Revolutionizing Encrypted Traffic Classification with MH-Net: A Multi-View Heterogeneous Graph Model

Haozhen Zhang[1,2,3*], **Haodong Yue**[1,2,3*], **Xi Xiao**[1,2,3†], **Le Yu**[4], **Qing Li**[2], **Zhen Ling**[5], **Ye Zhang**[6]

[1]Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China
[2]Peng Cheng Laboratory, Shenzhen, China
[3]Key Laboratory of Cyberspace Security, Ministry of Education of China, Zhengzhou, China
[4]Nanjing University of Posts and Telecommunications, Nanjing, China
[5]Southeast University, Nanjing, China
[6]National University of Singapore, Singapore
zhang-hz21@tsinghua.org.cn, yhd24@mails.tsinghua.edu.cn, xiaox@sz.tsinghua.edu.cn, yulele08@njupt.edu.cn, liq@pcl.ac.cn, zhenling@seu.edu.cn, e1124294@u.nus.edu

Abstract

With the growing significance of network security, the classification of encrypted traffic has emerged as an urgent challenge. Traditional byte-based traffic analysis methods are constrained by the rigid granularity of information and fail to fully exploit the diverse correlations between bytes. To address these limitations, this paper introduces MH-Net, a novel approach for classifying network traffic that leverages multi-view heterogeneous traffic graphs to model the intricate relationships between traffic bytes. The essence of MH-Net lies in aggregating varying numbers of traffic bits into multiple types of traffic units, thereby constructing multi-view traffic graphs with diverse information granularities. By accounting for different types of byte correlations, such as header-payload relationships, MH-Net further endows the traffic graph with heterogeneity, significantly enhancing model performance. Notably, we employ contrastive learning in a multi-task manner to strengthen the robustness of the learned traffic unit representations. Experiments conducted on the ISCX and CIC-IoT datasets for both the packet-level and flow-level traffic classification tasks demonstrate that MH-Net achieves the best overall performance compared to dozens of SOTA methods.

Code — https://github.com/ViktorAxelsen/MH-Net

Introduction

As advancements in computer network technology continue and various devices connect to the Internet, user privacy becomes increasingly vulnerable to malicious attacks. While encryption technologies like VPNs and Tor (Ramadhani 2018) offer protection to users (Sharma, Dangi, and Mishra 2021; Xiao et al. 2024), they can paradoxically serve as tools for attackers to conceal their identities. Traditional data packet inspection (DPI) methods have lost effectiveness against encrypted traffic (Papadogiannaki and Ioannidis 2021). Designing a universally effective method to classify an attacker's network activities (e.g., website browsing or application usage) from encrypted traffic remains a formidable challenge.

In the past few years, many methods have been proposed to enhance the capability of encrypted traffic classification techniques. Among them, statistic-based methods (Taylor et al. 2016; Hayes and Danezis 2016; van Ede et al. 2020; Panchenko et al. 2016; Xu, Geng, and Jin 2022) generally rely on hand-crafted traffic statistical features and then leverage a traditional machine learning model for classification. However, they require heavy feature engineering and are susceptible to unreliable flows (Zhang et al. 2023). With the burgeoning of representation learning (Le-Khac, Healy, and Smeaton 2020), some methods also use deep learning models to conduct traffic classification, such as pre-trained language models (Lin et al. 2022; Meng et al. 2022), neural networks (Liu et al. 2019; Zhang et al. 2023; Zhao et al. 2023), etc. Although these methods demonstrate competitive performance, they fail to sufficiently uncover the fine-grained correlations between traffic bytes, which can be attributed to the following two disadvantages. (1) **Rigid information granularity constrained by bytes**. Most existing methods treat a byte as an indivisible unit by default, which ignores the diverse granularity of information contained in traffic data. To give a practical example, a Chinese character is represented by two bytes, while an English character is represented by only one byte, indicating that traffic data contains information of different granularities universally (note that the granularity is not necessarily bytes, it may also be bits). (2) **Lack of consideration of multiple correlation types between bytes**. Current methods mix the correlation of bytes at different positions in the byte sequence, which overlooks and fails to utilize the differences between diverse types of correlation (e.g., the correlation types between bytes in the header and bytes in the payload are different). Consequently, how to uncover and leverage the potential fine-grained correlations between traffic bytes to enhance traffic classification is a salient problem.

To address the above challenges, in this paper, we propose a novel model named **MH-Net**, which classifies network traffic using multi-view heterogeneous traffic graphs.

*These authors contributed equally.
†Corresponding author.

In particular, MH-Net first aggregates different numbers of traffic bits into multiple types of traffic units (e.g., 4-bit units and 8-bit units), which facilitates the diversity of information granularity of traffic data. Since graphs have excellent capabilities in processing relational data, MH-Net further leverages point-wise mutual information (PMI) to convert diverse types of traffic unit sequences into multi-view traffic graphs. On top of this, considering the distinct functionality in different parts of the traffic unit sequence, MH-Net introduces three types of unit correlations, i.e., header-header, header-payload, and payload-payload unit correlations, and designs a heterogeneous traffic graph encoder for multi-view heterogeneous graph representation learning. To strengthen the robustness of traffic unit representations, we additionally perform contrastive learning in a multi-task manner on traffic graphs. To comprehensively evaluate MH-Net, we conduct both the packet-level and flow-level traffic classification tasks on the ISCX and CIC-IoT datasets. Experimental results show that MH-Net achieves competitive performance and ranks first among all the state-of-the-art methods. Further analysis of traffic units also reveals the potential trade-off between complementarity and interference between traffic units with different information granularity.

To summarize, our main contributions include:

- We propose a novel model named MH-Net, which aggregates different numbers of traffic bits into multiple types of traffic units to construct multi-view traffic graphs, enriching the diversity of information granularity while improving model performance.
- MH-Net further introduces three types of traffic unit correlations to model the heterogeneity of traffic graphs and employs a heterogeneous graph neural network for feature extraction. Furthermore, we conduct contrastive learning in a multi-task manner to enhance the robustness of traffic unit representations.
- We conduct experiments on both the packet-level and flow-level traffic classification tasks using the ISCX and CIC-IoT datasets. The experimental results show that MH-Net achieves the overall best performance compared with dozens of baselines.

Related Work

Flow-level Traffic Classification Methods. Flow-level traffic classification methods aim to classify traffic flows, which can be summarized into three categories.

