Keep Experts Diverse: A Task-Aware MoE for Multi-Task Traffic Analysis

Jiadong Fu[1,2], **Jiang Fang**[3,4*], **Jiyan Sun**[1*], **Laile Xi**[1,2], **Shangyuan Zhuang**[1,2]
Liru Geng[1,2], **Yinlong Liu**[1,2], **Zhiqiang Lv**[1,2]
[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] Electronic Engineering Institute, National University of Defense Technology, Hefei, China
[4] Jianghuai Advance Technology Center, Hefei, China
{fujiadong, sunjiyan, xilaile1998, zhuangshangyuan, gengliru, liuyinlong, lvzhiqiang}@iie.ac.cn, jiangfang2025@gmail.com

Abstract

Network traffic analysis is crucial for maintaining the security of networks. Yet deploying accurate models on edge nodes remains challenging due to protocol diversity, complex traffic behaviors, and stringent resource constraints. Although recent deep learning models achieve strong performance, their dense architectures incur high inference latency, making them unsuitable for edge deployments. Mixture-of-Experts offers a promising solution through conditional computation, yet existing methods overlook latent inter-task relations and suffer from severe expert load imbalance, leading to expert collapse and suboptimal efficiency.

To address these limitations, we propose TAMoE, a novel Task-Aware Mixture-of-Experts framework for multi-task network traffic analysis. TAMoE integrates task semantics into both routing and representation learning through a novel Task-Aware Routing, enabling the router to dynamically select experts conditioned on both traffic features and task information. Besides, to train stably usable and scalable multi-task models, we further design a collaborative multi-task training strategy that encompasses both a dataset mixed construction and a joint training process. Experiments on six public benchmarks show that TAMoE achieves state-of-the-art results. Deployment on an NVIDIA® Jetson AGX Orin edge node demonstrates that TAMoE reduces inference latency by 60.6% compared to dense models and lowers the Gini Coefficient from 0.391 to 0.275 relative to Multi-Router MoE, achieving more balanced expert utilization.

1 Introduction

Network traffic analysis serves as the cornerstone for ensuring the security and reliability of modern edge infrastructures. [Agarwal *et al.*, 2024; Dandamudi *et al.*, 2025]. As edge networks evolve, traffic patterns have become increasingly protocol-diverse and behaviorally complex, driven by heterogeneous edge devices and encrypted communications. [Javaheri *et al.*, 2023]. This complexity presents a fundamental conflict in the edge network. Traffic analysis models require sophisticated, yet real-world network security demands immediate, sub-second responsiveness. Any delay in traffic decision-making can compromise security and reliability.

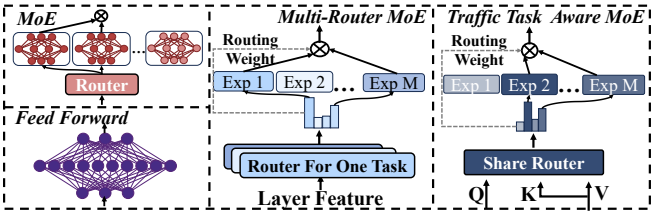

Figure 1: Comparison of forwarding strategies: single-task with Feed-Forward and MoE, multi-task with independent routers, and Traffic Task Aware MoE.

Traditional approaches [Lotfollahi *et al.*, 2020; Liu *et al.*, 2019] typically treat traffic analysis as a set of isolated single-task problems. deploying independent models for application classification, anomaly detection, and related tasks not only incurs substantial computational overhead, but also leads to redundant feature extraction, which collectively increase inference latency and deployment complexity on resource-constrained edge nodes[Nascita *et al.*, 2024]. These tasks are deeply entangled, *i.e.*, They often share latent semantic features and exhibit strong inter-task correlations [Chen *et al.*, 2024]. This limitation calls for a paradigm shift toward Multi-Task Learning, where a unified model concurrently handles diverse analysis tasks [Liu *et al.*, 2024]. However, deploying such unified models on edge nodes faces a critical bottleneck: *Inference Latency*. State-of-the-art pre-trained models [Lin *et al.*, 2022; Wang *et al.*, 2024a; Zhao *et al.*, 2023; Zhou *et al.*, 2025] typically rely on dense architectures (e.g., Transformers), where the fully connected Feed-Forward Networks (FFNs) force the activation of all parameters for every packet [Zhang *et al.*, 2021]. This dense computation paradigm results in prohibitive inference latency, rendering them unsuitable for real-time edge scenarios where speed is paramount [Wei *et al.*, 2024].

A promising answer lies in Mixture-of-Experts (MoE) [Cai

*Corresponding authors: Jiyan Sun and Jiang Fang.

et al., 2025], which introduce a conditional computation paradigm for dense models [Zhang *et al.*, 2021]. By activating only a subset of experts per forward pass, this method effectively addresses the computational bottlenecks of dense models, significantly reducing inference latency. In multi-task settings, this paradigm extends to Multi-router MoE (MMoE) [Zhang *et al.*, 2025], employing independent routers to select experts from a shared pool for task-specific processing. However, MMoE remains constrained by two critical limitations. **(1) Inadequate task correlation.** MMoE assigns an independent router to each task, which operates in isolation without considering inter-task relationships. This design neglects the underlying semantic correlations among tasks, limiting the model's ability to share and transfer knowledge across tasks [Upadhyay *et al.*, 2024]. **(2) Unbalanced expert utilization.** For multi-task training, MMoE routers rely solely on the current hidden features. Since different tasks may produce highly similar hidden features, the router tends to assign them repeatedly to the same subset of experts [Zhou *et al.*, 2022]. Consequently, many experts are only a few frequently activated, resulting in severe expert load imbalance, namely expert collapse [Wang *et al.*, 2024b]. These limitations reveal a critical gap: existing MoE framework either ignore task semantics or fail to ensure balanced expert utilization in multiple traffic analyses.

