2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

CTRL: Contrastive Traffic Recognition with Lightweight network

Zijia Song[1[0009-0008-5499-705X]], Yelin Wang[2[0009-0003-8272-6066]], Pan Chen[1[0009-0001-0724-7028]], Zhaobin Shen[1[0009-0003-6552-4964]], Longxi Li[1[0009-0009-1588-6610]], Wanyu Chen[1[0000-0003-4470-9655]] and Yuliang Lu[1[0000-0002-8502-9907]]

[1] National University of Denfence Technology, Hefei, China
[2] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
luyuliang@nudt.edu.cn, wanyuchen@nudt.edu.cn

Abstract. Network traffic classification plays a vital part in network security and management. Facing the growing sophistication of encryption techniques, various works focus to acquire underlying features and try to achieve more advanced identification. However, current methods mostly depend on pre-training or using large models to fit implicit relationship, which could cause imbalance unsupervised learning due to long-tail distribution and may be impractical for intrusion detection because of large cost. Therefore, we propose a non-pretrained lightweight framework, termed Contrastive Traffic Recognition with Lightweight network (CTRL), to fully explore spatial-temporal features in traffic. Specifically, two-stream architecture is adopted to decouple the mixed feature extraction while lightweight encoder is further improved to avoid weak representation. By employing contrastive loss, single model can grasp common knowledge from different views, which realizes better traffic recognition. Extensive experiments conducted on six public traffic datasets from various tasks validate the more superior performances of our CTRL which maintains the fewest parameters, compared to state-of-the-art approaches with an average improvement of 7.5%.

Keywords: Traffic classification, Lightweight model, Spatial-temporal features, Contrastive learning.

1 Introduction

Efficient network traffic recognition is essential for managing networks, ensuring security, and improving user experiences. By accurately categorizing traffic from various services and applications, it also plays a pivotal role in threat detection and service optimization [1]. However, with the rise of encryption technologies (e.g., TLS) and anonymous networks (e.g., VPN, Tor), how to explore underlying features from traffic data has grown increasingly complicated [2], presenting significant challenges to traditional classification methods [3].

Confronted with the problem mentioned above, several studies, like ET-BERT [6], YaTC [7], and NetMamba [8], have applied self-supervised pre-training to probe common features from large volumes of unlabeled traffic. However, the predominance of

normal traffic in unlabeled data limits the understanding of malicious flows with few scale at pre-training stage. Additionally, unsupervised representation models generally require substantial parameters, posing challenges for resource-limited gateway devices. It is believed that traffic contains spatial and temporal features [25], thus focusing on relationships between different views can reveal deeper insights. For instance, Seydali et al. [9] employed 1D-CNN, Attention-Bi-LSTM as well as SAE to extract and aggregate features for classification. Hu et al. [10] enhanced the performance by cascading CNN and LSTM. But spatial and temporal models both need to participate in the evaluation period, which may cause inference delays and storage costs.

Through the above analysis, large models may be impractical for embedded devices, but lightweight networks are also struggling to fit sophisticated relationships within traffic flow, which is crucial for intrusion detection. Therefore, we propose Contrastive Traffic Recognition with Lightweight network (CTRL) to achieve real-time recognition with less memory cost while extract generalized features to ensure high accuracy [4]. To be specific, standard input is built by segmenting and recombining the packet payload of raw flow and transformed into diverse formats with different views. Original single feature extractor is decoupled to lightweight spatial and temporal encoders while each stream fully extracts single view features, avoiding weak representation by disentangled feature learning [21]. In order to achieve the fusion of both features, we redefine the problem as an alignment issue and adopt contrastive loss to transfer knowledge in the training process. When evaluating, we only employ the spatial model for traffic classification, which realizes to apply lightweight network with fully learned common spatial-temporal knowledge for better recognition.

