基于长程依赖与局部动态感知的加密流量预训练模型构建

张 磊[1,2*],刘 晓[1,2],张龙港[1,2],何 清[1,2]

(1. 重庆大学微电子与通信工程学院,重庆 400044;2. 重庆市生物感知与多模态智能信息处理重点实验室,重庆 400044)

摘 要: 网络流量表征与分类在网络管理和安全中发挥重要作用。由于仅依赖局部卷积运算,传统的卷积神经网络(Convolutional Neural Network,CNN)已不足以应对当前网络流量的复杂性和变化性。基于Transformer架构的方法在流量序列处理中取得了一定效果,但其自注意力机制带来的平方级计算复杂度导致了较高的资源消耗,且在处理长序列时效率较低,同时该架构未优先考虑时间依赖关系,导致局部序列一致性容易丢失。近期引入的状态空间模型(State-Space Models,SSM)Mamba实现了随序列长度线性扩展的计算效率,在局部动态感知和长序列处理方面具有优势,但由于其将历史信息压缩至固定维度的隐藏状态中,限制了对长距离依赖和全局流量特征的获取能力。针对上述问题,本文提出一种基于长程依赖与局部动态感知的加密网络流量预训练模型(Encrypted Traffic pre-trained model with Long-range Dependency and dynamic Perception,ET-LDP)。该模型摒弃了传统的架构拼接,选择依据网络流量包含流级别全局统计模式与字节级别局部语义的多粒度特性,从流量特征的角度将多种类型的模块进行集成,构建了跨机架构协同结构。ET-LDP在Mamba层内部引入混合专家(Mixture of Experts,MoE)机制,通过共享参数与私有门控相结合的路由策略,模型将不同语义的token动态分配给特定专家处理,在保持较低计算开销的同时提高了模型容量与局部特征的感知能力。在提取全局长程依赖方面,模型设计了混合注意力模块,首先通过代理注意力(Agent Attention)利用代理token聚合局部信息并减少冗余计算,然后结合softmax自注意力完成细粒度的上下文关联提取。为解决Mamba模块与注意力模块在信息流向与特征空间上的差异,本文进一步设计了跨架构桥接组件,通过信息交互机制实现了两类异构特征的结构性对齐与相互补充。这种灵活的架构允许针对特定资源和目标进行配置,从而兼顾各自的优势。考虑到网络流量和网络任务的多样性,ET-LDP结合自监督机制,从大规模无标签流量数据中预训练通用表征。在虚拟专用网络(Virtual Private Network,VPN)、洋葱路由器(The onion router,Tor)、恶意软件及跨平台应用等7个公开网络流量数据集的评估表明,本文提出的ET-LDP模型在分类性能上均优于目前的最优(State Of The Art,SOTA)方法。在采用新加密协议的中国科技网TLS 1.3流量(China Science and Technology NETwork-TLS 1.3,CSTNET-TLS 1.3)数据集上,相较于同类最优的模型准确率提升了0.013 2。在效率测试中,当批大小为64时,该模型的推理显存占用为同等规模Transformer模型的0.6倍,在吞吐量与计算延迟上表现出较好的平衡。

关键词: 网络流量分类;Mamba-Transformer混合模型;预训练;混合专家(MoE);状态空间模型(SSM);深度学习

基金项目: 国家自然科学基金(No.92570110,No.62271090);重庆市自然科学基金(No.CSTB2024NSCQ-JQX0038)

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112(2026)04-1835-22

电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20260269

Encrypted Traffic Pre-Trained Model with Long-Range Dependency and Local Dynamic Perception

ZHANG Lei[1,2*], LIU Xiao[1,2], ZHANG Longgang[1,2], HE Qing[1,2]

(1. *School of Microelectronics and Communication Engineering*, *Chongqing University*, *Chongqing* 400044, *China*;
2. *Chongqing Key Laboratory of Bio-Perception & Multimodal Intelligent Information Processing*, *Chongqing* 400044, *China*)

Abstract: Network traffic representation and classification plays an important role in network security and management. Due to local convolution operations, traditional convolutional neural networks (CNN) are insufficient to handle the complexity and variability of modern network traffic. Although Transformer-based methods have achieved certain results in traffic sequence processing, the quadratic computational complexity of their self-attention mechanisms leads to high resource consumption and low efficiency when processing long sequences. Meanwhile, this architecture does not prioritize time dependencies, leading to a loss of local sequence consistency. The recently introduced state-space model (SSM) Mam-

收稿日期:2026-03-02;录用日期:2026-04-14;责任编委:赵瑞杰

*通信作者:张磊

ba achieves computational efficiency that scales linearly with sequence length, showing advantages in local dynamic perception and long-sequence processing. However, because it compresses historical information into a fixed-dimensional hidden state, its ability to capture long-range dependencies and global traffic features is limited. To address these issues, this paper proposes encrypted traffic pre-trained model with long-range dependency and dynamic perception (ET-LDP), an encrypted network traffic pre-trained model based on long-range dependency and local dynamic perception. Instead of traditional architecture concatenation, the model integrates multiple types of modules from the perspective of traffic characteristics. Based on the multi-granularity nature of network traffic, which includes flow-level global statistical patterns and byte-level local semantics, a cross-architecture collaborative structure is constructed. ET-LDP introduces the mixture-of-experts (MoE) mechanism inside the Mamba layer. Through a routing strategy combining shared parameters and private gating, the model dynamically allocates tokens with different semantics to specific experts, which improves model capacity and local feature perception capabilities while maintaining a low computational overhead. For extracting global long-range dependencies, the model designs a mixed attention module. It first utilizes Agent Attention to aggregate local information and reduce redundant computation via agent tokens, and then combines softmax self-attention to complete fine-grained context association extraction. To solve the differences in information flow and feature space between the Mamba module and the attention module, this paper further designs cross-architecture bridging components, realizing structural alignment and mutual complementation of the two heterogeneous features through an information interaction mechanism. This flexible architecture allows configuration for specific resources and targets. Considering the diversity of network traffic and tasks, ET-LDP adopts a self-supervised learning mechanism to pre-train generalized representations from large-scale unlabeled traffic data. Evaluations on 7 public network traffic datasets, covering virtual private network (VPN), the onion router (Tor), malware, and cross-platform applications, indicate that the proposed ET-LDP model outperforms state of the art (SOTA) methods in classification performance. On the China science and technology network-TLS 1.3 (CSTNET-TLS 1.3) dataset using new encryption protocols, the accuracy is improved by 0.013 2 compared to the best-performing counterpart. In efficiency tests, at a batch size of 64, the inference memory consumption of the model is 0.6 times that of similar Transformer models, showing a good balance in throughput and computational latency.