To tackle the above challenges and enable the next generation of intelligent network analysis, we propose a **T**ask **A**ware **M**ixture-**o**f-**E**xperts framework for multiple traffic analysis task, **TAMoE**, which redefines how traffic models scale, generalize, and adapt for edge-scale AI. Firstly, unlike prior MoE, we propose a novel Task-Aware Routing mechanism that leverages an attention mechanism to explicitly fuse input features with task embeddings. This enables dynamic, intent-conditioned expert selection while strictly preserving the low-inference-latency advantage. Distinctively, our design facilitates a single, intelligent shared router across all tasks. This strategy allows the model to capture latent inter-task correlations, thereby fostering effective expert collaboration and knowledge transfer. More importantly, this design provides semantic guidance, directing the router to activate a broader, more diverse set of experts. This structurally alleviates expert load imbalance and mitigates the risk of expert collapse by reducing over-reliance on a fixed subset of experts. Secondly, to optimize shared architecture, we introduce a collaborative training strategy in which each sample is explicitly paired with its task embedding to direct the routing process. By integrating multiple task-specific objectives into a composite loss, this enables the training of a general traffic feature extraction model that efficiently adapts to new tasks, which is crucial for edge-scale AI. Extensive experiments conducted on publicly available datasets demonstrate the effectiveness of TAMoE. Evaluations on an edge node (NVIDIA® JETSON AGX ORIN) demonstrate that TAMoE reduces inference latency by 60.6% compared with YATC, and reduces the Gini Coefficient from 0.391 to 0.275, compared with MMoE, effectively alleviating the risk of expert collapse. In summary, our work makes the contributions:

1. We propose TAMoE, Task-Aware Mixture-of-Experts, the MoE model designed for network traffic analysis, which significantly reduces inference latency.
2. We design a novel Task-Aware Routing mechanism that utilizes attention mechanism to introduce task semantics into expert selection, which adequately captures the correlations of multi-task and alleviate expert collapse risk.
3. We conduct extensive experiments on six traffic analysis tasks. The experimental results demonstrate that our method achieves state-of-the-art (SOTA) performance.

2 Related Work

2.1 Network Traffic Analysis Methods

Early studies on network traffic analysis primarily relied on conventional machine learning, where manually crafted statistical features were designed to characterize flow-level behaviors [Shao *et al.*, 2019; Taylor *et al.*, 2016; van Ede *et al.*, 2020]. While these methods provided acceptable performance in static or single-task settings, their heavy reliance on hand-crafted feature engineering hindered adaptability to dynamic network environments. With the advent of deep learning, researchers shifted toward automatically extracting representations directly from raw packet or flow data [Lotfollahi *et al.*, 2020; Liu *et al.*, 2019; Lian *et al.*, 2025a; Lian *et al.*, 2025b]. These models adopted deep neural networks to capture spatial or temporal dependencies within traffic, thereby improving the accuracy of tasks such as intrusion detection and anomaly detection. However, their high parameter counts and reliance on large-scale labeled datasets limited their scalability in edge environments. Pre-trained traffic models have been introduced to enhance transferability by learning generalizable representations from large-scale unlabeled traffic [Zhao *et al.*, 2023; Lin *et al.*, 2022; Wang *et al.*, 2024a; He *et al.*, 2020; Zhou *et al.*, 2025; Lian *et al.*, 2026b]. These methods typically rely on strategies such as masked token prediction [Lin *et al.*, 2022], or flow reconstruction [Wang *et al.*, 2024a] to capture latent protocol semantics and communication patterns[Lian *et al.*, 2026a]. After pre-training, the learned representations can be fine-tuned on smaller task-specific datasets.

Despite their success, existing models face two critical limitations. First, these models are predominantly built upon the Transformer architecture, where the fully connected feed-forward networks (FFNs) constitute a significant computational bottleneck [Wei *et al.*, 2024; Zhang *et al.*, 2021]. This dense computation results in excessive inference latency, especially when handling the long sequences of network traffic, making such models unsuitable for edge AI. Second, fine-tuning is typically performed on a task-specific basis, which involves training and deploying a separate, complete model copy for each downstream task. It fails to explore the shared knowledge that often exists in different traffic tasks, leaving an open gap that our proposed framework aims to fill.

2.2 Mixture of Experts (MoE)

The Mixture-of-Experts (MoE) paradigm [Cai *et al.*, 2025] has recently gained attention for its ability to scale model

capacity while maintaining efficient inference through conditional computation. This conditional strategy has enabled MoE to achieve remarkable success in large-scale natural language processing and computer vision. However, its potential in the domain of network traffic analysis remains largely unexplored. Given the growing complexity and scale of network traffic, applying MoE to this domain presents a promising opportunity to balance excessive inference latency with practical deployment efficiency. To extend MoE to multi-task scenarios, the Multi-gate Mixture-of-Experts (MMoE) framework was introduced [Cai *et al.*, 2025; Zhang *et al.*, 2025]. MMoE employs a shared pool of experts with independent task-specific routers, allowing each task to learn its own routing strategy. While effective in capturing task-specific variations, MMoE suffers from two critical limitations in the context of network traffic analysis: (1) the isolation of routers hindering cross-task knowledge sharing, and (2) routing mechanisms rely solely on hidden features, which often results in repeatedly selecting the same subset of experts. This causes severe load imbalance, namely expert collapse and results in inefficient knowledge sharing and degraded task performance.

Our work addresses these limitations by introducing a Task-Aware Mixture-of-Experts framework. Unlike conventional methods, TAMoE incorporates task semantic embeddings into the routing process, enabling task-sensitive expert selection that fosters better collaboration across tasks while alleviating expert imbalance. This design enhances both the effectiveness and efficiency of multi-task traffic analysis, making it suitable for deployment in edge-scale AI.

3 TAMoE Design

As illustrated in Figure 2, TAMoE comprises a traffic preprocessing pipeline, a novel Task-Aware MoE layer, and a multi-task collaborative training strategy. Initially, to handle protocol heterogeneity, the preprocessing module transforms raw PCAP traffic into unified token sequences via flow extraction, anonymization, and payload tokenization. Subsequently, to reconcile high parameter capacity with low inference latency, we propose a unified TAMoE framework. Here, a router leverages task embedding to enable the routing network to select a broader range of experts, thereby increasing the likelihood of activating more experts, enabling efficient conditional computation while mitigating expert collapse. Furthermore, by optimizing a composite loss over an interleaved multi-task dataset, the collaborative training strategy anable the TAMoE model to capture latent inter-task correlations, fostering robust knowledge sharing and ensuring superior transferability to unseen tasks.

3.1 Traffic Preprocess

To enable unified representation and efficient modeling of multi-protocol and heterogeneous network traffic, we employ a comprehensive traffic preprocessing pipeline that transforms raw PCAP packets into tokens [Zhao *et al.*, 2023]. The pipeline includes four stages: flow splitting, packet anonymization, packet alignment, and packet tokenization.