In short, our contributions are summarized as follows: (1) We provide a groundbreaking perspective, which firstly transform the spatial-temporal fusion problem into an alignment issue and design a novel framework for traffic recognition termed CTRL. (2) We propose a lightweight spatial encoder and introduce NetMamba as temporal encoder with full considerations of mobile adaptation as well as efficiency. (3) We conduct extensive experiments in various tasks and prove the advantages of our proposed CTRL.

2 Related Works

Multimodal alignment. Multimodality enhances understanding and decision-making by integrating information from multiple sensory modalities [38]. Thereinto, ALBEF [39] aligns visual and language representations, using momentum distillation to improve multimodal embeddings. FLAVA [40] enhances multitask and cross-modal learning by jointly pre-training text and images. ALIGN [41] jointly trains language and image encoders, significantly enhancing performance across various vision and text benchmarks. In recent years, research around CLIP has further optimized computational efficiency and model representation capabilities. For instance, FLIP [42] brings lower computation and faster training times by randomly removing a large number of image patches during training process while SoftCLIP [43] applies fine-grained interior self-similarity as a softening target to alleviate the strict mutual exclusion problem.

Moreover, latent diffusion models generate reliable text embeddings as condition by using pretrained text encoder of CLIP and CLIPSelf [44] enhances region-level representation through self-distillation from CLIP's image encoder, which proves the powerful effects of CLIP.

Encrypted traffic classification based on deep learning. Encrypted traffic classification plays a crucial role in information security, and with the rise of deep learning, deep learning-based encrypted traffic classification has emerged as the mainstream approach. Qu et al. [45] presents an input-agnostic hierarchical deep learning framework for traffic fingerprinting that abstracts heterogeneous traffic features into homogeneous vectors. Zhang et al. [46] proposes a byte-level traffic graph construction approach and the Temporal Fusion Encoder with Graph Neural Networks (TFE-GNN), which uses dual embedding, GNN-based encoding, and cross-gated feature fusion. DE-GNN [47] introduces a dual embedding layer for separate encoding of packet header and payload, develops PacketCNN for feature extraction, and employs Graph Neural Networks for flow-level analysis. YaTC [7] introduces a masked autoencoder (MAE) based traffic transformer with hierarchical flow representations and multi-level attention mechanisms, significantly improving classification accuracy across real-world datasets. To further tackle inefficiencies in model architectures and representation quality, NetMamba [8] proposes a linear-time state space model tailored for networking, alongside a refined traffic representation scheme that mitigates bias and preserves essential information.

3 Methodology

3.1 Overview of Framework

It is essential to develop generic traffic analysis algorithm through non-pretrained approach rather than two-stage methods which may lead to huge consumption and unbalanced learning. Therefore, we propose CTRL to explore more information from traffic data for better classification and the conceptual overview is shown in **Fig. 1**. It is believed that traffic implicitly contains spatial and temporal information, hence we decouple the traditional single feature extractor and apply two-stream network for acquiring spatial-temporal representation more adequately. To be specific, original traffic is converted to spatial and temporal format with corresponding labels as supervised guidance. Each stream includes an encoder network $F_i(\cdot), i \in \{\mathbf{S}, \mathbf{T}\}$, a MLP layer $L_i(\cdot), i \in \{\mathbf{S}, \mathbf{T}\}$ for classification, and a projection head $\mathrm{H}_i(\cdot), i \in \{\mathbf{S}, \mathbf{T}\}$ which is used to map the data from original space into the latent space for fusion. Lightweight backbone is adopted with training from scratch and each single stream extracts feature with different views. The models and training paradigm will be described below in detail.