• *Statistical Feature Based Methods.* Many methods use statistical features to represent packet properties and utilize traditional machine learning models for classification. AppScanner (Taylor et al. 2016) extracts features from traffic flows based on bidirectional flow characteristics, while CUMUL (Panchenko et al. 2016) uses cumulative packet length as its feature. ETC-PS (Xu, Geng, and Jin 2022) strengthens packet length sequences by applying the path signature theory, and hierarchical clustering is also leveraged for feature extraction by Conti *et al.* (Conti et al. 2015).

• *Fingerprinting Matching Based Methods.* Fingerprinting denotes traffic characteristics and is also used in traffic identification. FlowPrint (van Ede et al. 2020) generates traffic fingerprints by creating correlation graphs that compute activity values between destination IPs. K-FP (Hayes and Danezis 2016) uses the random forest to construct fingerprints and identifies unknown samples through k-nearest neighbor matching.

• *Deep Learning Based Methods.* Deep learning has demonstrated powerful learning abilities, and many traffic classification methods are based on it. RBRN (Zheng et al. 2020), DF (Sirinam et al. 2018), and FS-Net (Liu et al. 2019) all use statistical feature sequences (e.g., packet length sequences) as inputs for convolutional neural networks (CNNs) or recurrent neural networks (RNNs). Additionally, there are some methods using raw bytes as features. EBSNN (Xiao et al. 2022) combines RNNs with the attention mechanism to process header and payload byte segments. ET-BERT (Lin et al. 2022) conducts pre-training tasks on large-scale traffic datasets to learn a powerful raw byte representation, which is time-consuming and expensive. Graph neural networks (GNNs) are another model that can be used for traffic classification tasks. GraphDApp (Shen et al. 2021) builds traffic interaction graphs from traffic bursts and uses GNNs for representation learning. TFE-GNN (Zhang et al. 2023) employs point-wise mutual information (Yao, Mao, and Luo 2019) to construct byte-level traffic graphs and designs a traffic graph encoder for feature extraction. YaTC (Zhao et al. 2023) features a masked autoencoder-based traffic transformer to enable efficient feature extraction while boosting performance.

Packet-level Traffic Classification Methods. In contrast, packet-level traffic classification methods identify diverse categories of each network packet. Securitas (Yun et al. 2015) generates n-grams for raw bytes and utilizes Latent Dirichlet Allocation (LDA) to form protocol keywords as features, followed by SVM, C4.5 decision tree, or Bayes network for packet classification. 2D-CNN (Lim et al. 2019) and 3D-CNN (Zhang et al. 2020) treat packet bytes as pixel values and convert them into images, which are further fed into 2D-CNNs and 3D-CNNs for packet classification. DP (Lotfollahi et al. 2020) leverages CNNs and autoencoders to extract byte features. BLJAN (Mao et al. 2021) explores the correlation between packet bytes and their labels and encodes them into a joint embedding space to classify packets. EBSNN (Xiao et al. 2022) and ET-BERT (Lin et al. 2022) can also perform packet classification. But, they still require two independent training or fine-tuning to conduct flow-level and packet-level tasks, which are computationally expensive. PacRep (Meng et al. 2022) leverages triplet loss (Schroff, Kalenichenko, and Philbin 2015) without data augmentation and jointly optimizes multiple packet-level tasks to learn a better packet representation.

In summary, existing methods conduct traffic classification without sufficiently considering the informative correlations contained in raw bytes, thus facing performance bottlenecks.

Methodology

In this section, we will introduce the components of MH-Net, which include multi-view traffic graph construction, heterogeneous traffic graph representation learning, and

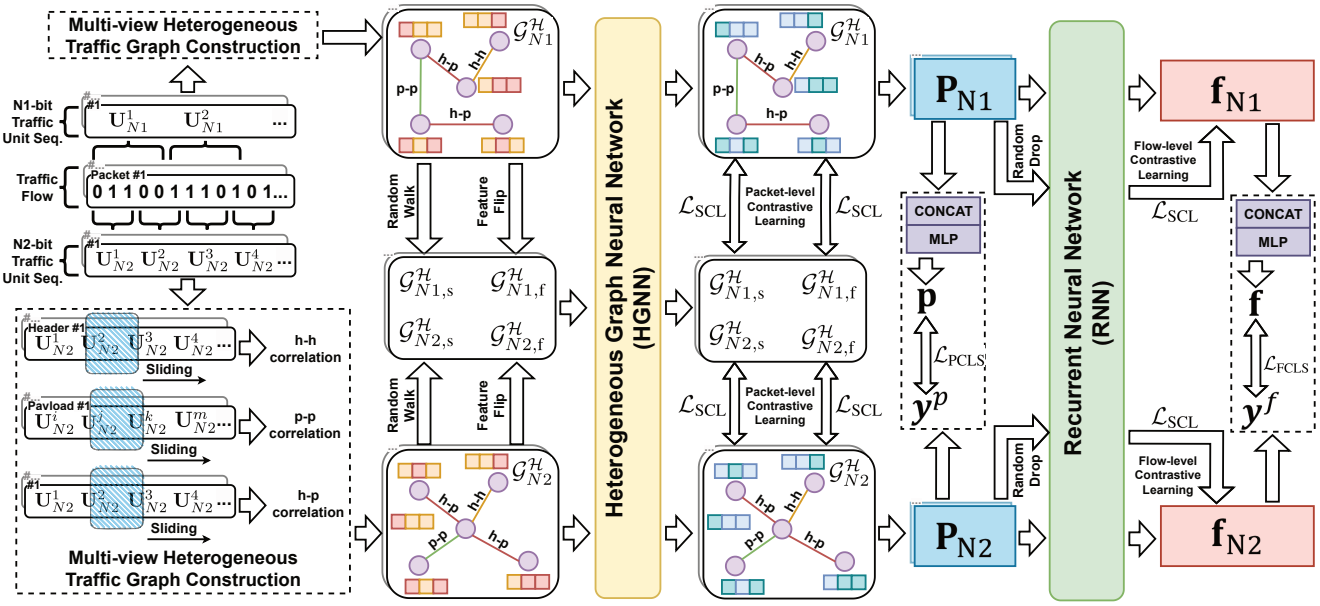

Figure 1: MH-Net Model Architecture.

multi-task training of MH-Net. The MH-Net model architecture is shown in Figure 1.

Multi-View Traffic Graph Construction

In this section, we first describe the rationale behind the utilization of traffic units and detail the construction of multi-view traffic graphs using traffic units.