Flow Splitting and Packet Anonymization. This preserves the complete sequence order and contextual information necessary for modeling flow-level temporal patterns. We begin by segmenting raw captured traffic into semantically complete session flows using the network five-tuple (SrcIP, DstIP, SrcPort, DstPort, Protocol). Packets sharing the same five-tuple are grouped into a flow $F_i = \{P_1, P_2, \ldots, P_{n_i}\}$, where P_j denotes the j-th packet in the flow. Besides, to mitigate the risk of sensitive information leakage, IP addresses and port numbers are anonymized, by masking or zeroing, while preserving the overall data distribution [Arp *et al.*, 2022; Jacobs *et al.*, 2022]. This process preserves the temporal structure of session context and reduces the risk of model overfitting, which is essential for analysing patterns of traffic.

Packet Alignment and Tokenization. For compatibility with batched processing, we normalize the length of all traffic sequences, creating uniform input tensors by either padding shorter sequences or truncating longer ones. Let a packet be represented as a tuple $P = (s, d)$, where s denotes the header bytes and d denotes the payload bytes. We align each segment independently to fixed lengths L_s and L_d, respectively. If the actual header is longer than L_s, it is truncated; if it is shorter, it is padded with zero-value bytes. After aligning the traffic packet length, each aligned flow F_i is converted into a 1D token sequence. Every byte is split into 4-bit tokens, yielding a token sequence $T = \{t_1, t_2, \ldots, t_L\}$, where $L = \frac{n_i}{4} \times L_s \times L_d$, L represents the total number of tokens in a flow. Each input sample x therefore corresponds to a fixed-size token representation. This design ensures compatibility across traffic tasks and enable effective learning of temporal and structural patterns.

3.2 Task-Aware MoE Framework

The router function of traditional Mixture-of-Experts relies solely on input features. Such a design not only restricts expert activation to a limited subset of experts but also overlooks the inter-task correlations. In this section, we introduce Task-Aware Mixture of Experts (TAMoE), a novel framework that integrates task-aware routing into expert selection. By conditioning routing decisions on both token representations and task embeddings, TAMoE achieves more balanced expert utilization and effectively leverages inter-task correlations.

Task-Aware MoE Model Design. As shown in Figure 2, the Task-Aware MoE-based Model consists of two parts: Task-Aware Traffic Encoder f_θ, and Task-Specific Prediction Head g_t. First, since models operate in continuous vector spaces, effective representation requires both token semantics and positional context. To this end, we adopt a learnable token embedding layer that projects each discrete traffic token into a continuous $d_{model} = 192$ dimensional vector. Besides, we use a fixed sinusoidal positional encoding scheme, which adds a unique positional vector to each token embedding. Then, the resulting embedding sequence is processed by a stack of task-aware MoE blocks, with the number of blocks set to $N = 4$ in this study. Each task-aware MoEBlock is composed of two sub-layers: a multi-head self-attention and a TAMoE layer. RMS normalization and residual connections are applied to each sub-layer to ensure training stability.

To accommodate multiple tasks training, we append a separate, lightweight classification head for each task after the Task-Aware Traffic Encoder f_θ. Each head g_t is typically a

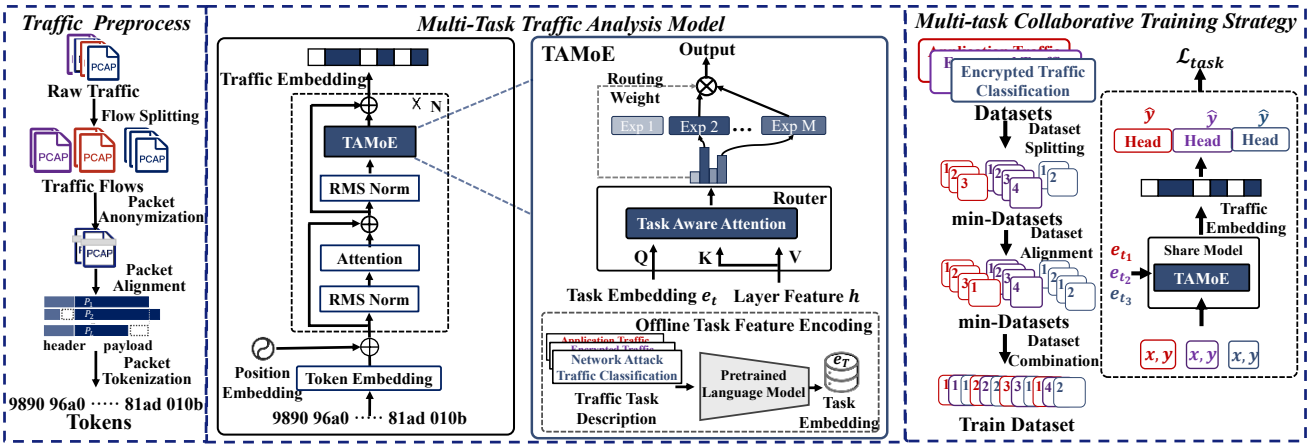

Figure 2: Overview of the TAMoE framework. (Left) A preprocessing pipeline converts raw traffic into tokenized sequences. (Center) A multi-task model based on the TAMoE layer, which feeds the corresponding task embedding and sample features to the router. (Right) A collaborative training strategy combines data from multiple tasks to train the shared model jointly.

simple Multi-Layer Perceptron (MLP) that takes the final representation from the encoder and maps it to the task-specific output space. This modular design allows the traffic encoder to serve as a powerful, general-purpose feature extractor. This encoder can be fine-tuned using a small set of labeled data and transferred to various tasks.

Task Description Embedding. To capture the semantic relationships between traffic analysis tasks, we adopt a pretrained language model embedding that extracts task-descriptive semantic features as Task Embedding e_t. This approach enriches the router process with valuable semantic information, enhancing its ability to select experts more effectively for each task. Specifically, we begin by defining a textual description for each traffic task t. This description (e.g., "Darknet Network Attack Traffic Classification") is then treated as a textual prompt and encoded using a Pre-trained Language Model, such as BERT [Devlin *et al.*, 2018], to extract the task embedding, e_t. represented as: $e_t = \text{LanguageModel}(t)$. Notably, these task embeddings are pre-computed and stored, ensuring zero additional computational cost during the actual training or inference stages.

