MVTBA achieves the highest identification accuracy of all seven malicious families. It can be observed that the identification accuracy of



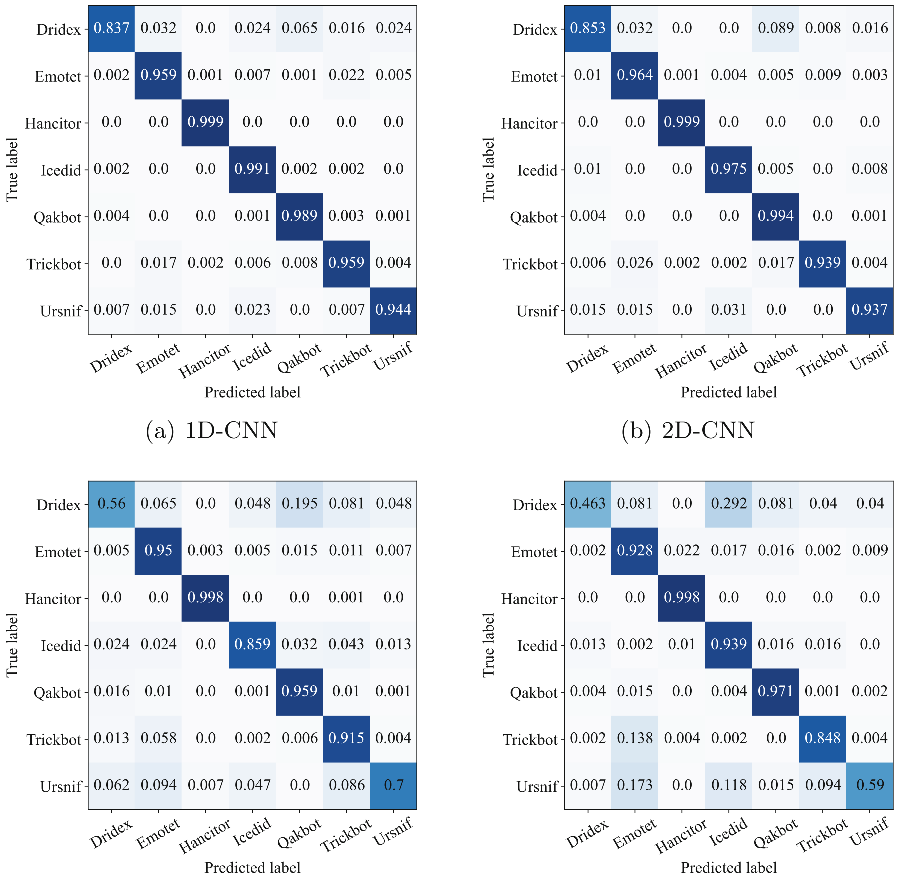
True label \ Predicted label	Dridex	Emotet	Hancitor	Icedid	Qakbot	Trickbot	Ursnif
Dridex	0.869	0.032	0.0	0.0	0.073	0.008	0.016
Emotet	0.003	0.982	0.0	0.009	0.003	0.0	0.001
Hancitor	0.0	0.0	1.0	0.0	0.0	0.0	0.0
Icedid	0.002	0.002	0.0	0.994	0.0	0.0	0.0
Qakbot	0.0	0.0	0.0	0.001	0.998	0.0	0.0
Trickbot	0.0	0.004	0.0	0.002	0.0	0.993	0.0
Ursnif	0.007	0.0	0.0	0.015	0.0	0.0	0.976

Fig. 6. Confusion matrix of MVTBA

models on Driex and Ursnif is relatively low, which can be attributed to their minimal number of samples (615 and 633). On Hancitor with the largest sample size, all models exhibit significant improvements in accuracy, with MVTBA achieving a perfect accuracy of 100%. Even on Driex with the smallest sample size, MVTBA can still obtain a comprehensive feature representation with the highest accuracy of 86.9%, which is respectively 3.2%, 1.6%, 30.9%, and 40.6% higher than that of 1D-CNN, 2D-CNN, CNN-LSTM, and LSTM. Such consistent and prominent performance demonstrates the effectiveness of MVTBA in the context of imbalanced encrypted malicious traffic identification.