Keywords: network traffic classification; Mamba-Transformer hybrid model; pre-training; mixture of experts (MoE); state-space models (SSM); deep learning

Foundation Item(s): National Natural Science Foundation of China (No.92570110, No.62271090); Chongqing Natural Science Foundation (No.CSTB2024NSCQ-JQX0038)

0 引言

网络流量分类在网络管理和网络安全中发挥着至关重要的作用[1]。通过使用加密[如传输层安全协议(Transport Layer Security, TLS)]和匿名网络技术[如虚拟专用网络(Virtual Private Network, VPN)、洋葱路由器(The onion router, Tor)]分析复杂流量,这些系统可以捕获流量的准确表示,从而增强流量分类能力。这不仅有利于提高各种服务的质量,还可以显著提高网络系统的安全性[2]。目前,网络流量分类已从传统方法发展到深度学习模型。早期的网络流量分类方法主要依赖于人工设计的特征或统计属性[3-5]。然而,这类方法无法准确捕捉有效的流量特征,且不适用于加密技术快速发展的当前网络环境。随着深度学习的兴起,研究人员开始利用卷积神经网络(Convolutional Neural Network, CNN)来提升性能[6-7]。然而,CNN在捕捉序列顺序信息方面存在局限性,容易产生偏差,计算效率也相对较低。基于Transformer的方法,如Trafficformer(面向流量数据的高效预训练模型)[8]、TrafficLLM(基于通用流量表示与大型语言模型的增强型网络流量分析模型)[9]和ET-BERT(基于预训练变换器的上下文数据报表示)[10],与CNN相比拥有强大的序列建模能力,且无需依赖固定窗口大小。此外,自注意力机制不会压缩上下文信息,即其可捕捉序列中任意位置元素之间的依赖关系。因此,采用基于Transformer的方法进行捕捉网络流量特征是当前流行的一种方案。尽管该方法具有这些优势,但其平方复杂度使得这些方法在计算上仍然具有挑战性。近期的状态空间模型(State-Space Models, SSM),如Mamba[11],其独特的选择机制使其能够根据输入序列过滤噪声并保留相关信息,实现了序列长度的线性扩展,更有利于处理长距离关系。因此,对于需要具有一定顺序关系的网络流量数据,Mamba更能发挥其优势。然而,由于流量数据本身具有多层次的信息,每一个数据流可以拆分为头部与负载等部分,如Mamba类的SSM逐步处理数据,限制了其联系局部信息与全局上下文的能力。由于这些缺陷,Mamba在性能上仍然落后于同等规模的Transformer模型。

使用自监督机制从大量无标签的流量数据中进

行预训练的方法已经显示出良好的性能。基于Transformer的预训练网络流量模型[12-14]与传统的监督学习范式相比取得了更好的性能。然而,这些现有模型面临一些重大挑战。采用Transformer架构的方法在处理流量数据时,由于Transformer的设计本身并未优先考虑时间依赖性,因此难以保持局部序列的一致性。相比之下,Mamba模型擅长对序列顺序和时间依赖性进行建模,能够逐步处理序列。这种设计对于具有一定顺序关系的网络流量数据十分有利。此外,其采用将所有历史信息压缩到历史状态中的记忆方式,内存开销较小且固定。但Mamba的历史信息压缩特性导致其隐藏状态只能在有限的范围内捕捉这种关系,而流量具有层次结构,体现在包级和流级别上。因此,Mamba可能无法捕获更广泛的流模式,这限制了其在具有多层次信息的流量数据中的表现。现在已有一些其他领域的混合Transformer-Mamba架构的研究[15-19],通过考察各种混合集成模式,最终表明混合Attention-Mamba的性能优于纯模型,同时吞吐量也高于原始Transformer。

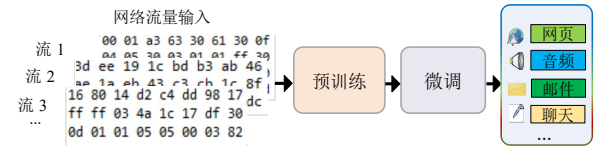
综合以上考虑,本文提出了一种基于长程依赖与局部动态感知的加密流量预训练模型(Encrypted Traffic pre-trained model with Long-range Dependency and dynamic Perception, ET-LDP)。如图1所示,ET-LDP采用Mamba-Transformer混合架构从原始流量中提取流量信息,并以Patch Embedding形式输入模型。ET-LDP采用掩码自动编码器(Masked Autoencoder, MAE)结构,在大量未标注数据集上进行自监督预训练,并在少量标注数据上进行微调。在重建掩码过程中,采用堆叠双向Mamba层的方法替代传统大型解码器来保持预训练阶段的轻量性。为了使Mamba模块具备与Transformer相近的活跃参数和容量,ET-LDP在部分Mamba层中添加了混合专家(Mixture-of-Experts, MoE)架构。这种灵活的架构允许针对特定资源和目标进行配置,允许模型利用两种神经网络的优势互补来克服单一模型的固有缺陷。一方面,Mamba凭借其线性复杂度的递归选择机制,有效解决了单一Transformer模型的计算瓶颈,并能捕捉流量序列顺序和时间依赖性,弥补了Transformer设计中对序列顺序不敏感的短板;另一方面,Transformer利用具备全局感受野的自注意力机制,突破了Mamba在有限的范围内将历史信息压缩至固定维度隐状态带来的层级关系间的精细关联建模约束。此外,采用Agent注意力[20]和softmax自注意力作为混合注意力模块的关键组件,整合了强大的自注意力机制和高效的线性注意力机制的双重优势。然而,若使Mamba与Transformer这两种异质性架构高效融合,则需综合考虑二者的信息交互方式。通过仔细测试发现,最佳融合位置存在于两种模块交替的中间层。基于此,本文采用不同的融合策略处理两种方向的信息流,促进Mamba与Transformer模块在表达空间上的协同建模。这些改进可确保ET-LDP在计算效率和性能间取得良好的平衡。

综上所述,本文的贡献如下:

(1)提出了ET-LDP加密网络流量预训练模型,探索了Mamba与Transformer混合模型在网络流量分类中的潜力。

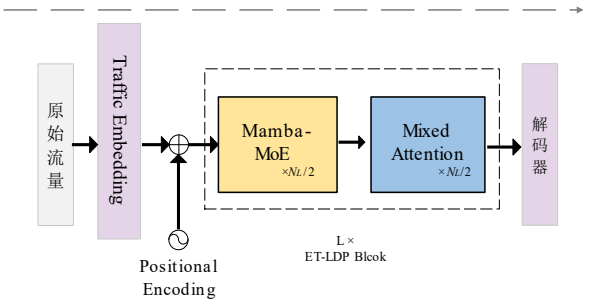
(2)构建了面向流量多粒度特征的跨机制协同组件。针对流量结构的异构性,在Mamba层中引入MoE机制,实现了特征的动态路由与局部感知增强;同时巧妙设计了结合Agent注意力与softmax注意力的协同模块,以高效提取全局长程依赖。系统地研究了Mamba和Transformer模块在网络流量分类任务中的集成模式,设计了跨架构桥接模块,实现了异构特征的对齐与优势互补。

(3)进行了详细的实验,并与现有的网络流量模型进行了比较。ET-LDP在网络流量分类的7个数据集中取得了当前最优水平(State Of The Art, SOTA)性能,并实现了性能与效率的平衡。



(a) 网络流量分类

(a) Network traffic classification



(b) ET-LDP混合结构

(b) ET-LDP hybrid architecture

图1　网络流量分类任务示例及ET-LDP混合架构

Figure 1　Network traffic classification task and ET-LDP hybrid architecture

1　相关工作

1.1　基于Transformer的流量分类

随着网络环境的快速发展,流量分类通常需依赖预训练的大型模型将原始流量投影到低维空间以获

得鲁棒特征，从而提升下游任务的性能。基于Transformer的预训练模型已被广泛应用于多种流量任务中。例如，文献[10]提出的ET-BERT模型和文献[21]提出的PERT模型，均基于完整的词汇表对流量进行建模，并针对下游分类任务进行微调。面向网络流量的生成式预训练Transformer模型（Generative Pre-trained Transformer for Network traffic，NetGPT）[22]和Lens（面向网络流量的基础模型）[12]遵循自然语言处理方式，构建网络流量预训练模型，专注于流量理解和生成。然而，此类模型侧重字符级建模，难以充分挖掘流量中的高层语义。为此，文献[23]借鉴计算机视觉领域的MAE方法，提出另一种流量分类器（Yet another Traffic Classifier，YaTC），进一步提升了模型的鲁棒性与泛化能力。尽管上述方法在流量分类任务中取得有效的性能提升，但基于纯Transformer的模型面临着计算和内存效率低下的问题。因此，亟须研究能够权衡性能和效率的流量分类解决方案。

1.2 基于Mamba的特征学习

Mamba是一种新颖的SSM，它将时变参数融合到SSM中，并构建了一种硬件感知算法，以实现高效的训练和推理。近年来，Mamba在计算机视觉[24]、自然语言处理[25]等多个领域取得了显著进展。NetMamba[26]为网络流量分类提出了一种新颖的状态空间预训练模型，该模型取代了Transformer架构，提高了流量序列建模的效率。然而，受限于纯Mamba模型的约束，其性能未能取得更进一步的突破。鉴于Transformer在局部建模方面的优势与Mamba的长序列建模能力，已有研究[15-19]尝试将二者融合构建混合架构模型以互补优势。然而，简单的堆叠或拼接结构忽视了建模机制差异，导致难以充分整合两者优点，且在适配网络流量数据上存在一定局限。与上述通用混合架构不同，本文提出的ET-LDP不同于现有仅依靠残差连接传递信息的方法，本文方法在Mamba与Attention层之间具有跨架构桥接机制，解决了异构语义空间的对齐问题。此外，本文还设计了非均匀的MoE布局，平衡效率与性能。

1.3 线性注意力

最初，线性注意力机制[27]被提出是为了解决softmax自注意力机制的二次复杂度问题。它在Q和K中使用额外的核函数ϕ，将计算复杂度降低至$\mathcal{O}(N)$。后续工作[28-33]在理论和模型结构方面进一步推进了线性注意力机制的发展。尽管这些方法有效，但它们主要是通过近似softmax的方式来近似线性复杂度，因此，在表达能力上仍逊色于标准softmax自注意力机制。本文使用的Agent注意力机制[19]整合了两种注意力机制的优势，既保留了softmax注意力的计算，又采用代理agent减少冗余，更利于提升在流量数据场景下的适配度与性能。

1.4 MoE机制

MoE是一类允许大幅增加模型参数量，而不会对模型训练和推理效率产生太大影响的技术，由Jacobs等人[34]和Jordan等人[35]提出。MoE模型采用稀疏激活的专家路由机制，显著提升模型表达能力与计算效率。早期的MoE方法，如Shazeer等人[36]提出的Sparsely-Gated MoE在机器翻译中展示出令人瞩目的扩展性，其核心思想是将输入动态路由到一小部分专家网络，避免全量参数参与计算，从而实现“大模型、小计算”。目前，MoE架构已成为提升大语言模型扩展性的关键技术，被广泛应用于生成式预训练变换器第4代（Generative Pre-trained Transformer 4，GPT-4）、Mixtral及DeepSeek等先进模型中，在保持推理成本可控的同时实现了千亿级参数的模型容量[37]。近年来的研究进一步探索MoE的鲁棒性与任务适应性。例如，视觉混合专家模型（Vision Mixture of Experts，V-MoE）[38]引入可微分门控函数以提升专家选择的泛化能力；任务级混合专家模型（Task-level Mixture of Experts，Task-MoE）[39]强调多任务场景下的专家专精机制；M6混合专家模型（Multi-Modality to Multi-Modality Multitask Mega-transformer with MoE，M6-MoE）[40]探索在中文语义建模中的大规模应用。此外，如Router Attention[41]等方法尝试将MoE融入Transformer的前馈网络层（Feed-Forward Network，FFN）层或注意力模块，实现更细粒度的专家适配。基于此，本文针对流量数据的多粒度异构性，设计相应的MoE层，并引入部分Mamba模块的FFN层中，以期提供更强的性能。