Task-Aware Routing Mechanism. For enabling expert selection based on task-specific semantic features and adaptation across diverse tasks, we propose a Task-Aware routing mechanism. This module leverages the Task Embedding as the query (Q) and applies it to the input feature representations,h, which serve as both keys (K) and values (V). The resulting attention weights reflect the semantic relevance between the task intent and input features, thereby measuring which feature positions are most informative.

$$\begin{aligned} &\text{TaskAttention}(e_t, h) = softmax\left(\frac{Q_t K^\top}{\sqrt{d_k}}\right) V \\ &Q_t = e_t * W_Q, K = h * W_k, V = h * W_v \end{aligned} \tag{1}$$

These W are learnable parameters that project task embeddings and input features into a unified attention space. The attention output represents the alignment of input features with task semantics while simultaneously serving as a routing signal that dynamically guides feature selection and expert allocation based on task intent. Next, the output z is fed into a linear layer to project to expert dimension.

$$\begin{aligned} z &= \text{TaskAttention}(e_t, h) \\ \ell &= W_r z + b_r, \ell \in \mathbb{R}^M \end{aligned} \tag{2}$$

where W_r and b_r are the parameters of the linear layer. M represents the number of experts. Then, we apply a **top-k sparse router** strategy by computing the softmax only over the logits of the top-k experts, thereby determining the expert activation distribution.

$$r_i = \begin{cases} \frac{\exp(\ell_i)}{\sum_{j \in \text{Top-}k} \exp(\ell_j)}, & i \in \text{Top-}k \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

After obtaining the weighted output of each token from the selected experts, we add a residual connection and apply layer normalization to produce the final feature representation. Each expert Expert_i receives the same hidden layer input h and produces an output with the same dimensionality. Finally, we aggregate the outputs of the top-k experts by weighting them with the router scores.

$$out_i = \text{Expert}_i(h) \quad out = \sum_{i=1}^{M} r_i \cdot out_i \tag{4}$$

Unlike conventional MoE that rely on auxiliary balancing losses, or routing regularization to prevent collapse, TAMoE is trained without any such artificial constraints. By incorporating task features, this design encourages the routing network to make more diverse expert selections during multi-task training. This promotes a more balanced activation across the expert pool, which mitigates the risk of expert collapse. While simultaneously enabling the router to learn and leverage the semantic relationships between tasks.

3.3 Multi-Task Collaborative Training Strategy

Multi-Task Dataset Mixed Construction. Ensuring effective representation across multiple network traffic tasks and

avoiding bias due to sample imbalance, we adopt a multi-task collaborative training strategy based on balanced and aligned dataset mixing. Let there be $t \in T$ tasks, each associated with a dataset $\mathcal{D}^t = (x_1^t, y_1^t), \ldots, (x_{N_t}^t, y_{N_t}^t)$, where $N_t = |\mathcal{D}^t|$. Each dataset is first partitioned into equal-sized chunks:

$$\mathcal{D}^t = \bigcup_{i=1}^{m} \mathcal{D}_i^t, \quad \text{with} \quad |\mathcal{D}_i^t| = B \tag{5}$$

To unify the number of chunks across all tasks, we duplicate and expand smaller datasets until each task has $m = \max_t \lceil N_t/B \rceil$ segments. This ensures that each task is equally represented during training. During training, we interleave these task-specific segments to form multi-task mixed batches: $\mathcal{B}_i = \left[\mathcal{D}_i^1, \mathcal{D}_i^2, \ldots, \mathcal{D}_i^T\right], i = 1, \ldots, m$. This alternating mixing strategy ensures that all tasks are fairly represented during each training round, so the model doesn't focus too much on one particular task. By combining data from different tasks evenly, the model can learn both shared patterns across tasks and features unique to each task.

Multi-Task Training Step. For each input batch, the identity of the current task t is encoded into a fixed-length vector e_t using the Task Description Embedding, which can be precomputed and stored for all tasks. During training or inference, e_t can be retrieved in real time, avoiding repeated computation. Each input sample x_i^t is passed through the Task-Aware Traffic Encoder f_θ to produce a sequence of token-level feature representations. Specifically, the task vector e_t is injected into the TAMoE of the task-aware MoE block as part of the router input. The router computes expert scores using both the token-level features and e_t. The top-k most relevant experts are selected via sparse routing. The selected experts process the token features, generating transformed feature representations that are adaptive to both the input traffic and task semantics. The global feature vector produced by the MoE block is fed into the task-specific classification head g_t to generate predictions $\hat{y}_i^t = g_t(f_\theta(x_i^t, e_t))$. For multi-task training, the total loss is computed as the sum of task-specific losses across all tasks:

$$\mathcal{L}_{total} = \sum_{t=1}^{T} \mathcal{L}_{task}^t\left(\hat{y}_i^t, y_i^t\right) \tag{6}$$

where $\mathcal{L}_{task}$ denotes the cross-entropy loss function. This training strategy optimizes multiple tasks simultaneously, enabling TAMoE to learn diverse task-specific semantic features as well as the underlying correlations among tasks.

4 Experiments

4.1 Experiment Setup

Dataset and Baseline

We evaluate TAMoE on six widely adopted datasets covering three distinct traffic analysis domains: Application Classification (CrossPlatform-Android and CrossPlatform-iOS [Ren *et al.*, 2019]), Network Intrusion Detection (CICIoT2022 [Dadkhah *et al.*, 2022], USTCTFC2016 [Wang *et al.*, 2017]), and Encrypted Traffic Classification (ISCXVPN2016 [Draper-Gil *et al.*, 2016], ISCXTor2016 [Lashkari *et al.*, 2017]). For benchmarking, we compare TAMoE against state-of-the-art methods across four major paradigms: (1) Machine Learning (e.g., AppScanner [Taylor *et al.*, 2016], FlowPrint [van Ede *et al.*, 2020]), (2) Deep Learning (e.g., DeepPacket [Lotfollahi *et al.*, 2020], FS-Net [Liu *et al.*, 2019]), (3) Pre-training Frameworks (e.g., ET-BERT [Lin *et al.*, 2022], YaTC [Zhao *et al.*, 2023], NetMamba [Wang *et al.*, 2024a] and TrafficFormer [Zhou *et al.*, 2025]), and (4) MoE-based approaches. This comprehensive setup ensures a rigorous assessment of TAMoE's performance and generalization.