Traffic Units with Diverse Granularity. To uncover the diverse information granularity contained in raw bytes, we attempt to seek more beyond the byte itself and instead aggregate raw bits of different lengths into various traffic units. Traffic units with different bit numbers can "interpret" or "express" the transmitted data from distinct perspectives, thereby being able to mine potential highly discriminative features. Taking character encoding as an example, given binary data with 16 bits, it can be converted into a Chinese character using 16-bit traffic units or 2 English characters using 8-bit traffic units. Therefore, we can draw on various traffic units to perform multi-view feature extraction on the same traffic data to derive better traffic representations.

Traffic Graph Construction. Although we aggregate traffic data into traffic units, they are still scattered individuals, which is not conducive to mining the potential fine-grained correlations between a sequence of traffic units. Inspired by (Zhang et al. 2023; Yao, Mao, and Luo 2019), due to the natural advantages of graph structure in data correlation modeling, we further convert the traffic unit sequence into a traffic graph to make it an interconnected whole. For a sequence of N-bit traffic units, we leverage point-wise mutual information (PMI) to quantify the correlation between traffic units. PMI employs a sliding window on a traffic unit sequence and counts the number of times two units co-occur within the window. The co-occurring frequency of the two units and the frequency of each unit are used to compute PMI, which can be formulated as:

$$p(\mathbf{U}_N^i, \mathbf{U}_N^j) = \frac{\#W(\mathbf{U}_N^i, \mathbf{U}_N^j)}{\#W}, \quad p(\mathbf{U}_N^i) = \frac{\#W(\mathbf{U}_N^i)}{\#W} \tag{1}$$

$$\mathrm{PMI}(\mathbf{U}_N^i, \mathbf{U}_N^j) = \log \frac{p(\mathbf{U}_N^i, \mathbf{U}_N^j)}{p(\mathbf{U}_N^i)p(\mathbf{U}_N^j)} \tag{2}$$

where $\#W$ is the total number of sliding windows, $\#W(\mathbf{U}_N^i)$ is the number of times unit $\mathbf{U}_N^i$ appears in the sliding window, and $\#W(\mathbf{U}_N^i, \mathbf{U}_N^j)$ denotes the number of times unit $\mathbf{U}_N^i$ and $\mathbf{U}_N^j$ appear in the sliding window simultaneously. We connect two traffic units $\mathbf{U}_N^i$ and $\mathbf{U}_N^j$ if and only if $\mathrm{PMI}(\mathbf{U}_N^i, \mathbf{U}_N^j) > 0$ is satisfied, which indicates high unit correlation. For the traffic data in each packet, we select two types of traffic units, e.g., N1-bit and N2-bit traffic units, and apply the PMI algorithm on their sequences to construct the multi-view traffic graph $\mathcal{G}_{N1}$ and $\mathcal{G}_{N2}$ for each packet, respectively. Note that the node features of the traffic graph are the value of each traffic unit.

Heterogeneous Traffic Graph Representation Learning

This section will present the further integration of heterogeneous correlations in traffic graphs. A heterogeneous traffic graph encoder is elaborately designed to conduct heterogeneous traffic graph representation learning.

Heterogeneous Traffic Unit Correlations. In the byte sequence of a packet, the header and payload have information heterogeneity due to their different functions (the former carries the metadata of the packet, and the latter carries the

actual transmitted content). However, the homogeneous correlations between traffic units in $\mathcal{G}$ restrict the full utilization of the heterogeneity in the header and payload. Therefore, we argue that the correlations between traffic units should have multiple types according to their positions in the traffic unit sequence. Intuitively, we propose three types of traffic unit correlations, i.e., header-header (h-h), payload-payload (p-p), and header-payload (h-p) correlations, to alleviate the above issue. We then apply the PMI algorithm to the traffic unit sequences of the header, payload, and header + payload (i.e., the entire traffic unit sequence), respectively, to obtain three types of traffic edges, which are further integrated into a heterogeneous traffic graph $\mathcal{G}^{\mathcal{H}}$.

In this way, we transform $\mathcal{G}_{N1}$ and $\mathcal{G}_{N2}$ into multi-view heterogeneous traffic graphs $\mathcal{G}^{\mathcal{H}}_{N1}$ and $\mathcal{G}^{\mathcal{H}}_{N2}$ whereby the correlations between traffic units becomes more fine-grained, enabling seamless heterogeneous fusion of the header and payload in a single traffic graph.

Heterogeneous Traffic Graph Encoder. So far, we have obtained traffic graphs $\mathcal{G}^{\mathcal{H}}_{N1}$ and $\mathcal{G}^{\mathcal{H}}_{N2}$ for each packet, which will be further fed into a heterogeneous traffic graph encoder for traffic representation learning. Specifically, the heterogeneous traffic graph encoder features a heterogeneous graph neural network (HGNN) to extract discriminative features of traffic graphs. HGNN employs GraphSAGE (Hamilton, Ying, and Leskovec 2017) as its basic backbone, and the model weights are not shared across edge types. Generally, the forward pass of HGNN in the l-th layer can be described as:

$$\mathbf{m}_v^{(l)} = \mathrm{MSG}^{(l)}\left(\left\{\mathbf{h}_u^{(l-1)}, u \in N(v)\right\}; \theta^{l,m}_{h-h}, \theta^{l,m}_{p-p}, \theta^{l,m}_{h-p}\right) \tag{3}$$

$$\mathbf{h}_v^{(l)} = \mathrm{AGG}^{(l)}\left(\mathbf{h}_v^{(l-1)}, \mathbf{m}_v^{(l)}; \theta^{l,a}_{h-h}, \theta^{l,a}_{p-p}, \theta^{l,a}_{h-p}\right) \tag{4}$$

where $\mathbf{h}_v^{(l)}$ are the embedding vectors of nodes v in layer l, $\mathbf{m}_v^{(l)}$ is the computed neighbor message for node v in layer l, and $N(v)$ is the neighbors of node v. $\mathrm{MSG}^{(l)}(\cdot)$ is a message computation function parameterized by $\theta^{l,m}_{h-h}, \theta^{l,m}_{p-p}, \theta^{l,m}_{h-p}$ and $\mathrm{AGG}^{(l)}(\cdot)$ is a message aggregation function parameterized by $\theta^{l,a}_{h-h}, \theta^{l,a}_{p-p}, \theta^{l,a}_{h-p}$ in layer l. Then, we perform the element-wise average operation on all the node embedding vectors in the last layer to obtain the final packet-level traffic representation $\mathbf{p}_{N1}$ and $\mathbf{p}_{N2}$ for $\mathcal{G}^{\mathcal{H}}_{N1}$ and $\mathcal{G}^{\mathcal{H}}_{N2}$, respectively, which can be simplified as:

$$\mathbf{p}_{N1} = \mathrm{HGNN}(\mathcal{G}^{\mathcal{H}}_{N1}), \quad \mathbf{p}_{N2} = \mathrm{HGNN}(\mathcal{G}^{\mathcal{H}}_{N2}) \tag{5}$$