2 预备知识

2.1 基于Transformer的流量分类

Mamba是一个结构化的状态空间序列模型。如图2（a）所示，Mamba通过中间潜在状态$h(t)$将输入$x(t)$映射到输出$y(t)$：

$$h'(t)=\boldsymbol{A}h(t)+\boldsymbol{B}x(t) \tag{1}$$

$$y(t)=\boldsymbol{C}h(t)+\boldsymbol{D}x(t) \tag{2}$$

其中，参数$\boldsymbol{A}\in\mathbb{R}^{N\times N}$存储了所有历史信息，并控制先前隐藏状态对下一个隐藏状态的影响；$\boldsymbol{B}\in\mathbb{R}^{N\times 1}$控制输入$x(t)$对隐藏状态的影响程度；隐藏状态$h(t)$通过投影参数$\boldsymbol{C}\in\mathbb{R}^{N\times 1}$转换为输出；通常，$\boldsymbol{D}$提供了一个跳跃连接，因此可以假设$\boldsymbol{D}=\boldsymbol{0}$来简化。

为了获得更好的计算效率，上式中的连续参数$\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$进一步转化为离散参数。具体而言，假设步长为Δ，可应用零阶保持规则，按照以下公式获得离散参数：

$$\begin{cases} h_t = \overline{\boldsymbol{A}} h_{t-1} + \overline{\boldsymbol{B}} x_t \\ y_t = \boldsymbol{C} h_t \\ \overline{\boldsymbol{C}} = \boldsymbol{C} \end{cases} \tag{3}$$

其中,$\overline{\boldsymbol{A}} = e^{\Delta \boldsymbol{A}}$,$\overline{\boldsymbol{B}} = (\Delta \boldsymbol{A})^{-1}\left(e^{\Delta \boldsymbol{A}} - I\right) \cdot \Delta \boldsymbol{B}$。此外,对于大小为$N$的输入序列,可以应用核为$\boldsymbol{H}$的全局卷积来计算输出,计算式为

$$\begin{cases} \overline{\boldsymbol{H}} = \left(\boldsymbol{C}\overline{\boldsymbol{B}}, \boldsymbol{C}\overline{\boldsymbol{A}\boldsymbol{B}}, \cdots, \boldsymbol{C}\overline{\boldsymbol{A}}^{N-1}\overline{\boldsymbol{B}}\right) \\ \boldsymbol{y} = \boldsymbol{x} * \overline{\boldsymbol{H}} \end{cases} \tag{4}$$

其中,$\overline{\boldsymbol{H}} \in \mathbb{R}^N$表示结构化卷积核,$*$表示卷积操作。

2.2　Agent注意力

基于自注意力机制的Transformer结构因其在捕捉短程时间序列依赖关系方面出色的能力,在自然语言处理和计算机视觉领域有着重要的应用。Transformer的核心机制(softmax自注意力)如图2(b)所示,其表达式为

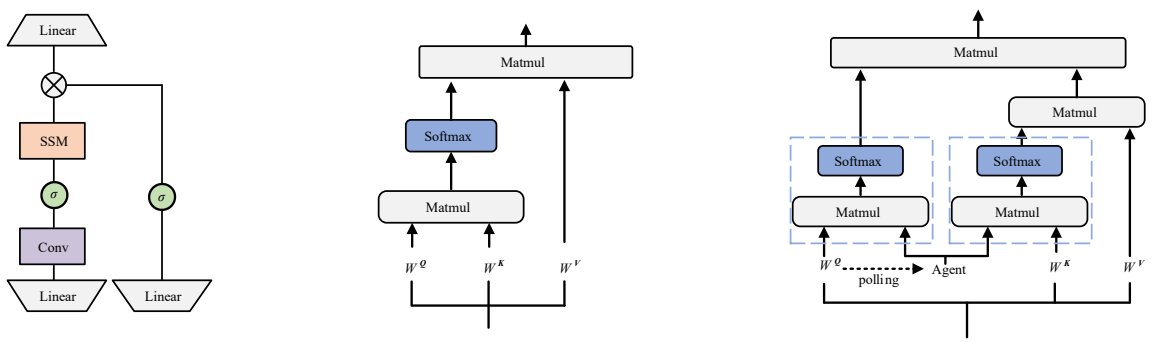
$$O^S = \sigma(\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}})\boldsymbol{V} \triangleq \mathrm{Attn}^S(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) \tag{5}$$

其中,$\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V} \in \mathbb{R}^{N \times C}$分别表示查询、键和值矩阵,$\sigma(\cdot)$表示softmax函数。

然而,softmax注意力机制强制计算所有查询-键值对之间的相似度,导致计算复杂度呈二次方增长,在计算长序列时会带来巨大的计算挑战。为降低计算开销,已有研究尝试通过设计稀疏全局注意力机制[42]或窗口注意力机制[43]来减少计算量。一般地,线性注意力的计算可表示为

$$O^{\phi} = \phi(\boldsymbol{Q})\phi(\boldsymbol{K})^{\mathrm{T}}\boldsymbol{V} \triangleq \mathrm{Attn}^{\phi}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) \tag{6}$$

其中,$\phi(\cdot)$为映射函数。但这类方法通常不可避免地损害了模型的全局建模能力。



(a) Mamba模块　(b) Softmax注意力　(c) Agent注意力

(a) Mamba module　(b) Softmax attention　(c) Agent attention

图2　基础模块结构图

Figure 2　Basic module structure diagram

最近,Han等人[20]提出了一种新的注意力范式来取代Transformer中的自注意力结构,即Agent注意力,其结构如图2(c)所示。由于全局自注意力机制本身具有冗余性,大量的查询token所需要的信息实际上是相同的。因此,Agent注意力采用代理token代替需要类似特征的查询token,从特征中捕获信息。这些远小于查询token数量的代理token大大降低了计算复杂度。Agent注意力方法引入了一个额外的代理矩阵$\boldsymbol{A}$,用于桥接查询和键之间的信息传递关系。最终的注意力输出可表示为

$$O^A = \underbrace{\mathrm{Attn}^S\left(\boldsymbol{Q}, \boldsymbol{A}, \underbrace{\mathrm{Attn}^S(\boldsymbol{A}, \boldsymbol{K}, \boldsymbol{V})}_{\text{Agent Aggregation}}\right)}_{\text{Agent Broadcast}} \tag{7}$$