Evaluation Metric and Implementation Details

We used accuracy (ACC) and macro-averaged F1-score (F1) as evaluation metrics. We employed the pre-trained BERT model, BERT-Base-Uncased, to generate task-specific embeddings. We employed a TAMoE Encoder with 4 layers as the traffic encoder model, where we configured the TAMoE component with 6 experts and selected the top 2 most relevant experts for each sample. Training was conducted with a batch size of 128. We used a cosine annealing learning rate scheduler with an initial learning rate of 0.0125 and a weight decay of 1e-05. In the fine-tuning stage, we employed a learning rate of 0.03. Experiments were performed using Pytorch 2.0.0 and on an NVIDIA® 3090 24G GPU.

4.2 Performance against Baselines

As shown in Table 1, TAMoE consistently achieves state-of-the-art performance across all six datasets, outperforming classical machine learning, deep learning, pre-trained, and MoE-based baselines on both accuracy and F1-score.

(1) Comparison to Machine Learning Approaches. On *CrossPlatform-Android*, TAMoE boosts the F1-score from FlowPrint's 0.7565 to 0.9835, showcasing the limitations of hand-crafted features and the clear advantages of expert-driven traffic modeling. **(2) Comparison to Deep Learning Approaches.** TAMoE consistently delivers higher accuracy and F1-scores; for example, *on CrossPlatform-iOS*, it increases the F1-score from DeepPacket's 0.9034 to 0.9787. This gain comes from task-aware routing, which encourages more robust and transferable representations than conventional single-stream architectures. **(3) Comparison to Pre-train and Finetune Models.** Pretrained models offer strong baselines, yet their unified representations limit cross-task adaptability. TAMoE surpasses them across all benchmarks—for instance, improving the F1-score on CrossPlatform-Android from NetMamba's 0.9260 to 0.9835. The improvement reflects TAMoE's ability to leverage multiple specialized experts to capture more diverse traffic patterns. **(4) Comparison to MoE-based Approaches.** Compared with the Multi-Router MoE, TAMoE achieves higher scores on every dataset. On CrossPlatform-Android, it improves the F1-score from 0.9731 to 0.9835. These results confirm that task-aware routing and attention enable more effective expert utilization, resulting in more accurate and reliable multi-task traffic analysis.

Overall, the results demonstrate that TAMoE not only improves classification accuracy but also enhances robustness across heterogeneous platforms, varied encryption protocols, and diverse real-world traffic distributions.

Method	CrossPlatform-Android		CrossPlatform-IOS		CICIoT2022		USTCTFC2016		ISCXTor2016		ISCXVPN2016	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
Machine Learning Based Approach												
AppScaner [Taylor *et al.*, 2016]	0.3868	0.3866	0.3205	0.3130	0.7556	0.6938	0.6998	0.6633	0.4034	0.2113	0.7643	0.7256
FlowPrint [van Ede *et al.*, 2020]	0.8739	0.8700	0.8712	0.8603	0.5820	0.4643	0.7992	0.7755	0.1316	0.0306	0.9666	0.9681
Deep Learning Based approach												
Deeppacket [Lotfollahi *et al.*, 2020]	0.8805	0.8138	0.9204	0.9034	-	-	0.9640	0.9641	0.7449	0.7473	0.9329	0.9321
FS-Net [Liu *et al.*, 2019]	0.4631	0.4628	0.3884	0.3884	0.4391	0.4375	0.4542	0.4542	0.7198	0.7198	0.7322	0.7322
TFE-GNN [Zhang *et al.*, 2023]	0.8141	0.8067	0.8241	0.8130	0.9915	0.9907	0.9747	0.9734	0.7692	0.7618	0.8428	0.8447
Pre-train and Finetune Based Approach												
ET-BERT [Lin *et al.*, 2022]	0.9166	0.9071	0.9105	0.8850	0.9937	0.9937	0.9915	0.9913	0.9980	0.9980	0.9566	0.9565
YaTC [Zhao *et al.*, 2023]	0.9074	0.9074	0.9562	0.9562	0.9584	0.9583	0.9786	0.9785	0.9890	0.9892	0.9772	0.9769
NetMamba [Wang *et al.*, 2024a]	0.9257	0.9260	0.9248	0.9242	0.9944	0.9944	0.9982	0.9982	0.9993	0.9991	0.9826	0.9826
TrafficFormer [Zhou *et al.*, 2025]	0.9473	0.9472	0.9468	0.9466	0.9931	0.9931	0.9957	0.9955	0.9953	0.9951	0.9783	0.9883
MoE Based Approach												
Multi-Router MoE	0.9734	0.9731	0.9618	0.9618	0.9949	0.9949	0.9904	0.9902	0.9917	0.9915	0.9854	0.9852
TAMoE(our)	**0.9836**	**0.9835**	**0.9789**	**0.9787**	**0.9964**	**0.9965**	**0.9988**	**0.9987**	**0.9996**	**0.9996**	**0.9877**	**0.9877**

Table 1: Model performance under different datasets. Top-3 best methods are underlined, and Top-1 best methods are **bold**.

4.3 Expert Specialization

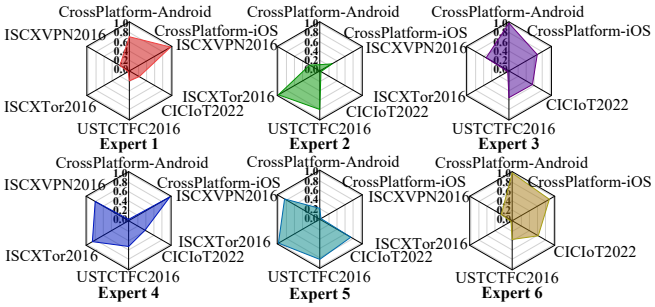

Figure 3: Radar charts of task preferences for six experts. The value on the axis is the evaluation score normalized using Min-Max scaling for each expert individually, highlighting relative specialization patterns rather than absolute performance.

To further understand how TAMoE leverages expert specialization, we analyzed the allocation behavior of each expert by examining the number of tokens they processed for different tasks. As illustrated in Figure 3, we visualize the relative task preferences of six experts using radar charts. The radar plots reveal highly diverse and non-overlapping preference profiles, indicating that no expert dominates across all tasks. Instead, each expert develops its own distinct competency region, shaped by the task-aware routing. This confirms that the router does not collapse to uniform routing but successfully guides tasks toward the most compatible experts.