We can further obtain the flow-level traffic representation $\mathbf{f}_{N1}$ and $\mathbf{f}_{N2}$ using a recurrent neural network (RNN):

$$\mathbf{f}_{N1} = \mathrm{RNN}(\mathbf{p}^1_{N1}, ..., \mathbf{p}^L_{N1}), \quad \mathbf{f}_{N2} = \mathrm{RNN}(\mathbf{p}^1_{N2}, ..., \mathbf{p}^L_{N2}) \tag{6}$$

where L is the length of a traffic flow.

Multi-Task Training of MH-Net

We aim to jointly train MH-Net in a multi-task manner for better optimization. The training objective of MH-Net mainly includes traffic classification and contrastive learning tasks, which will be presented in detail below.

Traffic Classification Tasks. In this work, we conduct the flow-level and packet-level traffic classification tasks in MH-Net simultaneously. As shown in Figure 1, we utilize an unshared traffic classifier with a multilayer perceptron (MLP) to transform the flow-level and packet-level representations and calculate the corresponding traffic classification task loss, respectively:

$$\mathbf{f} = \mathrm{CONCAT}(\mathbf{f}_{N1}, \mathbf{f}_{N2}), \quad \mathbf{p} = \mathrm{CONCAT}(\mathbf{p}_{N1}, \mathbf{p}_{N2}) \tag{7}$$

$$\mathcal{L}_{\text{FCLS}} = \mathrm{CE}(\mathrm{MLP}(\mathbf{f}), \boldsymbol{y}^f), \quad \mathcal{L}_{\text{PCLS}} = \mathrm{CE}(\mathrm{MLP}(\mathbf{p}), \boldsymbol{y}^p) \tag{8}$$

where $\mathrm{CONCAT}(\cdot)$ denotes concatenation operation, $\mathrm{CE}(\cdot)$ is the cross entropy loss function, $\boldsymbol{y}^f$ is the flow label, and $\boldsymbol{y}^p$ is the packet label that is consistent with the flow label it belongs to.

Contrastive Learning at Dual Levels. Inspired by the powerful representation learning capabilities of contrastive learning (van den Oord, Li, and Vinyals 2018; He et al. 2020; Chen et al. 2020), which aims to learn semantic-invariant representations by contrasting positive and negative sample pairs derived from various data augmentation, we propose to utilize it to further enhance the multi-view packet-level and flow-level traffic representations in MH-Net. In particular, MH-Net features a supervised contrastive loss (Khosla et al. 2020) to take better advantage of the data labels during training, which can be formulated as:

$$\mathcal{L}_{\text{SCL}}(\mathbf{z}) = \sum_{i \in I} \frac{-1}{|M(i)|} \sum_{m \in M(i)} \log \frac{\exp\left(\mathbf{z}_i \cdot \mathbf{z}_m / \tau\right)}{\sum_{k \in K(i)} \exp\left(\mathbf{z}_i \cdot \mathbf{z}_k / \tau\right)} \tag{9}$$

where $\mathbf{z}$ refers to the embedding vector collection of two groups of augmented data samples from the same source. $i \in I$ is the index of an arbitrary augmented sample in $\mathbf{z}$, $K(i) \equiv I \backslash \{i\}$, and $M(i) \equiv \{m \in K(i) : \boldsymbol{y}_m = \boldsymbol{y}_i\}$ is the indices of all positive samples of i, conditioned on the same data label. The symbol $\cdot$ represents the inner product operation and $\tau \in \mathcal{R}^+$ denoted the temperature parameter.

Packet-level Contrastive Learning. Since the packet-level traffic representation is derived based on the traffic graph, we need to generate an augmented graph for further contrastive learning (You et al. 2020; Qiu et al. 2020). In this work, we mainly adopt two augmentation approaches: graph structure and node feature augmentation. For graph structure augmentation, we leverage the random walk algorithm (Tong, Faloutsos, and Pan 2006), which starts by randomly picking a node and iteratively performs a random traversal on its neighbors to obtain the augmented traffic graph $\mathcal{G}^{\mathcal{H}}_{N,\text{s}}$. As for node feature augmentation, we augment the traffic graph by flipping its node features (Hou et al. 2022), resulting in another augmented traffic graph $\mathcal{G}^{\mathcal{H}}_{N,\text{f}}$. After obtaining the embedding vector of $\mathcal{G}^{\mathcal{H}}_{N,\text{s}}$ and $\mathcal{G}^{\mathcal{H}}_{N,\text{f}}$, the packet-level contrastive loss can be formulated as:

$$\begin{aligned}\mathcal{L}_{\text{PCL}} = \sum_{i}^{\{1,2\}} & \mathcal{L}_{\text{SCL}}(\{\mathrm{HGNN}(\mathcal{G}^{\mathcal{H}}_{Ni,\text{s}}), \mathbf{p}_{Ni}\}) + \\ & \mathcal{L}_{\text{SCL}}(\{\mathrm{HGNN}(\mathcal{G}^{\mathcal{H}}_{Ni,\text{f}}), \mathbf{p}_{Ni}\})\end{aligned} \tag{10}$$

Notably, we treat the embedding vector $\mathbf{p}_N$ of the original traffic graph $\mathcal{G}_N^{\mathcal{H}}$ as an "anchor" without augmentation for training stability (with slight abuse of notation). By directly perturbing structures and features on the traffic graph, the semantic-invariant representations contained in traffic units (i.e., nodes) can be well-uncovered through contrastive learning, thus enhancing model performance.