与传统注意力机制不同,注意力参数$\boldsymbol{Q}$和$\boldsymbol{K}$不直接用于计算注意力得分,而是通过合并来自$\boldsymbol{Q}$和$\boldsymbol{K}$的信息生成代理矩阵$\boldsymbol{A} \in \mathbb{R}^{n \times N}$,对其进行解耦与重组,以促进它们之间的信息传递。具体来说,第一个softmax注意力计算将代理$\boldsymbol{A}$作为$\boldsymbol{K}$,将代理特征的全局信息广播到每个查询。第二个softmax注意力计算将代理$\boldsymbol{A}$视为查询,从$\boldsymbol{V}$中聚合代理特征。因此,Agent注意力不仅具有softmax注意力的优秀建模能力,还具有类似线性注意力的线性复杂度。

$$\begin{aligned} O^A &= \sigma\left(\boldsymbol{Q}\boldsymbol{A}^{\mathrm{T}}\right)\sigma\left(\boldsymbol{A}\boldsymbol{K}^{\mathrm{T}}\right)\boldsymbol{V} \\ &= \phi_q(\boldsymbol{Q})\phi_k(\boldsymbol{K})^{\mathrm{T}}\boldsymbol{V} \\ &= \underbrace{\mathrm{Attn}^{\phi_{q/k}}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})}_{\text{Generalized Linear Attn}} \# \end{aligned} \tag{8}$$

在流量数据中,并非所有字节或字段均等价重要。在加密流量数据中,头部各字节信息均具有不同的含义,某些字节信息可能更具判别性,而加密负载信息更强调整体的模式,冗余性较高,单个字节提供的判别性信息较少[44]。因此,Agent注意力采用代理token的方式对于加密流量分类任务十分友好,能够在小于softmax注意力计算复杂度的情况下有效减少冗余性,提供相近甚至更高的性能表现。

2.3 MoE层

假设N_e位专家$\{E_i\}_{i=1}^{N_e}$，每位专家均为一个可训练的前馈网络，且参数数量相同。对于每个token嵌入值x，计算得分$h(x)=Wx\in\mathbb{R}^{N_e}$，其中$W$为可训练的线性投影。进一步使用softmax进行归一化：

$$p_i(x)=\frac{\exp\left(h(x)_i\right)}{\sum_{i=1}^{N_e}\exp\left(h(x)_i\right)} \tag{9}$$

在训练中，top-k路由选择每个token中$k\geqslant 1$个最合适的专家。当使用$k=1$进行简化时，x的MoE层输出为

$$y=p_I E_I(x) \tag{10}$$

其中，$I=\arg\max_i\ p_i(x)$。路由选择由可学习的参数组成，并与网络的其他部分一起训练。

3 方法介绍

3.1 流量数据预处理

原始流量数据通常由两台主机的两个端口之间传输的数据包组成。每个数据包包含固定结构的报头和可变长度的有效负载，具有异构性与长度不一致等特点。为了确保模型能够获得一致的输入形式，对原始流量数据进行标准化预处理非常必要。

首先，将原始网络流量划分为不同的会话流，每个会话流对应于通信双方在一定时间内的端到端数据交互，且数据包遵循相同的传输协议。然后，为了保留有价值的信息并消除不必要的干扰，对数据包中的源IP地址、目标IP地址以及端口进行匿名化处理。为保证输入一致性，需要通过填充和裁剪将所有数据包标准化为统一长度320字节，其中报头和有效载荷设置为固定字节长度，分别为80字节和240字节。具体而言，将报头保留80字节足以完整覆盖头部及其扩展选项字段，避免关键通信控制信息的丢失；240字节的有效载荷既能捕获应用层初期的核心结构化语义，又能有效防止填充过多的零数据带来的干扰。

为了使模型能够高效地处理流量数据包，使用块嵌入技术对数据包进行嵌入化处理。首先，将经过划分后的会话流pcap文件的前5个包分为一组，包含少于5个包的组会通过填充零的方式进行补足。经过处理后获得长度为1 600字节的会话流，并转换为$L=1\,600$长度的整数数组。这些数组进行标准化处理，并以npy格式存储。当预处理后的网络流量数据进行嵌入时，整数数组依次划分为长度为P的块，总共得到$N_x=L/P$个块，每个块被线性映射到一个维度为D的向量，并添加可学习的位置编码。其中$P=4$，$D=256$。为了能够捕捉整个数据流的全局信息，借鉴基于深度双向Transformer的预训练语言理解模型（Bidirectional Encoder Representations from Transformers，BERT）[45]中的设计，引入一个可训练的类标记。最终的流嵌入表示为

$$\boldsymbol{X}=\left[\boldsymbol{X}_{N_x};\ x_{\text{cls}}\right]+\boldsymbol{E}_{\text{pos}} \tag{11}$$

其中，$\boldsymbol{X}\in\mathbb{R}^{N\times D}$表示最终的流嵌入，$N=N_x+1$，$x_{\text{cls}}\in\mathbb{R}^{1\times D}$表示类标记，$\boldsymbol{E}_{\text{pos}}\in\mathbb{R}^{N\times D}$表示位置嵌入。考虑到Mamba从前到后逐步建模的特性以及流量数据中包含的顺序关系，本文将类标记拼接在流嵌入的末尾以增强信息聚合。

3.2 提出的ET-LDP预训练模型

与现有的纯Transformer架构相比，ET-LDP模型融合Mamba局部动态感知、Agent全局注意力和MoE专家路由的跨机制协同架构。这种设计更加契合加密流量的结构化报头与高熵载荷并存的多粒度特性，且减轻了平方级计算复杂度带来的资源压力。如图3所示，模型整体架构包括三个主要组成部分：流量预处理模块、特征编码器和轻量级解码器。以下部分重点介绍网络流量的通用学习表示。编码器由多个模块堆叠构成，每个模块内部集成了MoE机制的Mamba层、自注意力层和Agent注意力层，并且在Mamba层与注意力层之间设计了过渡层以增强它们的协同建模。编码器在预训练和微调阶段的作用相同，均用于提取网络流量特征。解码器由两个Mamba层构成，仅在预训练阶段启用。3.3节提出了ET-LDP编码器的整体结构，并详细描述了其核心组件的构成。3.3.1节提出了作用于Mamba的MoE；3.3.2节提出了混合注意力机制，以增强编码器在复杂流量场景下的表达能力；3.3.3节提出了Mamba层和注意力层之间适配两种不同信息流向的跨架构桥接模块，包括Mamba层转向注意力层的M2T-Bridge模块和注意力层转向Mamba层的T2M-Bridge模块。