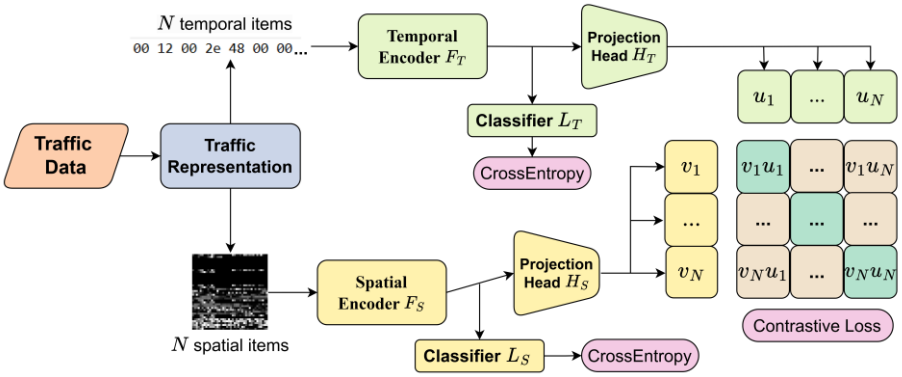

Fig. 1. The overall framework of CTRL. Traffic representation module handles the raw traffic to generate corresponding spatial and temporal items while N denotes the batch size. Decoupling feature extraction network can fully explore disentangled information and spatial-temporal fusion is reframed as an alignment issue. Contrastive loss is employed to realize the integration of partial knowledge and cross-entropy guarantees proper learning direction.

3.2 Traffic Representation

An effective traffic recognition approach necessitates a robust representation framework with appropriate granularity to differentiate diverse application scenarios or transmission intents for precise traffic analysis. Building upon the design principles proposed by Wang et al. [8], we derive hierarchical representations from raw traffic data and construct comprehensive flow-level characterizations. While traffic inherently contains spatial and temporal attributes, existing methods predominantly capture entangled features rather than pursuing disentangled representations. Specifically, the spatial perspective emphasizes localized patterns by structuring flow representations into matrices, which are then processed through convolutional encoders to extract spatial features. Meanwhile, the temporal perspective captures sequential dependencies within flows by converting representations into one-dimensional sequences, enabling temporal modeling via architectures like BERT [35] or Mamba [36]. This dual-view processing allows thorough exploitation of decoupled features [20], empowering individual feature extractors to capture richer knowledge from distinct perspectives through subsequent spatial-temporal fusion mechanisms.

3.3 Spatial and Temporal Encoder

Due to insufficient storage and computing resources, deploying high-performance models with high computational requirements on gateway devices is unpractical. Therefore, lightweight feature extractors are necessary for traffic classification and each single stream is supposed to focus on only one view to avoid underfitting because of the little scale of model.

For spatial encoder, we design a effective lightweight network to learn structural information of traffic data, enabling accurate identification of various classes. Specifically, our proposed spatial encoder adopts ShuffleNet v2 0.5 × architecture [18] which

could balance recognition accuracy as well as the lightweight performance. 0.5 × indicates the channel numbers shrinks to 0.5 times compared with the original model, which leads to smaller size. To further reduce computation cost and enhance the model, we leverage advanced techniques and redesign the basic module in ShuffleNet v2 0.5 × .

We find that the 1×1 convolution in the middle stages of ShuffleNet v2 still result in significant computational overhead, as also demonstrated in the work by S. Han et al. [32]. Inspired by GhostNet [5], we adopt a series of cheap linear operations to replace all 1 × 1 convolutions in stage 3 of original network to ulteriorly reduce operands while keep valuable information. Specific operation is according to the following function:

$$y_{ij} = \Phi_{i,j}(y'_i), \quad \forall i = 1, \dots, m \quad j = 1, \dots, s \tag{1}$$

where y'_i is the i-th intrinsic feature map and $\Phi_{i,j}$ indicates the j-th linear operation for generating the j-th ghost feature map y_{ij}. Through applying cheap operation on the 3-rd stage, we can further decrease the FLOPs of model while maintain the classification performance at the same time.

To better capture global information and address convolutional limitations, we incorporate Agent-Attention [34] behind the Global Pool layer of spatial model. Compared with Self-Attention [14], it has a lower time complexity with almost linear complexity while it performs better than Linear-Attention [15]. Moreover, placing this module in the final layer allows to acquire global receptive field while only increase little computational load due to fewer input dimensions for Agent-Attention [29]. The formula of is shown as follows:

$$O^A = \text{Attn}^S\big(Q, A, \text{Attn}^S(A, K, V)\big) \tag{2}$$

where Attn^S represents Softmax attention and $Q, K, V \in \mathbb{R}^{N \times C}$ denote query, key, and value matrices. $A \in \mathbb{R}^{N \times C}$ is defined as agent tokens. $N$is batch size with feature dimension as C while n indicates agent numbers far less than N.