Flow-level Contrastive Learning. We further conduct the flow-level contrastive learning task by randomly dropping packets from a traffic flow with a certain probability $P_{\text{PD}} \in [0, 1]$ and obtain an augmented traffic flow. Similarly, the flow-level contrastive loss can be formulated as:

$$\mathcal{L}_{\text{FCL}} = \sum_{i}^{\{1,2\}} \mathcal{L}_{\text{SCL}}(\{\text{RNN}(\mathbf{p}_{Ni}^1 \odot \rho_1, ..., \mathbf{p}_{Ni}^L \odot \rho_L), \mathbf{f}_{Ni}\}) \tag{11}$$

where $\rho_i \in \{0, 1\}$ is drawn from a Bernoulli distribution $\rho_i \sim \mathcal{B}(P_{\text{PD}})$, denoting whether to drop the packet. Such learning paradigms can also help the model capture the common characteristics of the traffic flow through augmentation, leading to a robust flow-level representation.

Overall Training Objective. In summary, we propose the overall end-to-end multi-task training objective of MH-Net as follows:

$$\mathcal{L} = \mathcal{L}_{\text{PCLS}} + \mathcal{L}_{\text{FCLS}} + \alpha\mathcal{L}_{\text{PCL}} + \beta\mathcal{L}_{\text{FCL}} \tag{12}$$

where $\alpha, \beta \in [0, 1]$ are the coefficients that control the contribution of the packet-level and flow-level contrastive tasks, respectively.

Experiments

Experimental Settings

Dataset To thoroughly evaluate MH-Net on the packet-level and flow-level traffic classification tasks, we adopt the CIC-IoT (Dadkhah et al. 2022), ISCX VPN-nonVPN (Gil et al. 2016) and ISCX Tor-nonTor (Lashkari et al. 2017) datasets. In the experiment, we conduct all experiments independently on these five datasets, i.e., CIC-IoT, ISCX-VPN, ISCX-NonVPN, ISCX-Tor, and ISCX-NonTor.

Since our method performs both the flow-level and packet-level classification tasks, we first adopt stratified sampling to partition the flow-level training and testing dataset into 9:1 according to the number of traffic flows for all datasets. All packets in the flow-level training and testing datasets are directly used as the packet-level training and testing datasets, respectively. The category of each packet is consistent with the traffic flow to which it belongs.

Implementation Details and Baselines Since the value range of an N-bit traffic unit is determined by the number of bits it contains (i.e., 2^N), we utilize 4-bit and 8-bit traffic units to construct multi-view heterogeneous traffic graphs to achieve a trade-off between diversity and computational costs. The max flow length (i.e., the max packet number within a flow) is set to 15. The number of layers in HGNN is set to 4, and we initialize the RNN to LSTM in MH-Net by default. As for the random walk, we set the scale of subgraphs following (Qiu et al. 2020), and the restart probability is 0.8. We set the packet dropping ratio P_{PD} to 0.6, and the temperature coefficient τ is 0.07. The objective coefficients α and β are set to 1.0 and 0.5, respectively. We implement MH-Net and conduct all experiments with PyTorch and Deep Graph Library. The experimental results are reported as the mean over five runs on an NVIDIA RTX 3080.

As for evaluation metrics, we use Overall Accuracy (AC) and Macro F1-score (F1). We compare MH-Net with the flow-level and packet-level methods for a comprehensive comparison. The comparison baselines include **Flow-level Traffic Classification Methods** (i.e., AppScanner (Taylor et al. 2016), K-FP (K-Fingerprinting) (Hayes and Danezis 2016), CUMUL (Panchenko et al. 2016), ETC-PS (Xu, Geng, and Jin 2022), FS-Net (Liu et al. 2019), DF (Sirinam et al. 2018), ET-BERT (Lin et al. 2022), GraphDApp (Shen et al. 2021), TFE-GNN (Zhang et al. 2023), YaTC (Zhao et al. 2023)) and **Packet-level Traffic Classification Methods** (i.e., Securitas (Yun et al. 2015), 2D-CNN (Lim et al. 2019), 3D-CNN (Zhang et al. 2020), DeepPacket (DP) (Lotfollahi et al. 2020), BLJAN (Mao et al. 2021), EBSNN (Xiao et al. 2022)).

For the rest of the experiment section, we will evaluate MH-Net from the following research questions:

RQ1: How does MH-Net perform on the packet-level and flow-level classification tasks?

RQ2: How much does each module of MH-Net contribute to the model performance?

RQ3: How sensitive is MH-Net to hyper-parameters, and how does the choice and combination of traffic units affect model performance?

Comparison Experiments (RQ1)

The comparison results of the flow-level and packet-level traffic classification tasks on the CIC-IoT and ISCX datasets are shown in Tables 1 and 2, respectively.

Flow-level Traffic Classification Results. According to Table 1, we can draw several conclusions w.r.t. the flow-level task. (1) MH-Net achieves the overall best results w.r.t. all the metrics on the CIC-IoT and ISCX datasets, followed by TFE-GNN and YaTC, respectively. (2) Compared with traditional statistical feature methods, our approach surpasses them by a large margin due to the sufficient utilization of traffic bytes instead of statistical features. As for deep learning methods, MH-Net still has obvious advantages. (3) Among these methods, although TFE-GNN and YaTC also leverage raw bytes for representation learning, the overall performance is still significantly worse than that of MH-Net, which implies insufficient byte utilization and ignorance of fine-grained byte correlations in their model design. (4) Additionally, ET-BERT, which features a large number of parameters and pretraining on large-scale datasets, reaches decent results on the ISCX-nonVPN dataset, but its computational overhead is outrageously large.

Packet-level Traffic Classification Results. From Table 2, we can also conclude several observations and findings regarding the packet-level task. (1) MH-Net still has absolute superiority over other baselines, followed by EBSNN-LSTM and EBSNN-GRU. (2) Although EBSNN shows competitive results across all the datasets, its overall results are still inferior to MH-Net by a noticeable margin,