3.3 ET-LDP编码器

ET-LDP编码器结构共划分为四个阶段，每个阶段采用不同的混合结构，并按照一定的顺序排列。鉴于流量数据与图像或自然语言数据类型的不同，其本质上属于结构化时序数据或者结构化token，并没有很强的空间拓扑结构。因此，在设计网络结构时虽然采用了分阶段特征提取，但避免了传统视觉模型中常见的分辨率下采样操作。每阶段保持特征尺寸不变，通过逐层递增的块数（depths=[1,2,4,8]）进行建模。一方面，保留完整的token序列有助于最大程度地保存原始流量中的时序与结构特征，避免信息压缩带来的判别性损失；另一方面，逐层递增的block数构建了从浅到深的语义建模路径，通过深度实现表达能力提

升,使得模型能够在统一特征空间中实现层次化、多粒度的流量理解。具体而言,第 i 个阶段包含 $L_i/2$ 个 Mamba 层,$L_i/4$ 个 agent 注意力层和 $L_i/4$ 个自注意力层,其中 $i\in\{1,2,3,4\}$,$L\in\{1,2,4,8\}$。特殊地,当 $i=1$ 时,仅使用 Mamba 层进行构建。当 $i=2$ 时,在 Mamba 层后,选择使用 softmax 自注意力层或 agent 注意力层进行计算。为了平衡计算效率和性能,本结构设计在后三个阶段中,保持 Mamba 层与注意力层的比例为 1∶1。

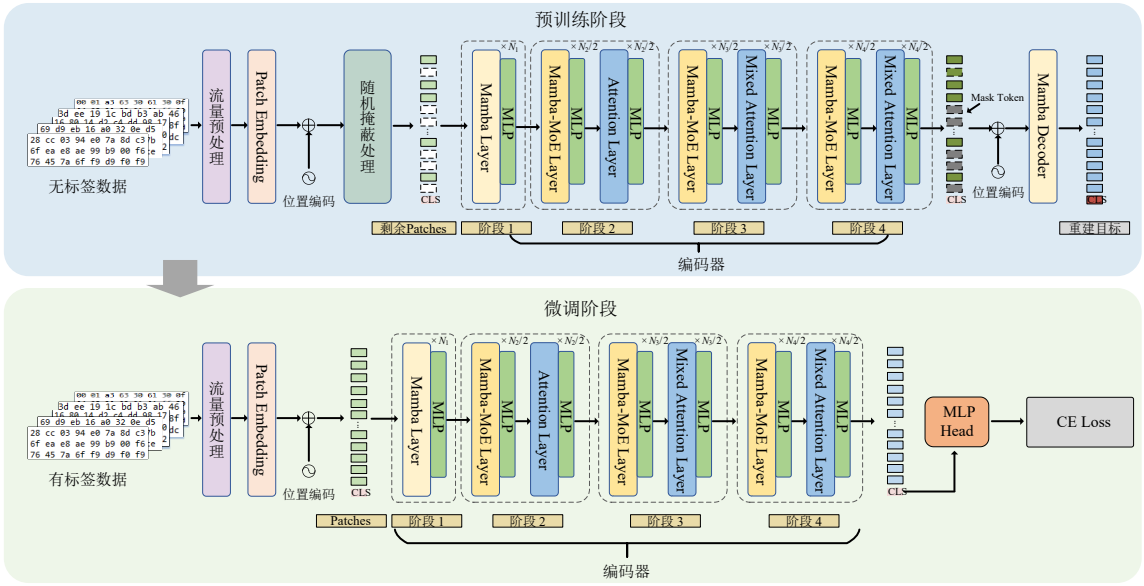


图 3　ET-LDP 模型架构图

Figure 3　ET-LDP architecture diagram

由于 Mamba 中的位置感知能力主要由因果卷积隐式的局部位置信息提供,在缺乏外部位置信息引导时,能够关注的范围有限。Transformer 的设计本身并未优先考虑时间依赖性,难以保持局部序列一致性,因此,作为第一个编码层可能会损失流级别的序列顺序信息。尽管 Agent 注意力实现了近线性的复杂度且有利于减少冗余信息,但不可避免会损失一些信息。因此,在第一阶段的编码器采用单层 Mamba 并结合位置编码来实现,确保在开始时提取足够的局部特征与更广泛的流模式,可以表示为

$$\boldsymbol{F}_{\text{pre}}=\text{MLP}\left(\text{Mamba\_layer}\left(\boldsymbol{X}\right)\right) \tag{12}$$

其中,$\boldsymbol{F}_{\text{pre}}$ 表示由第一阶段模块获得的粗略特征。考虑到 Mamba 的因果特性,其递归机制能够在 SSM 的计算过程中有效捕捉邻近 token 之间的相对位置关系。这使得 Mamba 能够自然地解释网络流量序列中的顺序。此外,与时间序列和自然语言序列不同,网络流量序列的上下文顺序是弱相关的。因此,在编码器的后续三个阶段中,仍需加强位置关系建模。基于此,编码器的后三个阶段首先均使用 Mamba 作为起始模块,对带有位置编码的流特征 $\boldsymbol{F}_{\text{pre}}$ 进行处理,使其获取流量数据的有效位置关系,然后送入后续的编码器阶段提取完整特征 $\boldsymbol{F}_{\text{enc\_out}}$。这种方法确保了位置信息和长期依赖关系都能被有效捕捉。第 i 个阶段 $(i>1)$ 中的计算过程可以表示为

$$\begin{cases}\boldsymbol{F}_{\text{mam}_i}=\text{Mamba}_i^{\text{sub}}\left(\boldsymbol{F}_{\text{pre}}\right)\\ \boldsymbol{F}_{\text{mlp}_i}=\text{MLP}\left(\boldsymbol{F}_{\text{mam}_i}\right)\\ \boldsymbol{F}_{\text{attn}_i}=\text{Attention}_i^{\text{sub}}\left(\boldsymbol{F}_{\text{mlp}_i}\right)\\ \boldsymbol{F}_{\text{m2t}_i}=\text{M2T}\left(\boldsymbol{F}_{\text{attn}_i},\boldsymbol{F}_{\text{mlp}_i}\right)\\ \boldsymbol{F}_{\text{enc\_out}_i}=\text{MLP}\left(\boldsymbol{F}_{\text{m2t}_i}\right)\end{cases} \tag{13}$$