For temporal encoder, the architecture of NetMamba encoder is adopted due to liner-time state space module and notable performance. By jointly trained with spatial encoder, it is suitable to learn common patterns within flows.

3.4 Objective Loss

Spatial-Temporal Contrastive Learning. Different from previous approaches that cascade models to extract the mixing spatial-temporal features, our single stream only focuses on spatial or temporal information. In order to acquire more generalized and comprehensive representation, these fully learned single features need to be integrated, which leads one stream to learn knowledge from the other perspective, enhancing the ability to understand common information respectively from both views [23,31]. For better realizing the fusion of disentangled features, we innovatively transform this problem as an alignment issue and the representations from separated views are explicitly

pulled into the same embedding space [30]. Concretely, corresponding spatial and temporal items are treated as matched pairs. Contrastive loss $\mathcal{L}_{\mathrm{CL}}$ is employed to maximize the representation similarity between matched pairs while minimize the similarity between negative pairs [22]. The format of this loss is shown as follows:

$$\begin{aligned}\mathcal{L}_{\mathrm{CL}} = -\frac{1}{2\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}\Big(\log\frac{\exp(\mathcal{S}(\mathrm{u_i},\mathrm{v_i})/\tau)}{\sum_{\mathrm{j}=1}^{\mathrm{N}}\exp(\mathcal{S}(\mathrm{u_i},\mathrm{v_j})/\tau)} \\ +\log\frac{\exp(\mathcal{S}(\mathrm{v_i},\mathrm{u_i})/\tau)}{\sum_{\mathrm{j}=1}^{\mathrm{N}}\exp(\mathcal{S}(\mathrm{v_i},\mathrm{u_j})/\tau)}\Big)\end{aligned} \tag{3}$$

where N denotes batch size and u_i as well as v_i are representations in latent space respectively from two different views. $\tau$is a learnable temperature parameter and $\mathcal{S}(\cdot,\cdot)$ indicates cosine similarity. Through contrastive learning, each single network can also acquire information from the other fully explored view and get more generalized representation [24].

Supervised Classification Guidance. Contrastive loss could integrate information from multiple views. However, the poor understanding of traffic data, especially in early process of training, can also propagate between the two streams, which brings troubles for extracting valuable features. Hence, we require new loss function to inject supervised information to ensure proper optimization direction [33]. The formula of supervised objective is displayed as follows:

$$\mathcal{L}_{\mathrm{CE-k}} = -\sum_{i=1}^{N}\sum_{c=1}^{M} y_{ic}\log L_k\big(F_k(x_i)\big)_c, k \in \{\mathbf{S},\mathbf{T}\} \tag{4}$$

where N denotes batch size and M indicates the number of classes. The calculation of this loss in each stream is similar. x_i indicates spatial or temporal sample coordinated with k and y_{ic} is one-hot encoding corresponding to the label. Through the guidance of ground truth, models can extract spatial and temporal features associated with the category and no longer keep an eye on the redundant information, which realizes beneficial fusion between these features rather than negative effects of noises from each stream.

The Overall Loss. In order to enlarge the view of model with labels as optimization supervision for generalized classification, we combine the above objectives and seek to minimize the overall loss functions simultaneously:

$$\min_{\theta}\alpha\mathcal{L}_{\mathrm{CL}} + \beta\mathcal{L}_{\mathrm{CE-S}} + \beta\mathcal{L}_{\mathrm{CE-T}} \tag{5}$$

where θdenotes all learnable parameters of CTRL while α and β are hyper-parameters used to control the impacts of different objectives. CTRL adopts single-stage joint training strategy with end-to-end optimization. Furthermore, in the initial stage of training process, α employs small value, which reduces noises from bad representations to guarantee correct training direction. With the development of training, each stream can fully extract features from single view, hence the weight of contrastive loss is increased to provide abroad receptive field. In the evaluation process, we only launch spatial model for inference, result from the reason that spatial model is taught how to extract common temporal features during training. Through combining these losses, front layers of each

stream can focus on separated features from different views while the back layers may pay attention to the common information, which realizes sufficient feature extraction and fusion while avoids inertia due to coupled learning from mixed features.