Model	CIC-IoT		ISCX-Tor		ISCX-nonTor		ISCX-VPN		ISCX-nonVPN		Avg. Rank
	AC	F1	AC	F1	AC	F1	AC	F1	AC	F1	
AppScanner	0.9674	0.9620	0.7543	0.6163	0.9153	0.8273	0.8889	0.8722	0.7576	0.7486	6
K-FP	0.9349	0.9254	0.7771	0.6313	0.8741	0.8167	0.8713	0.8747	0.7551	0.7387	8
CUMUL	0.9153	0.9101	0.6686	0.4997	0.8605	0.7627	0.7661	0.7644	0.6187	0.5897	9
ETC-PS	0.9218	0.9122	0.7486	0.6033	0.9365	0.8486	0.8889	0.8851	0.7273	0.7208	7
FS-Net	0.9805	0.9759	0.8286	0.7242	0.9278	0.8285	0.9298	0.9234	0.7626	0.7555	5
DF	0.8664	0.8601	0.6514	0.4719	0.8568	0.7590	0.8012	0.7921	0.6742	0.6701	10
ET-BERT	0.9565	0.9122	0.9543	0.9397	0.9029	0.8332	0.9532	0.9463	**0.9167**	0.9235	4
GraphDApp	0.6808	0.7263	0.4286	0.2281	0.6936	0.5352	0.6491	0.5740	0.4495	0.3614	11
TFE-GNN	0.9699	0.9666	**0.9886**	0.9855	0.9390	0.8507	0.9591	0.9536	0.9040	**0.9240**	2
YaTC	0.9397	0.9105	0.9868	0.9869	**0.9579**	**0.9522**	0.9605	0.9671	0.7546	0.7544	3
MH-Net	**0.9900**	**0.9896**	**0.9886**	**0.9886**	0.9465	0.9453	**0.9942**	**0.9941**	0.9141	0.9141	**1**

Table 1: Experimental Results on CIC-IoT, ISCX Tor-nonTor, and ISCX VPN-nonVPN Datasets w.r.t. *Flow*-level Traffic Classification Task. (BOLD indicates the best score, and UNDERLINE denotes the second-best one. Avg. Rank indicates the average ranking of each model's results across all datasets and metrics.)

Model	CIC-IoT		ISCX-Tor		ISCX-nonTor		ISCX-VPN		ISCX-nonVPN		Avg. Rank
	AC	F1	AC	F1	AC	F1	AC	F1	AC	F1	
Securitas-C4.5	0.8675	0.8571	0.9520	0.9429	0.8916	0.8652	0.7250	0.7250	0.5833	0.6212	4
Securitas-SVM	0.6526	0.7098	0.8229	0.8046	0.7333	0.7180	0.6375	0.6454	0.5750	0.6933	9
Securitas-Bayes	0.6667	0.7541	0.8083	0.6938	0.8062	0.7842	0.6125	0.6736	0.5416	0.6938	10
2D-CNN	0.7745	0.7863	0.3781	0.3576	0.6560	0.6877	0.2887	0.2219	0.5517	0.4638	11
3D-CNN	0.8788	0.8779	0.7837	0.7534	0.4791	0.4273	0.8445	0.8436	0.5346	0.5114	8
DP-SAE	0.7284	0.6134	0.8190	0.8078	0.8244	0.7889	0.7227	0.6849	0.6985	0.6952	6
DP-CNN	0.8664	0.8607	0.6622	0.6342	0.8784	0.8633	0.9270	0.9283	0.4326	0.3201	7
BLJAN	0.9643	0.9620	0.7580	0.7762	0.1348	0.2081	0.8539	0.7646	0.7356	0.7640	5
EBSNN-GRU	0.9440	0.9431	0.9319	0.9208	0.9271	0.9207	0.9467	0.9463	0.8103	0.8096	3
EBSNN-LSTM	0.9759	**0.9819**	0.7663	0.7183	**0.9519**	**0.9484**	0.9527	0.9531	0.8156	0.8128	2
MH-Net	**0.9806**	0.9800	**0.9916**	**0.9917**	0.9156	0.9122	**0.9768**	**0.9766**	**0.8822**	**0.8814**	**1**

Table 2: Experimental Results on CIC-IoT, ISCX Tor-nonTor, and ISCX VPN-nonVPN Datasets w.r.t. *Packet*-level Traffic Classification Task. (BOLD indicates the best score, and UNDERLINE denotes the second-best one. Avg. Rank indicates the average ranking of each model's results across all datasets and metrics.)

which can be attributed to the fact that the byte segment in EBSNN cannot fully exploit the informative correlation between bytes, thus causing performance bottleneck. (3) Securitas, which uncovers n-gram keywords for frequency matching and ranks fourth according to the Avg. Rank in Table 2, performs much worse than MH-Net due to the inflexible and rigid keyword patterns, reflecting the importance and effectiveness of mining informative byte correlations.

From the overall perspective of the flow-level and packet-level tasks, MH-Net reaches the best comprehensive performance, verifying the effectiveness of our proposed model.

Ablation Study (RQ2)

For a clear understanding of MH-Net's architecture design, we conduct a comprehensive ablation study on the CIC-IoT and ISCX-VPN dataset w.r.t. F1-Score, and the results are shown in Table 3. From Table 3, we can conclude that the 8-bit traffic unit contributes more to the model performance than the 4-bit one. Still, the information carried by the latter cannot be ignored (the detailed analysis of traffic units will be presented later in RQ3). After we convert the heterogeneous traffic graph into a homogeneous one (i.e., with only one edge type), we can see a significant drop in the results, showing that it is necessary to model the heterogeneous correlations between traffic units (or bytes). We also investigate how contrastive learning tasks contribute to the results. The results indicate that integrating packet-level and flow-level contrastive loss can improve the classification performance of both tasks, with the latter having a more salient impact.

In short, MH-Net achieves the best results on flow-level and packet-level tasks compared to various variants.

More Analysis on MH-Net (RQ3)

In this section, we first analyze the sensitivity of MH-Net. Then, to investigate the impact of traffic units, we further analyze the choice and combination of traffic units.

Variant	CIC-IoT		ISCX-VPN	
	Flow	Packet	Flow	Packet
w/o 4-bit View	0.9800	0.9681	0.9498	0.9355
w/o 8-bit View	0.8842	0.8970	0.1702	0.1488
w/o Hetero.	0.9154	0.9551	0.9120	0.9010
w/o L_{PCL}	0.9835	0.9618	0.9524	0.9524
w/o L_{FCL}	0.9767	0.9631	0.9467	0.9204
MH-Net	**0.9896**	**0.9800**	**0.9941**	**0.9766**

Table 3: Ablation Study of MH-Net on the CIC-IoT and ISCX-VPN Dataset w.r.t. F1-Score.

Sensitivity Analysis. We conduct sensitivity analysis on the ISCX-VPN dataset to investigate the impact of the packet-level and flow-level contrastive loss ratio α and β. Figure 2 shows that (1) both tasks show a gradual improvement trend with the increase of α, which implies the effectiveness of packet-level contrastive learning. (2) In contrast, the impact of β on model performance fluctuates slightly and reaches its best at about $\beta = 0.5$, which may be due to the excessive randomness caused by the augmentation of the traffic flow. One promising solution may be introducing learnable augmentation to drop packets adaptively.