More concretely, Experts 1, 3, and 6 exhibit strong affinity for application-level classification tasks, consistently showing the highest normalized scores on CrossPlatform-Android and CrossPlatform-iOS. However, these experts perform weakest on encrypted traffic datasets such as ISCXTor2016, suggesting that their learned representations focus on application semantics rather than encrypted flow characteristics. In contrast, other experts (e.g., Experts 4 and 5) show complementary patterns, excelling on encrypted traffic tasks while placing lower emphasis on application-oriented ones.

Overall, these findings indicate pronounced expert heterogeneity and specialization, validating that TAMoE successfully decomposes complex multi-task traffic analysis into a coordinated ensemble of task-aligned experts.

4.4 Expert Load Balancing

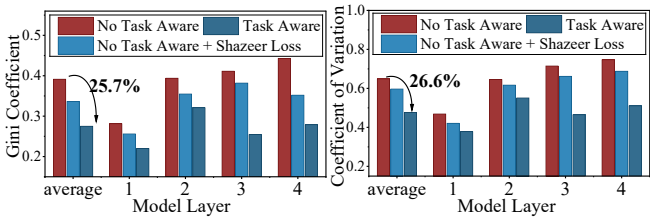

Figure 4: Evaluation of expert load balancing, comparing our TaMoE(Task-Aware) MoE against a baseline No Task Aware and No Task Aware + Shazeer Loss across model layers.

As shown in Figure 4, we report two standard inequality metrics, the Gini Coefficient [Dorfman, 1979] and the Coefficient of Variation (CV) [Dou *et al.*, 2024], where lower values indicate a more balanced workload.

Across all layers, TAMoE achieves the lowest Gini and CV among the three variants (No Task Aware, No Task Aware + Shazeer Loss [Fedus *et al.*, 2022], and our Task-Aware MoE). On average, the Gini score drops from 0.391 (No Task Aware) to 0.275 with TAMoE, while the average CV decreases by 26.6%. Although the Shazeer loss partially alleviates imbalance, it still lags behind TAMoE, indicating that task-aware routing provides additional, complementary gains. Notably, the advantage of TAMoE is more pronounced in deeper layers: while the gap is relatively small in Layer 1, the differences in Layers 3 and 4 are significantly larger. This suggests that task-conditioned routing is particularly beneficial when representations become more task-specific. These findings demonstrate that TAMoE substantially mitigates expert load imbalance and reduces the risk of expert collapse.

4.5 Computational Efficiency

As shown in Table 2, we evaluate computational efficiency on the NVIDIA® JETSON AGX ORIN edge device (batch size = 128). TAMoE achieves an inference latency of

Method	Parameters	FLOPs	Times (ms)
TFE-GNN	4.43e+07	2.36e+09	2.12e+00
ET-BERT	1.87e+08	2.25e+10	1.93e+00
YATC	2.20e+06	1.81e+09	4.85e-01
NetMamba	2.30e+06	3.66e+09	3.42e-01
Multi-Router MoE	7.70e+06	1.37e+09	1.73e-01
TAMoE	7.92e+06	1.42e+09	1.91e-01

Table 2: Model parameters, FLOPs and time overhead.

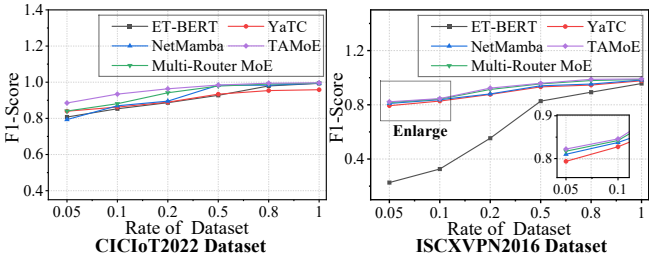

Figure 5: Model performance under different dataset proportions in the CICIoT2022 dataset and ISCXVPN2016 dataset.

1.91e–01 ms, outperforming TFE-GNN (2.12e+00 ms), ET-BERT (1.93e+00 ms), and lightweight models such as YATC. This represents a 60.6% reduction in latency compared with YATC. Although TAMoE has more parameters than YATC (7.92e+06 vs. 2.20e+06), its conditional MoE computation ensures that only a small subset of experts is activated per sample, substantially lowering the computational cost. Compared with MultiRouterMoE, TAMoE introduces additional parameters due to task-aware routing, yet it maintains a single shared model for all tasks, eliminating the need for multiple task-specific deployments.

4.6 Few-Shot Evaluation

To evaluate the cross-task transfer ability, we first train the feature extractor on five datasets using the Multi-Task Collaborative Training Strategy and then fine-tune it on the remaining dataset. As shown in Figure 5, TAMoE exhibits clear advantages in low-resource settings. On CICIoT2022, with only 5% labeled data, TAMoE achieves an F1-score of 0.8851, surpassing all baselines, including ET-BERT (0.8074), NetMamba (0.7934), YATC (0.8395), and Multi-Router MoE (0.8404). This reflects that TAMoE can reach great performance with only a small set of labels.

TAMoE maintains its performance lead as the dataset size increases. At the 10% proportion, TAMoE reaches 0.9341, significantly outperforming the next-best baseline (0.8810). Although all methods converge with the full dataset, TAMoE's robustness under sparse labels is particularly valuable for edge-scale scenarios.

4.7 Ablation Studies

Robustness of Task Description

To verify robustness, we evaluate TAMoE on CICIoT2022 under four varied task descriptions, including vague and noisy phrasing. As summarized in Table 3, the performance is extremely stable, with $\leq 0.06\%$ variance in F1-score across all descriptions, indicating that the router is not sensitive to the specific wording.

ID	Description Type	F1
T1	Standard description	0.9965
T2	Synonym-rich paraphrase	0.9961
T3	Vague / under-specified	0.9963
T4	Noisy (typos / extra words)	0.9959

Table 3: Impact of task description on TAMoE on CICIoT2022.

Method	ACC	F1
TAMoE (default)	**0.9964**	**0.9966**
w/o MoE(Transform)	0.9584	0.9583
w/o Position Embedding	0.9736	0.9733
w/o Collaborative Training	0.9764	0.9677
w/o TaskAware	0.9341	0.9338
One-Hot Task Embedding	0.9764	0.9759
Multi-Task Shared-Transformer-Encoder	0.9518	0.9517

Table 4: Ablation on TAMoE settings.