4 Experiments

4.1 Experimental Datasets

To assess the effectiveness and generalization of CTRL, our experiments are conducted on four classification tasks with six available real-world traffic datasets.

Encrypted Communication Classification. This task expects to distinguish encrypted communications by analyzing encrypted traffic. We use ISCXTor2016 [11] with Tor traffic data across 8 categories and VPN traffic data from 7 categories in the ISCXVPN2016 [12] as evaluation datasets.

Encrypted Application Classification. The task aims to identify the type of application based on network traffic under various encryption protocols. The CrossPlatform(Android) [13] and CrossPlatform(iOS) [13] as evaluation datasets respectively include 254 and 253 applications categories.

Attack Traffic Classification. The objective of this task is to detect potential attack traffic. Following Wang et al. [8], we employ the partial CICIoT2022 [16] with 6 categories to verify the performance of different approaches on this task.

Malware Traffic Classification. This task focuses on identifying traffic associated with malware activities. USTC-TFC2016 [17] with 20 categories is applied for this task, aiding in proactive threat detection. For above datasets, the encryption target is the payload, while the protocol header remains unencrypted.

4.2 Implementation Details

In our experimental setup, we utilize 4 NVIDIA GeForce RTX 4090 GPUs to train the models with a batch size of 128. The learning rates are set to 0.005 for spatial model and 0.002 for temporal model alongside a linear learning rate scaling policy. Adam is employed to facilitate the training process and the training process lasts for 300 epochs.

Table 1. The comparison results on ISCXVPN2016, ISCXTor2016 and USTC- TFC2016. AC and F1 respectively denotes accuracy and F1-score.

Method	ISCXVPN2016		ISCXTor2016		USTC-TFC2016	
	AC	F1	AC	F1	AC	F1
AppScanner	0.7643	0.7256	0.4034	0.2113	0.6998	0.6633
FlowPrint	0.9666	0.9681	0.1316	0.0306	0.7992	0.4381
FS-Net	0.7023	0.6660	0.7020	0.6999	0.7755	0.2672
DF	0.6287	0.2540	0.3324	0.0700	0.5845	0.4915
Deeppacket	0.8021	0.8017	0.3681	0.2681	0.8849	0.8883
2D-CNN	0.8126	0.8064	0.3462	0.3366	0.9226	0.9205
3D-CNN	0.8109	0.8079	0.3489	0.3396	0.9155	0.9116
TFE-GNN	0.8428	0.8447	0.7692	0.7618	0.9747	0.9734
CTRL(Ours)	**0.9883**	**0.9877**	**0.9986**	**0.9941**	**0.9963**	**0.9952**

4.3 Comparison Evaluation

We conduct extensive experiments on various tasks to validate the efficacy of CTRL while the results of comparison experiments are shown in **Table 1** and **Table 2**. AC and F1 respectively represent classification accuracy and F1-score while results of previous works are from [6], [7] and [8].