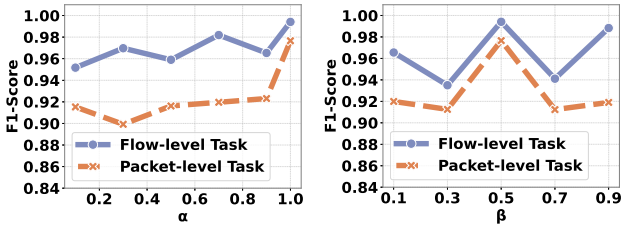

Figure 2: Sensitivity Analysis w.r.t. α and β on the ISCX-VPN Dataset.

The Choice of Traffic Units. As shown in Figure 3, we aggregate different numbers of traffic bits into traffic units and conduct experiments using single-view heterogeneous traffic graphs. Figure 3 shows that the 8-bit traffic unit achieves the best result, while others are much worse than it. In addition, both the flow-level and packet-level tasks show almost the same fluctuation trend w.r.t. F1-Score with traffic unit changes. The results also suggest that byte (i.e., 8-bit) is the primary unit of information transmission, and the reason for the decline in the performance of other traffic units may come from two aspects: (1) they break the integrity of bytes; (2) the traffic graph's size is limited by the traffic unit's bit length, which may be one of the influencing factors.

The Combination of Traffic Units. To further reveal how the combination of traffic units affects model performance, we conduct experiments on any two traffic unit combinations of 2, 4, 6, 8, and 10, and the result is shown in Figure 4. From Figure 4, we can draw the following conclusions. (1) The combination of 4&8-bit traffic units reaches

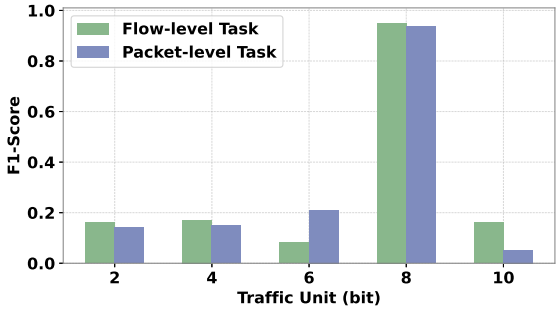

Figure 3: The Choice of Traffic Units on the ISCX-VPN Dataset.

the best result, followed by the combination of 8&10-bit traffic units. (2) Combining two traffic units can improve the model performance to a certain extent compared to a single traffic unit (e.g., 4&6-bit, 4&8-bit, 4&10-bit, and 8&10-bit traffic units), but it may also produce more negative results (e.g., 2&8-bit and 6&8-bit traffic units). This implies that there may be a trade-off between information complementarity and interference between traffic units with different information granularity, which can be potentially utilized to further improve model performance.

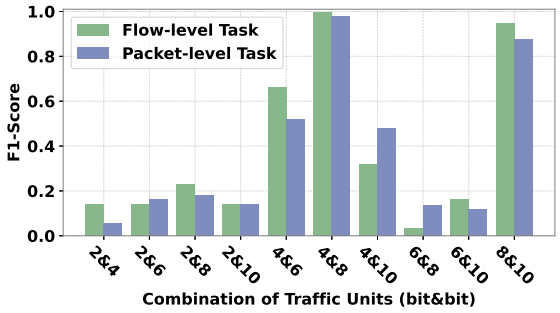

Figure 4: The Combination of Traffic Units on the ISCX-VPN Dataset.

Conclusion

This paper proposes an effective model named MH-Net, which constructs multi-view heterogeneous traffic graphs via point-wise mutual information by aggregating different numbers of traffic bits into traffic units. In particular, we uncover the heterogeneous byte correlations contained in the traffic graph and employ a heterogeneous graph neural network for graph representation learning. Furthermore, we conduct supervised contrastive learning in a multi-task manner to obtain more robust traffic representations. The experimental results show that MH-Net reaches the overall best performance on both the flow-level and packet-level traffic classification tasks compared with dozens of baselines. In addition, an extensive analysis of traffic units reveals the possibility of using complementary information from different traffic units to improve traffic classification performance.

Acknowledgments

This work was supported in part by the Major Key Project of PCL(No. PCL2023A06-4), Open Foundation of Key Laboratory of Cyberspace Security, Ministry of Education of China, Henan Key Laboratory of Network Cryptography (No.KLCS20240304), Natural Science Research Start-up Foundation of Recruiting Talents of Nanjing University of Posts and Telecommunications (Grant No. NY223164, NY224001), National Natural Science Foundation of China Grant No. 62022024, by CCF-Huawei Populus Grove Fund, by Jiangsu Provincial Key Laboratory of Network and Information Security Grant No. BM2003201, Key Laboratory of Computer Network and Information Integration of Ministry of Education of China Grant No. 93K-9, and Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *International Conference on Machine Learning*, 1597–1607.

Conti, M.; Mancini, L. V.; Spolaor, R.; and Verde, N. V. 2015. Analyzing Android Encrypted Network Traffic to Identify User Actions. *IEEE Transactions on Information Forensics and Security*, 11(1).

Dadkhah, S.; Mahdikhani, H.; Danso, P. K.; Zohourian, A.; Truong, K. A.; and Ghorbani, A. A. 2022. Towards the development of a realistic multidimensional IoT profiling dataset. In *2022 19th Annual International Conference on Privacy, Security & Trust (PST)*, 1–11. IEEE.

Gil, G. D.; Lashkari, A. H.; Mamun, M.; and Ghorbani, A. A. 2016. Characterization of Encrypted and VPN Traffic Using Time-Related Features. In *International Conference on Information Systems Security and Privacy*, 407–414.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *Conference on Neural Information Processing Systems*.

Hayes, J.; and Danezis, G. 2016. k-fingerprinting: a Robust Scalable Website Fingerprinting Technique. In *USENIX Security Symposium*, 1187–1203.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *Computer Vision and Pattern Recognition*, 9729–9738.

Hou, Z.; Liu, X.; Cen, Y.; Dong, Y.; Yang, H.; Wang, C.; and Tang, J. 2022. Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 594–604.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised Contrastive Learning. In *Conference on Neural Information Processing Systems*, 18661–18673.

Lashkari, A. H.; Draper-Gil, G.; Mamun, M. S. I.; and Ghorbani, A. A. 2017. Characterization of Tor Traffic Using Time Based Features. In *International Conference on Information System Security and Privacy*, 253–262.