The metrics on the MCFP dataset are shown in Table 7. MVTBA continues to exhibit the best performance among these models, achieving the highest accuracy of 98.96% and the highest macroF1 of 97.84%. The macroF1 of MVTBA is respectively 0.88%, 1.09%, and 1.64% higher than that of 1D-CNN, 2D-CNN, and LSTM. The performance of BiLSTM-Att still outperforms LSTM. As the sample size increases, the performance of MViT is better than that of BiLSTM-Att. As a widely adopted architecture for pre-training models, the performance of transformer is remarkable when dealing with large amounts of data.

(3) Performance in the Context of Small Data Samples: The accuracy rates on the USTC dataset are shown in Fig. 8. When the training samples consist of 10 classes, with a total quantity of 143402, MVTBA achieves the highest accuracy of 99.99%, which is respectively 0.23%, 1.48%, and 0.33%higher than that of 1D-CNN, 2D-CNN, and LSTM. It proves the excellence of MVTBA in fine-grained encrypted malicious traffic identification. Furthermore, it can be observed that the reduction of encrypted malicious traffic has a certain impact



(a) 1D-CNN

(b) 2D-CNN

(c) CNN-LSTM

(d) LSTM

Fig. 7. Confusion matrix of baseline models

Table 7. Metrics on MFCP of EXP.II

Model	Accuarcy (%)	macroP (%)	macroR (%)	macroF1 (%)
1D-CNN	98.52	97.08	96.85	96.96
2D-CNN	98.37	96.74	96.75	96.75
LSTM	98.06	95.61	96.97	96.2
CNN-LSTM	98.34	96.23	97.15	96.66
BiLSTMAtt	98.62	96.85	97.51	97.17
MViT	98.65	96.71	97.55	97.28
MVTBA	**98.96**	**97.84**	**97.84**	**97.84**

on classification accuracy. When the sample size is 4000, 2D-CNN proved to be the worst model, with the largest decrease in accuracy, dropping by 7.02%. Meanwhile, the accuracy of MVTBA is 94.73% and respectively 0.82%, 3.32%, and 2.14% higher than that of 1D-CNN, 2D-CNN, and LSTM. When the sample size is reduced to 3000, the accuracy of MVTBA is 94.46% and respectively 3.16%, 3.48%, and 2.07% higher than that of 1D-CNN, 2D-CNN, and LSTM. As the sample size further drops to 2000, only MVTBA manages to maintain an accuracy rate above 92%, while the accuracy of the other models is lower than 90%. These results indicate that MVTBA can still achieve good identification performance even in the context of small data samples.