It is evident that CTRL brings impressive and stable performances across all datasets over existing non-pretrained models. Specifically, for encrypted communication classification on ISCXVPN2016 and ISCXTor2016, our model achieves a notable improvement, especially in ISCXTor2016, compared to the previous state-of-the-art methods. For encrypted application classification on CrossPlatform(Android) and CrossPlatform(iOS), CTRL surpasses all baseline methods and outperforms the state-of-the-art models (Deeppacket [26] and FlowPrint [27]) by more than nearly 5% with all evaluation metrics. For malware traffic classification on USTC-TFC2016, the proposed approach realizes an enhancement over 2% than TFE-GNN [28] regardless of accuracy or F1-score. Moreover, compared with pre-trained models with more data for training, CTRL is still outstanding and the comparison results are displayed in Fig. 2. Thereinto, x-coordinate represents the parameter quantity of model while y-coordinate denotes the classification accuracy on CrossPlatform(iOS). The point size indicates the volume of data for training. As non-pretrained approach, only employing labeled data, CTRL which maintains the fewest parameters transcends ET-BERT and YaTC in accuracy while is slightly weaker than NetMamba.

We highlight the noteworthy on how to realize a generalized traffic recognition scheme. A appropriate data preprocessing is essential, which incorporates both header and payload data with less information loss. Meanwhile,a well-designed potential lightweight network plays a crucial role in exploring underlying features [19] and contrastive loss can lead single model to acquire common knowledge from different views, enhancing encrypted traffic analysis capabilities.

Table 2. The comparison results on CrossPlatform(Android) and CrossPlatform(iOS). AC and F1 respectively denotes accuracy and F1-score.

Method	CrossPlatform(Android)		CrossPlatform(iOS)	
	AC	F1	AC	F1
AppScanner	0.1626	0.1413	0.1718	0.1283
FlowPrint	0.8739	0.8700	0.8712	0.8603
FS-Net	0.0147	0.0034	0.0293	0.0025
DF	0.3862	0.2527	0.3106	0.2140
GraphDApp	0.4031	0.2703	0.3245	0.2297
TFE-GNN	0.8141	0.8067	0.8241	0.8130
Deeppacket	0.8805	0.8138	0.9204	0.9034
CTRL(Ours)	**0.9432**	**0.9405**	**0.9671**	**0.9583**

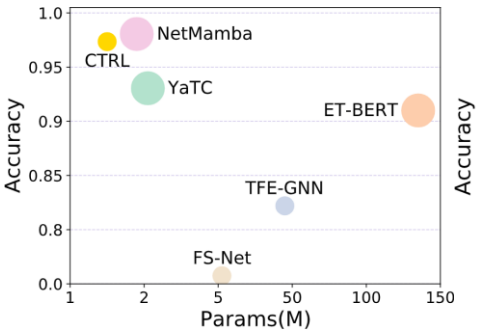

Fig. 2. Visualized comparison on CrossPlatform(iOS) with point size as training set scale.

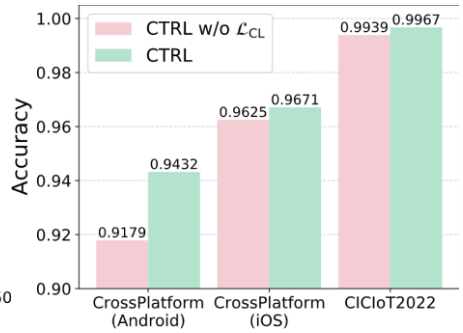

Fig. 3. Ablation results on various encrypted traffic datasets to figure out the effects of $\mathcal{L}_{\mathrm{CL}}$.

4.4 Ablation Study

In order to figure out how contrastive loss influence the final performance; we make ablation experiments on several datasets and Fig. 3 displays the results.

Obviously, contrastive loss brings benefits in any condition, particularly an improvement of 2.5% on CrossPlatform(Android). Thus, we announce that model can extract more spatial-temporal features from fusion view when adding contrastive loss while the launched spatial model may not actively explore temporal information without this loss. As for the mechanism, it is claimed that employing different losses could reconstruct the classification boundary when model reaches sufficient volume. Specifically, lightweight spatial model may only generate coarse-grained boundary by just applying Cross-Entropy loss [37]. Yet with joint optimization of diverse objectives, model could get fine-grained boundary, which might be helpful for classification.