Le-Khac, P. H.; Healy, G.; and Smeaton, A. F. 2020. Contrastive Representation Learning: A Framework and Review. *IEEE Access*, 8: 193907–193934.

Lim, H.-K.; Kim, J.-B.; Heo, J.-S.; Kim, K.; Hong, Y.-G.; and Han, Y.-H. 2019. Packet-based Network Traffic Classification Using Deep Learning. In *International Conference on Artificial Intelligence in Information and Communication*, 046–051.

Lin, X.; Xiong, G.; Gou, G.; Li, Z.; Shi, J.; and Yu, J. 2022. ET-BERT: A Contextualized Datagram Representation with Pre-training Transformers for Encrypted Traffic Classification. In *The Web Conference*, 633–642.

Liu, C.; He, L.; Xiong, G.; Cao, Z.; and Li, Z. 2019. FS-Net: A Flow Sequence Network For Encrypted Traffic Classification. In *IEEE Conference on Computer Communications*, 1171–1179.

Lotfollahi, M.; Siavoshani, M. J.; Zade, R. S. H.; and Saberian, M. 2020. Deep packet: a novel approach for encrypted traffic classification using deep learning. *Soft Computing*, 24(3): 1999–2012.

Mao, K.; Xiao, X.; Hu, G.; Luo, X.; Zhang, B.; and Xia, S. 2021. Byte-Label Joint Attention Learning for Packet-grained Network Traffic Classification. In *International Workshop on Quality of Service*, 1–10.

Meng, X.; Wang, Y.; Ma, R.; Luo, H.; Li, X.; and Zhang, Y. 2022. Packet Representation Learning for Traffic Classification. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3546–3554.

Panchenko, A.; Lanze, F.; Zinnen, A.; Henze, M.; Pennekamp, J.; Wehrle, K.; and Engel, T. 2016. Website Fingerprinting at Internet Scale. In *Annual Network and Distributed System Security Symposium*.

Papadogiannaki, E.; and Ioannidis, S. 2021. A Survey on Encrypted Network Traffic Analysis Applications, Techniques, and Countermeasures. *ACM Computing Surveys*, 54(6): 1–35.

Qiu, J.; Chen, Q.; Dong, Y.; Zhang, J.; Yang, H.; Ding, M.; Wang, K.; and Tang, J. 2020. GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1150–1160.

Ramadhani, E. 2018. Anonymity communication VPN and Tor: a comparative study. *Journal of Physics: Conference Series*, 983(1): 012060.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *Computer Vision and Pattern Recognition*, 815–823.

Sharma, R.; Dangi, S.; and Mishra, P. 2021. A Comprehensive Review on Encryption based Open Source Cyber Security Tools. In *IEEE International Conference on Signal Processing, Computing and Control*, 614–619.

Shen, M.; Zhang, J.; Zhu, L.; Xu, K.; and Du, X. 2021. Accurate Decentralized Application Identification via Encrypted Traffic Analysis Using Graph Neural Networks.

IEEE Transactions on Information Forensics and Security, 16(1): 2367–2380.

Sirinam, P.; Imani, M.; Juarez, M.; and Wright, M. K. 2018. Deep Fingerprinting: Undermining Website Fingerprinting Defenses with Deep Learning. In *Conference on Computer and Communications Security*, 1928–1943.

Taylor, V. F.; Spolaor, R.; Conti, M.; and Martinovic, I. 2016. AppScanner: Automatic Fingerprinting of Smartphone Apps From Encrypted Network Traffic. In *IEEE European Symposium on Security and Privacy*, 439–454.

Tong, H.; Faloutsos, C.; and Pan, J.-Y. 2006. Fast random walk with restart and its applications. In *Sixth international conference on data mining (ICDM'06)*, 613–622. IEEE.

van den Oord, A.; Li, Y.; and Vinyals, O. 2018. Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748v2*.

van Ede, T.; Bortolameotti, R.; Continella, A.; Ren, J.; Dubois, D. J.; and Lindorfer, M. 2020. FlowPrint: Semi-Supervised Mobile-App Fingerprinting on Encrypted Network Traffic. In *Annual Network and Distributed System Security Symposium*.

Xiao, X.; Xiao, W.; Li, R.; Luo, X.; Zheng, H.; and Xia, S. 2022. EBSNN: Extended Byte Segment Neural Network for Network Traffic Classification. *IEEE Transactions on Dependable and Secure Computing*, 19(5): 3521–3538.

Xiao, X.; Zhou, X.; Yang, Z.; Yu, L.; Zhang, B.; Liu, Q.; and Luo, X. 2024. A comprehensive analysis of website fingerprinting defenses on Tor. *Computers & Security*, 136: 103577.

Xu, S.-J.; Geng, G.-G.; and Jin, X.-B. 2022. Seeing Traffic Paths: Encrypted Traffic Classification With Path Signature Features. *IEEE Transactions on Information Forensics and Security*, 17(1): 2166–2181.

Yao, L.; Mao, C.; and Luo, Y. 2019. Graph Convolutional Networks for Text Classification. In *AAAI Conference on Artificial Intelligence*, 7370–7377.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. Graph Contrastive Learning with Augmentations. In *Conference on Neural Information Processing Systems*, 5812–5823.

Yun, X.; Wang, Y.; Zhang, Y.; and Zhou, Y. 2015. A Semantics-Aware Approach to the Automated Network Protocol Identification. *IEEE/ACM Transactions on Networking*, 24(1): 583–595.

Zhang, H.; Yu, L.; Xiao, X.; Li, Q.; Mercaldo, F.; Luo, X.; and Liu, Q. 2023. TFE-GNN: A Temporal Fusion Encoder Using Graph Neural Networks for Fine-grained Encrypted Traffic Classification. In *The Web Conference*, 2066–2075.

Zhang, J.; Li, F.; Ye, F.; and Wu, H. 2020. Autonomous Unknown-Application Filtering and Labeling for DL-based Traffic Classifier Update. In *IEEE Conference on Computer Communications*, 397–405.

Zhao, R.; Zhan, M.; Deng, X.; Wang, Y.; Wang, Y.; Gui, G.; and Xue, Z. 2023. Yet another traffic classifier: A masked autoencoder based traffic transformer with multi-level flow representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 5420–5427.

Zheng, W.; Gou, C.; Yan, L.; and Mo, S. 2020. Learning to Classify: A Flow-Based Relation Network for Encrypted Traffic Classification. In *The Web Conference*, 13–22.