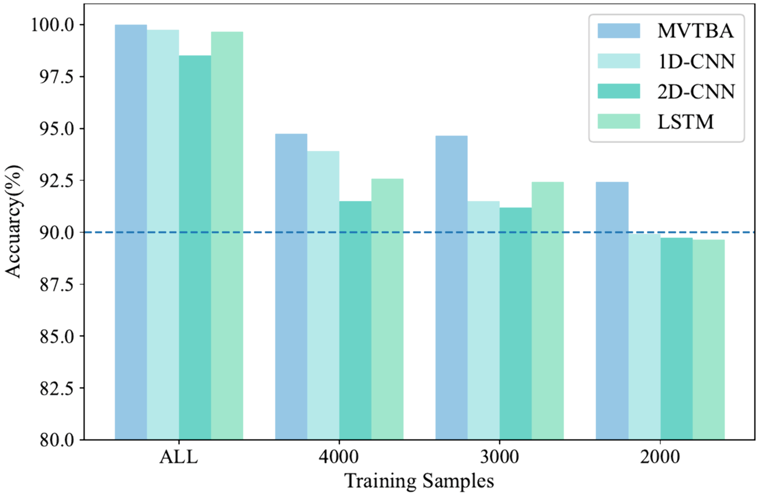


Fig. 8. Accuarcy of MVTBA, 1D-CNN, 2D-CNN, LSTM

(4) Comparison of Model Training Parameters and Size: The training parameters and size of MVTBA and other baseline models are shown in Table 8. MVTBA demonstrates a significantly smaller number of model training parameters and size compared to 1D-CNN and 2D-CNN. And the model size of MVTBA is close to that of LSTM, which has the smallest size among the baseline models. The training parameters of MVTBA are only 299,977, occupying a small amount of storage space.

Table 8. Comparison of model training parameters and size

Model	Trainable params	Params size (MB)
1D-CNN	5,826,438	22.23
2D-CNN	3,239,623	12.36
MVTBA	**299,977**	**1.14**
CNN-LSTM	243,330	0.929
LSTM	213,895	0.816

5 Conclusion

In this paper, we proposed a hybrid deep learning model MVTBA for encrypted malicious traffic identification. MVTBA can automatically extract spatial and temporal features, which accurately represent encrypted traffics. Local-global spatial features are extracted by the mobile vision transformer block, and temporal features are extracted by BiLSTM with Attention. Extensive experiments based on four datasets are conducted. The experimental results demonstrate that MVTBA can accurately identify different encrypted malicious traffic, even in the context of small data samples. Moreover, it outperforms the baseline models in terms of accuracy, precision, recall, and F1-score. As potential future work, more effort is needed to reduce the training time and improve the detection speed to achieve real-time and accurate identification of encrypted malicious traffic. In addition, it is also necessary to deploy and evaluate the performance of MVTBA in real network environments.

References

1. Google Transparency Report. https://transparencyreport.google.com/https/overview. Accessed 18 June 2023
2. Malware Capture Facility Project. https://www.stratosphereips.org/datasets-malware. Accessed 18 June 2023
3. Malware-traffic-analysis.net. https://malware-traffic-analysis.net. Accessed 18 June 2023
4. Spoiler: New ThreatLabz Report Reveals Over 85% of Attacks Are Encrypted. ThreatLabz State of Encrypted Attacks 2022 Report. https://www.zscaler.com/blogs/security-research/2022-encrypted-attacks-report. Accessed 18 June 2023
5. Anderson, B., McGrew, D.: Machine learning for encrypted malware traffic classification: accounting for noisy labels and non-stationarity. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1723–1732 (2017)
6. Bazuhair, W., Lee, W.: Detecting malign encrypted network traffic using perlin noise and convolutional neural network. In: 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0200–0206. IEEE (2020)
7. Chen, L., Gao, S., Liu, B., Lu, Z., Jiang, Z.: THS-IDPC: a three-stage hierarchical sampling method based on improved density peaks clustering algorithm for encrypted malicious traffic detection. J. Supercomput. **76**(9), 7489–7518 (2020)