5 Conclusions

In this paper, we employ two-stream networks to extract disentangled characteristics and reframe feature fusion as an alignment issue, proposing a novel approach named as CTRL. Through contrastive loss for knowledge transfer with cross-entropy keeping the proper learning direction, launched lightweight model could acquire common spatial-temporal information to realize better classification. Compared with different methods on various datasets, our approach is ahead of the other non-pretrained methods while also competitive with pretrained approaches, which can demonstrate the superiority of proposed CTRL.

References

1. Papadogiannaki, Eva, and Sotiris Ioannidis.: "A survey on encrypted network traffic analysis applications, techniques, and countermeasures." ACM Computing Surveys (CSUR) 54.6 (2021): 1-35.
2. Azab, Ahmad, et al.: "Network traffic classification: Techniques, datasets, and challenges." Digital Communications and Networks 10.3 (2024): 676-692.
3. Yuan, Lu, et al.: "Manticore: An Unsupervised Intrusion Detection System Based on Contrastive Learning in 5G Networks" ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2024.
4. Yang, Zhen, et al.: "A systematic literature review of methods and datasets for anomaly-based network intrusion detection." Computers & Security 116 (2022): 102675.
5. Han, Kai, et al.: "Ghostnet: More features from cheap operations." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.
6. Lin, Xinjie, et al.: "Et-bert: A contextualized datagram representation with pre-training transformers for encrypted traffic classification." Proceedings of the ACM Web Conference 2022.
7. Zhao, Ruijie, et al.: "Yet another traffic classifier: A masked autoencoder based traffic transformer with multi-level flow representation." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 37. No. 4. 2023.
8. Wang, Tongze, et al.: "NetMamba: Efficient Network Traffic Classification via Pre-training Unidirectional Mamba." arXiv preprint arXiv:2405.11449 (2024).
9. Seydali, Mehdi, et al.: "CBS: A Deep Learning Approach for Encrypted Traffic Classification With Mixed Spatio-Temporal and Statistical Features." IEEE Access (2023).
10. Hu, Xinyi, Chunxiang Gu, and Fushan Wei.: "[Retracted] CLD-Net: A Network Combining CNN and LSTM for Internet Encrypted Traffic Classification." Security and Communication Networks 2021.1 (2021): 5518460.
11. Lashkari, Arash Habibi, et al.: "Characterization of tor traffic using time based features." International Conference on Information Systems Security and Privacy. Vol. 2. SciTePress, 2017.
12. Draper-Gil, Gerard, et al.: "Characterization of encrypted and vpn traffic using time-related." Proceedings of the 2nd international conference on information systems security and privacy (ICISSP). 2016.
13. Ren, Jingjing, D. J. Dubois, and D. Choffnes.: "An international view of privacy risks for mobile apps." (2019).