Other Ablation Settings

As shown in Table 4, we evaluate the contribution of each component on *ISCXTor2016*.

1. w/o MoE (Transform). Replacing the MoE framework with a standard Transformer causes a clear drop (F1: 0.9966 $\rightarrow$ 0.9583), showing that a single shared representation is insufficient for heterogeneous traffic, while expert specialization is beneficial. **2. w/o Position Embedding.** Removing positional information degrades performance (F1: 0.9966 $\rightarrow$ 0.9733), confirming the importance of temporal order within flows. **3. w/o Collaborative Training.** Training on a single dataset instead of jointly across tasks reduces F1 to 0.9677, indicating that inter-task knowledge transfer is crucial for generalization. **4. w/o TaskAware.** Replacing the task-aware router with a conventional multi-router MoE leads to a substantial drop (F1: 0.9966 $\rightarrow$ 0.9338), highlighting the benefit of task-informed routing for effective expert utilization. **5. One-Hot Task Embedding.** Using one-hot task vectors instead of learned embeddings lowers F1 to 0.9759, implying that richer task representations better capture inter-task relations. **6. Multi-Task Shared-Transformer-Encoder.** All tasks share a single Transformer encoder with task-specific heads achieves only 0.9517 F1, far below TAMoE. This suggests that a fully shared encoder suffers from negative transfer due to conflicting gradients, whereas TAMoE's sparse, conditional computation enables expert specialization and mitigates such interference. Overall, these results demonstrate that each component plays a crucial role in achieving the superior performance of TAMoE.

5 Conclusion

In this work, we introduced TAMoE, a Task-Aware Mixture-of-Experts framework for multi-task network traffic analysis. By incorporating task semantics into the routing process, TAMoE enables dynamic expert selection based on both traffic features and task-level information, thereby mitigating expert collapse, improving cross-task knowledge sharing, and enhancing transferability to unseen tasks.

Acknowledgments

This work was supported by the Science and Technology Programme of STATE GRID Corporation of China (5100-202456026A-1-1-ZN).

References

[Agarwal *et al.*, 2024] Ishita Agarwal, Aanchal Singh, Aran Agarwal, Shruti Mishra, Sandeep Kumar Satapathy, Sung-Bae Cho, Manas Ranjan Prusty, and Sachi Nandan Mohanty. Enhancing road safety and cybersecurity in traffic management systems: Leveraging the potential of reinforcement learning. *IEEE Access*, 12:9963–9975, 2024.

[Arp *et al.*, 2022] Daniel Arp, Erwin Quiring, Feargus Pendlebury, Alexander Warnecke, Fabio Pierazzi, Christian Wressnegger, Lorenzo Cavallaro, and Konrad Rieck. Dos and don'ts of machine learning in computer security. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 3971–3988, Boston, MA, August 2022. USENIX Association.

[Cai *et al.*, 2025] Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. A survey on mixture of experts in large language models. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[Chen *et al.*, 2024] Wei Chen, Yuxuan Liu, Zhao Zhang, Fuzhen Zhuang, and Jiang Zhong. Modeling adaptive inter-task feature interactions via sentiment-aware contrastive learning for joint aspect-sentiment prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17781–17789, 2024.

[Dadkhah *et al.*, 2022] Sajjad Dadkhah, Hassan Mahdikhani, Priscilla Kyei Danso, Alireza Zohourian, Kevin Anh Truong, and Ali A. Ghorbani. Towards the development of a realistic multidimensional iot profiling dataset. In *2022 19th Annual International Conference on Privacy, Security, Trust (PST)*, pages 1–11, 2022.

[Dandamudi *et al.*, 2025] Sai Ratna Prasad Dandamudi, Jaideep Sajja, and Amit Khanna. Leveraging artificial intelligence for data networking and cybersecurity in the united states. *International Journal of Innovative Research in Computer Science and Technology*, 13:34–41, 2025.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Dorfman, 1979] Robert Dorfman. A formula for the gini coefficient. *The review of economics and statistics*, pages 146–149, 1979.

[Dou *et al.*, 2024] Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, et al. Loramoe: Alleviating world knowledge forgetting in large language models via moe-style plugin. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1932–1945, 2024.

[Draper-Gil *et al.*, 2016] Gerard Draper-Gil, Arash Habibi Lashkari, Mohammad Saiful Islam Mamun, and Ali A Ghorbani. Characterization of encrypted and vpn traffic using time-related. In *Proceedings of the 2nd international conference on information systems security and privacy (ICISSP)*, pages 407–414, 2016.

[Fedus *et al.*, 2022] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2022.

[He *et al.*, 2020] Hong Ye He, Zhi Guo Yang, and Xiang Ning Chen. Pert: Payload encoding representation from transformer for encrypted traffic classification. In *2020 ITU Kaleidoscope: Industry-Driven Digital Transformation (ITU K)*, pages 1–8, 2020.

[Jacobs *et al.*, 2022] Arthur S. Jacobs, Roman Beltiukov, Walter Willinger, Ronaldo A. Ferreira, Arpit Gupta, and Lisandro Z. Granville. Ai/ml for network security: The emperor has no clothes. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 1537–1551, New York, NY, USA, 2022. Association for Computing Machinery.

[Javaheri *et al.*, 2023] Danial Javaheri, Saeid Gorgin, Jeong-A Lee, and Mohammad Masdari. Fuzzy logic-based ddos attacks and network traffic anomaly detection methods: Classification, overview, and future perspectives. *Information Sciences*, 626:315–338, 2023.

[Lashkari *et al.*, 2017] Arash Habibi Lashkari, Gerard Draper Gil, Mohammad Saiful Islam Mamun, and Ali A Ghorbani. Characterization of tor traffic using time based features. In *International Conference on Information Systems Security and Privacy*, volume 2, pages 253–262. SciTePress, 2017.

[Lian *et al.*, 2025a] Xinglin Lian, Chengtai Cao, Yan Liu, Xovee Xu, Yu Zheng, and Fan Zhou. Facing anomalies head-on: Network traffic anomaly detection via uncertainty-inspired inter-sample differences. In *Proceedings of the ACM on Web Conference 2025 (WWW)*, pages 3908–3917, 2025.