8. Creech, G., Hu, J.: A semantic approach to host-based intrusion detection systems using contiguous and discontiguous system call patterns. IEEE Trans. Comput. **63**(4), 807–819 (2013)
9. Dai, R., Gao, C., Lang, B., Yang, L., Liu, H., Chen, S.: SSL malicious traffic detection based on multi-view features. In: Proceedings of the 2019 the 9th International Conference on Communication and Network Security, pp. 40–46 (2019)
10. Draper-Gil, G., Lashkari, A.H., Mamun, M.S.I., Ghorbani, A.A.: Characterization of encrypted and VPN traffic using time-related. In: Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP), pp. 407–414 (2016)
11. Fu, C., Li, Q., Xu, K.: Detecting unknown encrypted malicious traffic in real time via flow interaction graph analysis. ArXiv abs/2301.13686 (2023). https://api.semanticscholar.org/CorpusID:256415981
12. Fu, Z., et al.: Encrypted malware traffic detection via graph-based network analysis. In: Proceedings of the 25th International Symposium on Research in Attacks, Intrusions and Defenses (2022). https://api.semanticscholar.org/CorpusID:252910591
13. Gu, Y., Hao, X., Zhang, X.: Multi-granularity representation learning for encrypted malicious traffic detection. Chin. J. Comput. 1–12 (2023). https://kns.cnki.net/kcms/detail/11.1826.tp.20230421.1719.020.html
14. He, H.Y., Yang, Z.G., Chen, X.N.: PERT: payload encoding representation from transformer for encrypted traffic classification. In: 2020 ITU Kaleidoscope: Industry-Driven Digital Transformation (ITU K), pp. 1–8 (2020). https://doi.org/10.23919/ITUK50268.2020.9303204
15. Korczyński, M., Duda, A.: Markov chain fingerprinting to classify encrypted traffic. In: IEEE INFOCOM 2014-IEEE Conference on Computer Communications, pp. 781–789. IEEE (2014)
16. Li, R., Song, Z., Xie, W., Zhang, C., Zhong, G., Pei, X.: HALNet: a hybrid deep learning model for encrypted C&C malware traffic detection. In: Yang, M., Chen, C., Liu, Y. (eds.) NSS 2021. LNCS, vol. 13041, pp. 326–339. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-92708-0_21
17. Lin, X., Xiong, G., Gou, G., Li, Z., Shi, J., Yu, J.: ET-BERT: a contextualized datagram representation with pre-training transformers for encrypted traffic classification. In: Proceedings of the ACM Web Conference 2022, pp. 633–642 (2022)
18. Liu, J., Tian, Z., Zheng, R., Liu, L.: A distance-based method for building an encrypted malware traffic identification framework. IEEE Access **7**, 100014–100028 (2019)
19. Mehta, S., Rastegari, M.: MobileViT: light-weight, general-purpose, and mobile-friendly vision transformer. arXiv preprint arXiv:2110.02178 (2021)
20. Shen, M., et al.: Machine learning-powered encrypted network traffic analysis: a comprehensive survey. IEEE Commun. Surv. Tutor. **25**, 791–824 (2022)
21. Van Ede, T., et al.: FlowPrint: semi-supervised mobile-app fingerprinting on encrypted network traffic. In: Network and Distributed System Security Symposium (NDSS), vol. 27 (2020)
22. Wang, W., Zhu, M., Wang, J., Zeng, X., Yang, Z.: End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In: 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 43–48 (2017). https://doi.org/10.1109/ISI.2017.8004872
23. Wang, W., Zhu, M., Zeng, X., Ye, X., Sheng, Y.: Malware traffic classification using convolutional neural network for representation learning. In: 2017 International Conference on Information Networking (ICOIN), pp. 712–717. IEEE (2017)

24. Yang, F., Xu, J., Xiong, C., Li, Z., Zhang, K.: PROGRAPHER: an anomaly detection system based on provenance graph embedding. In: USENIX Security Symposium (2023). https://api.semanticscholar.org/CorpusID:259861739
25. Zhang, H., et al.: TFE-GNN: a temporal fusion encoder using graph neural networks for fine-grained encrypted traffic classification. In: Proceedings of the ACM Web Conference 2023 (2023). https://api.semanticscholar.org/CorpusID:258333744
26. Zhao, H., Zhang, S., Qiao, Z., Huang, X., Zhang, X.: On the performance of deep learning methods for identifying abnormal encrypted proxy traffic. In: 2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 1416–1423 (2022). https://doi.org/10.1109/TrustCom56396.2022.00200
27. Zheng, J., Zeng, Z., Feng, T.: GCN-ETA: high-efficiency encrypted malicious traffic detection. Secur. Commun. Netw. **2022** (2022)