其中,$\text{Mamba}_i^{\text{sub}}$ 和 $\text{Attention}_i^{\text{sub}}$ 分别表示第 i 个阶段中的 Mamba 子块(由 $N_i/2$ 个 mamba 层组成)和 attention 子块(由 $N_i/4$ 个 agent 注意力层和 $N_i/4$ 个 softmax 注意力层组成);M2T 为 Mamba 特征向 Attention 特征转化的 M2T-Bridge 模块,以增强两种异质性特征的融合;MLP 层用于适配 Mamba 层和 Attention 层,以自适应减少直接连接造成的结构化差异。

第 $i-1$ 个阶段 $(i>2)$ 与第 i 个阶段之间的计算过程可表示为

$$\begin{cases}\boldsymbol{F}_{\text{attn}_{i-1}}=\text{Attention}_{i-1}^{\text{sub}}\left(\boldsymbol{F}_{\text{mlp}_{i-1}}\right)\\ \boldsymbol{F}_{\text{mlp}_{i-1}}=\text{MLP}\left(\boldsymbol{F}_{\text{attn}_{i-1}}\right)\\ \boldsymbol{F}_{\text{mam}_i}=\text{Mamba}_i^{\text{sub}}\left(\boldsymbol{F}_{\text{mlp}_{i-1}}\right)\\ \boldsymbol{F}_{\text{t2m}_i}=\text{T2M}\left(\boldsymbol{F}_{\text{mam}_i},\boldsymbol{F}_{\text{mlp}_{i-1}}\right)\\ \boldsymbol{F}_{\text{enc\_out}_i}=\text{MLP}\left(\boldsymbol{F}_{\text{t2m}_i}\right)\end{cases} \tag{14}$$

其中，T2M 表示 Attention 特征向 Mamba 特征转化的 T2M-Bridge 模块，其作用与 M2T-Bridge 模块类似，用于另一种特征流向的转化与融合。

3.3.1 基于MoE的Mamba层

在 ET-LDP 模型中，Mamba 块是实现处理网络流量多层次语义的顺序序列分类不可或缺的关键组件。相较于 Transformer，Mamba 层更擅长流量数据的顺序建模与动态状态捕获。因此，将 MoE 机制引入 Mamba 层中，构建了 Mamba-MoE 层，可有效提高模型容量以应对复杂流量环境，尤其在处理网络流量这类多层次语义的序列数据时表现更为优越。对于标准 Mamba 块，其信息流处理情况如下：

$$\boldsymbol{X}^{l}=\mathrm{SSM}\left(\boldsymbol{X}^{l-1}\right)+\boldsymbol{X}^{l-1} \tag{15}$$

$$\boldsymbol{x}_n^{l}=\mathrm{FFN}\left(\boldsymbol{x}_n^{l}\right)+\boldsymbol{x}_n^{l} \tag{16}$$

其中，SSM 为状态空间模型，FFN 为前馈网络，$\boldsymbol{X}^{l}$ 为第 l 个 SSM 块之后所有 token 的隐藏状态，$\boldsymbol{x}_n^{l}$ 为第 l 个前馈网络之后第 n 个 token 的隐藏状态。

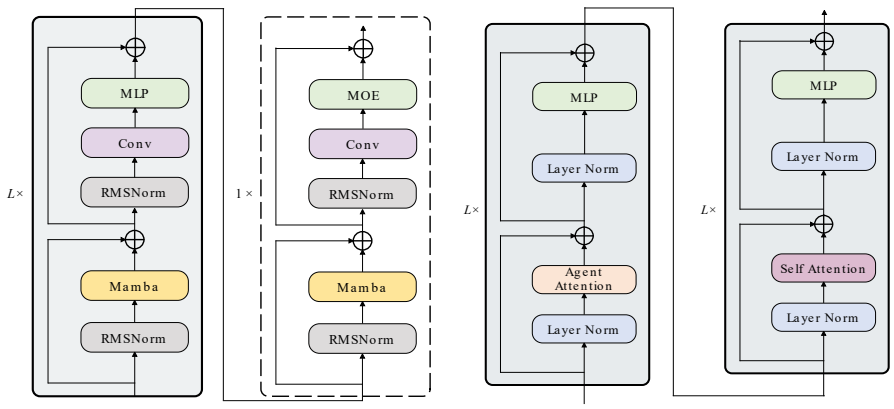
Mamba-MoE 使用 MoE 替换了原本的前馈网络，将每个 token 分配给前 m 位专家。将第 l 个 FFN 块替换为 MoE 块，其输出隐藏状态可定义如下：

$$\boldsymbol{x}_n^{l}=\sum_{m=1}^{N_e}\left(g_{m,n}\cdot\mathrm{FFN}_m\left(\boldsymbol{x}_n^{l}\right)\right)+\boldsymbol{x}_n^{l} \tag{17}$$

$$g_{m,n}=\begin{cases}p_{m,n}, & p_{m,n}\in\mathrm{top}K\left(p_{j,n}\mid 1\leqslant j\leqslant N_e\right\},K)\\ 0, & \text{otherwise}\end{cases} \tag{18}$$

$$p_{m,n}=\mathrm{Softmax}_m\left(\boldsymbol{X}_n^{l^N}e_m^{l}\right) \tag{19}$$

其中，N_e 为专家的数量，N 为 token 的总数，FFN_m 为第 m 个专家的前馈网络，$g_{m,n}$ 为针对第 n 个 token 分配给第 k 个专家的门控值，$p_{m,n}$ 为 token 与专家之间的亲和力，$\mathrm{top}K(\cdot,K)$ 为第 n 个 token 与所有专家之间亲和力的 top$K$ 选择函数，e_m^{l} 为第 l 层中第 k 个专家的质心。考虑到计算效率与性能的平衡，仅在每个阶段的最后一个 Mamba 层使用 MoE 机制，如图 4(a)所示。