[Lian *et al.*, 2025b] Xinglin Lian, Yu Zheng, Zhangxuan Dang, Chunlei Peng, and Xinbo Gao. Semi-supervised anomaly traffic detection via multi-frequency reconstruction. *Pattern Recognition*, 161:111215, 2025.

[Lian *et al.*, 2026a] Xinglin Lian, Chengtai Cao, Ting Zhong, Yong Wang, Kai Chen, and Fan Zhou. Decompose to understand, fuse to detect: Frequency-decoupled anomaly detection for encrypted network traffic. *arXiv: 2605.02970*, 2026.

[Lian *et al.*, 2026b] Xinglin Lian, Yu Zheng, Yan Liu, Fan Zhou, Chunlei Peng, and Xinbo Gao. Contextual masking distillation for network traffic anomaly detection. *IEEE Transactions on Information Forensics and Security*, 21:1273–1286, 2026.

[Lin *et al.*, 2022] Xinjie Lin, Gang Xiong, Gaopeng Gou, Zhen Li, Junzheng Shi, and Jing Yu. Et-bert: A contextualized datagram representation with pre-training transform-

ers for encrypted traffic classification. In *Proceedings of the ACM Web Conference 2022*, pages 633–642, 2022.

[Liu *et al.*, 2019] Chang Liu, Longtao He, Gang Xiong, Zigang Cao, and Zhen Li. Fs-net: A flow sequence network for encrypted traffic classification. In *IEEE INFOCOM 2019-IEEE Conference On Computer Communications*, pages 1171–1179. IEEE, 2019.

[Liu *et al.*, 2024] Lan Liu, Yongjie Yu, Yafeng Wu, Zhanfa Hui, Jun Lin, and Junhan Hu. Method for multi-task learning fusion network traffic classification to address small sample labels. *Scientific Reports*, 14(1):2518, 2024.

[Lotfollahi *et al.*, 2020] Mohammad Lotfollahi, Mahdi Jafari Siavoshani, Ramin Shirali Hossein Zade, and Mohammdsadegh Saberian. Deep packet: A novel approach for encrypted traffic classification using deep learning. *Soft Computing*, 24(3):1999–2012, 2020.

[Nascita *et al.*, 2024] Alfredo Nascita, Giuseppe Aceto, Domenico Ciuonzo, Antonio Montieri, Valerio Persico, and Antonio Pescapé. A survey on explainable artificial intelligence for internet traffic classification and prediction, and intrusion detection. *IEEE Communications Surveys & Tutorials*, 2024.

[Ren *et al.*, 2019] Jingjing Ren, DJ Dubois, and D Choffnes. An international view of privacy risks for mobile apps, 2019.

[Shao *et al.*, 2019] Guolin Shao, Xingshu Chen, Xuemei Zeng, and Lina Wang. Deep learning hierarchical representation from heterogeneous flow-level communication data. *IEEE Transactions on Information Forensics and Security*, 15:1525–1540, 2019.

[Taylor *et al.*, 2016] Vincent F Taylor, Riccardo Spolaor, Mauro Conti, and Ivan Martinovic. Appscanner: Automatic fingerprinting of smartphone apps from encrypted network traffic. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 439–454. IEEE, 2016.

[Upadhyay *et al.*, 2024] Richa Upadhyay, Ronald Phlypo, Rajkumar Saini, and Marcus Liwicki. Sharing to learn and learning to share; fitting together meta, multi-task, and transfer learning: A meta review. *IEEE Access*, 2024.

[van Ede *et al.*, 2020] Thijs van Ede, Riccardo Bortolameotti, Andrea Continella, Jingjing Ren, Daniel J. Dubois, Martina Lindorfer, David Choffnes, Maarten van Steen, and Andreas Peter. Flowprint: Semi-supervised mobile-app fingerprinting on encrypted network traffic. *Network and Distributed System Security Symposium (NDSS)*, 27, 2020.

[Wang *et al.*, 2017] Wei Wang, Ming Zhu, Xuewen Zeng, Xiaozhou Ye, and Yiqiang Sheng. Malware traffic classification using convolutional neural network for representation learning. In *2017 International conference on information networking (ICOIN)*, pages 712–717. IEEE, 2017.

[Wang *et al.*, 2024a] Tongze Wang, Xiaohui Xie, Wenduo Wang, Chuyi Wang, Youjian Zhao, and Yong Cui. Netmamba: Efficient network traffic classification via pre-training unidirectional mamba. *arXiv preprint arXiv:2405.11449*, 2024.

[Wang *et al.*, 2024b] Ziteng Wang, Jun Zhu, and Jianfei Chen. Remoe: Fully differentiable mixture-of-experts with relu routing. *arXiv preprint arXiv:2412.14711*, 2024.

[Wei *et al.*, 2024] Xiuying Wei, Skander Moalla, Razvan Pascanu, and Caglar Gulcehre. Building on efficient foundations: Effective training of llms with structured feedforward layers. *Advances in Neural Information Processing Systems*, 37:4689–4717, 2024.

[Zhang *et al.*, 2021] Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Moefication: Transformer feed-forward layers are mixtures of experts. *arXiv preprint arXiv:2110.01786*, 2021.

[Zhang *et al.*, 2023] Haozhen Zhang, Le Yu, Xi Xiao, Qing Li, Francesco Mercaldo, Xiapu Luo, and Qixu Liu. Tfe-gnn: A temporal fusion encoder using graph neural networks for fine-grained encrypted traffic classification. In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 2066–2075, New York, NY, USA, 2023. Association for Computing Machinery.

[Zhang *et al.*, 2025] Gong-Duo Zhang, Ruiqing Chen, Qian Zhao, Zhengwei Wu, Fengyu Han, Huan-Yi Su, Ziqi Liu, Lihong Gu, and Lin Zhou. Bagging-expert network for multi-task learning: A depolarization solution in multi-gate mixture-of-experts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22389–22397, 2025.

[Zhao *et al.*, 2023] Ruijie Zhao, Mingwei Zhan, Xianwen Deng, Yanhao Wang, Yijun Wang, Guan Gui, and Zhi Xue. Yet another traffic classifier: A masked autoencoder based traffic transformer with multi-level flow representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5420–5427, 2023.

[Zhou *et al.*, 2022] Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022.

[Zhou *et al.*, 2025] Guangmeng Zhou, Xiongwen Guo, Zhuotao Liu, Tong Li, Qi Li, and Ke Xu. Trafficformer: An efficient pre-trained model for traffic data. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 1844–1860, 2025.