14. Vaswani, A.: “Attention is all you need.” Advances in Neural Information Processing Systems (2017).
15. Shen, Zhuoran, et al.: “Efficient attention: Attention with linear complexities.” Proceedings of the IEEE/CVF winter conference on applications of computer vision. 2021.
16. Dadkhah, Sajjad, et al.: “Towards the development of a realistic multidimensional IoT profiling dataset.” 2022 19th Annual International Conference on Privacy, Security & Trust (PST). IEEE, 2022.
17. Wang, Wei, et al.: “Malware traffic classification using convolutional neural network for representation learning.” 2017 International conference on information networking (ICOIN). IEEE, 2017.
18. Ma, Ningning, et al.: “Shufflenet v2: Practical guidelines for efficient cnn architecture design.” Proceedings of the European conference on computer vision (ECCV). 2018.
19. Liu, Peng, et al.: “Lightweight High-Resolution Subject Matting in the Real World.” ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2024.
20. Guan, Quanlong, et al.: “Transformer Model with Multi-Type Classification Decisions for Intrusion Attack Detection of Track Traffic and Vehicle.” ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2024.
21. Wang, Xin, et al.: “Disentangled representation learning.” arXiv preprint arXiv:2211.11695 (2022).
22. Radford, Alec, et al.: "Learning transferable visual models from natural language supervision." International conference on machine learning. PMLR, 2021.
23. Yang, Chuanguang, et al.: “Mutual contrastive learning for visual representation learning.” Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 36. No. 3. 2022.
24. Yang, Chuanguang, et al.: “Online knowledge distillation via mutual contrastive learning for visual recognition.” IEEE Transactions on Pattern Analysis and Machine Intelligence 45.8 (2023): 10212-10227.
25. Lin, Kunda, Xiaolong Xu, and Honghao Gao.: “TSCRNN: A novel classification scheme of encrypted traffic based on flowspatiotemporal features for efficient management of IIoT.” Computer Networks 190 (2021): 107974.
26. Lotfollahi, Mohammad, et al.: “Deep packet: A novel approach for encrypted traffic classification using deep learning.” Soft Computing 24.3 (2020): 1999-2012.
27. Van Ede, Thijs, et al.: “Flowprint: Semi-supervised mobile-app fingerprinting on encrypted network traffic.” Network and distributed system security symposium (NDSS). Vol. 27. 2020.
28. Zhang, Haozhen, et al.: “Tfe-gnn: A temporal fusion encoder using graph neural networks for fine-grained encrypted traffic classification.” Proceedings of the ACM Web Conference 2023. 2023.
29. Srinivas, Aravind, et al.: “Bottleneck transformers for visual recognition.” Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.
30. Yang, Chuanguang, et al.: “CLIP-KD: An Empirical Study of CLIP Model Distillation.” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024.

31. Yang, Chuanguang, et al.: “Cross-image relational knowledge distillation for semantic segmentation.” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.
32. Han, Song, Huizi Mao, and William J. Dally.: “Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding.” arXiv preprint arXiv:1510.00149 (2015).
33. Yuyuan Sun, Xuan Wang, and Yuliang Lu.: “A review of deep learning model security research.” Information Countermeasure Technology. 2023.
34. Han, Dongchen, et al.: “Agent attention: On the integration of softmax and linear attention.” Proceedings of the European conference on computer vision (ECCV). 2024.
35. Devlin, Jacob.: “Bert: Pre-training of deep bidirectional transformers for language understanding.” arXiv preprint arXiv:1810.04805 (2018).
36. Gu, Albert, and Tri Dao.: “Mamba: Linear-time sequence modeling with selective state spaces.” arXiv preprint arXiv:2312.00752 (2023).
37. Du, Zhengxiao, et al.: “Understanding emergent abilities of language models from the loss perspective.” arXiv preprint arXiv:2403.15796 (2024).
38. Martin, Daniel, et al.: "Multimodality in VR: A survey." ACM Computing Surveys (CSUR) 54.10s (2022): 1-36.
39. Li, Junnan, et al.: "Align before fuse: Vision and language representation learning with momentum distillation." Advances in neural information processing systems 34 (2021): 9694-9705.
40. Singh, Amanpreet, et al.: "Flava: A foundational language and vision alignment model." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.
41. Jia, Chao, et al.: "Scaling up visual and vision-language representation learning with noisy text supervision." International conference on machine learning. PMLR, 2021.
42. Li, Yanghao, et al.: "Scaling language-image pre-training via masking." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023.
43. Gao, Yuting, et al.: "Softclip: Softer cross-modal alignment makes clip stronger." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 38. No. 3. 2024.
44. Wu, Size, et al.: "Clipself: Vision transformer distills itself for open-vocabulary dense prediction." arXiv preprint arXiv:2310.01403 (2023).
45. Qu, Jian, et al.: "An Input-Agnostic hierarchical deep learning framework for traffic fingerprinting." 32nd USENIX security symposium (USENIX Security 23). 2023.
46. Zhang, Haozhen, et al.: "Tfe-gnn: A temporal fusion encoder using graph neural networks for fine-grained encrypted traffic classification." Proceedings of the ACM Web Conference 2023. 2023.
47. Han, Xinbo, et al.: "DE-GNN: Dual embedding with graph neural network for fine-grained encrypted traffic classification." Computer Networks 245 (2024): 110372.