(a) Mamba-MoE模块 (b) Mixed Attention模块

(a) Mamba-MoE module (b) Mixed Attention module

图4 用于ET-LDP编码器的混合模块结构图

Figure 4 Diagram of the hybrid module structure for ET-LDP encoder

考虑到网络流量不是单一尺度的数据，其同时包含流级别的全局模式和字节级的局部语义。流级在特征层面包含长时序、统计模式、来源/行为特征，体现一个通信会话的整体语义，如协议、应用类型、攻击行为等。字节级特征包含精细语义、内容、结构字段，经过预处理后分块 token 可能表示 header 字段、payload 语义、TLS 证书片段等，属于精细语义、内容、结构字段等特征。如果专家结构对不同粒度信息不进行区分，很有可能导致多个路由专家在获取同一个流量信息时出现专家参数冗余。因此，Mamba-MoE 的共享-私有门控专家(Shared-Private Gated Expert，SPGE)结构采用共享参数和私有参数相结合的方法，由共享 FFN 分支、私有门控和私有 FFN 三部分组成。共享参数总是被激活，旨在捕获和整合流量整体的知识，同时减轻其他路由专家之间的参数冗余，提高效率。私有参数保证了更加准确和针对性的网络流量 token 级别的知识获取。路由到各个专家的令牌首先通过共享 FFN 存储公共知识的参数，然后通过私有知识的门控网络让私有专家决定他们应该从共享的隐藏状态中接收哪些信息。如图 5 所示，SPGE 的计算过程表示为

$$\boldsymbol{s}=\mathrm{ReLU}\left(\boldsymbol{X}^{l}\boldsymbol{W}_g^{s}\right)\boldsymbol{W}_d^{s} \tag{20}$$

$$\tilde{\boldsymbol{s}}=\sigma\left(\boldsymbol{s}\,\boldsymbol{W}_{gt}^{p,e}\right)\odot\boldsymbol{s} \tag{21}$$

$$\boldsymbol{y}=\mathrm{ReLU}\left(\tilde{\boldsymbol{s}}\,\boldsymbol{W}_g^{p,e}\right)\boldsymbol{W}_d^{p,e} \tag{22}$$

其中：$\boldsymbol{W}_g^{s}\in\mathbb{R}^{D\times D_s}$ 和 $\boldsymbol{W}_d^{s}\in\mathbb{R}^{D_s\times D}$ 分别表示共享分支

中的共享门控投影和下投影,它们承载流量的公共知识,始终激活;$\boldsymbol{W}_{gt}^{p,e} \in \mathbb{R}^{D\times 1}$ 为第 e 个专家的私有门控,与 Sigmoid 共同决定私有分支接收哪些信息;$\boldsymbol{W}_{g}^{p,e} \in \mathbb{R}^{D\times D_p}$ 和 $\boldsymbol{W}_{d}^{p,e} \in \mathbb{R}^{D_p\times D}$ 表示第 e 个专家的私有 FFN,主要作用于流量 token 级细粒度表示学习。

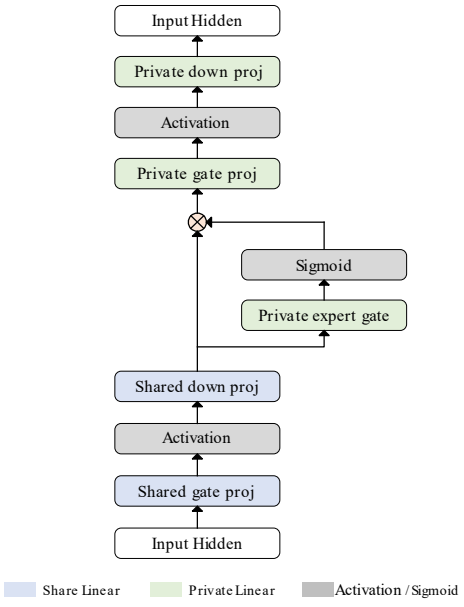


图 5　SPGE 结构图

Figure 5　Diagram of Shared-Private Gated Experts structure

3.3.2　混合注意力机制设计

Transformer 的性能虽然已在自然语言、计算机视觉等领域得到验证,但其平方级的计算复杂度带来了高昂的计算压力。为了在性能和计算需求之间找到合适的平衡点,ET-LDP 编码器在注意力层中引入了近似线性复杂度的代理注意力机制,以取代部分传统 Transformer 中的 softmax 自注意力机制,从而在 Attention 层中设计了一种融合传统注意力与线性注意力的混合注意力机制。该机制优先考虑“Agent 注意力先行,softmax 注意力后接”的策略。如图 4(b)所示,Agent 注意力通过引入代理机制,使局部 token 信息先汇聚到少数代表 token 上,以减少 token 间的冗余联系。这对于流量数据中结构重复、不均衡[46]的情况尤其有帮助。在完成代理归纳后,引入标准的 softmax 自注意力进一步建模 token 间的全局上下文依赖,使得细粒度的 token 之间可以直接建立联系,提升建模的完整性,从而防止 Mamba 和 Agent 注意力损失细节。为验证设计合理性,本文尝试了一系列注意力的混合策略,并比较了它们从头开始训练时的性能,结果证明了方法的合理性,具体细节见 4.4.1 节。

3.3.3　跨架构桥接模块

Mamba 层侧重于流量数据的顺序建模与动态状态捕获,而自注意力层更加擅长显式建模依赖。为了在两者之间建立有效的信息衔接机制,引入了 M2T-Bridge 模块,使得来自 Mamba 的状态特征能够在进入注意力层之前得到结构性适配与语义增强。

根据文献[47]对 Mamba 与注意力机制的分析,可以将式(4)表示为注意力矩阵的形式:

$$\boldsymbol{y}=\boldsymbol{\alpha}\boldsymbol{x} \tag{23}$$

$$\boldsymbol{\alpha}=\begin{bmatrix} \boldsymbol{C}_1\overline{\boldsymbol{B}_1} & 0 & \cdots & 0 \\ \boldsymbol{C}_2\overline{\boldsymbol{A}_2}\,\overline{\boldsymbol{B}_1} & \boldsymbol{C}_2\overline{\boldsymbol{B}_2} & \cdots & 0 \\ \vdots & \vdots & \vdots & 0 \\ \boldsymbol{C}_N\prod_{k=2}^{N}\overline{\boldsymbol{A}_k}\,\overline{\boldsymbol{B}_1} & \boldsymbol{C}_N\prod_{k=3}^{N}\overline{\boldsymbol{A}_k}\,\overline{\boldsymbol{B}_2} & \cdots & \boldsymbol{C}_N\overline{\boldsymbol{B}_N} \end{bmatrix} \tag{24}$$

其中,N 为序列长度,$\boldsymbol{\alpha}$ 为 Mamba 中的隐式注意力矩阵。将 $\boldsymbol{\alpha}$ 按照行号 i 与列号 j 展开可以得到:

$$\alpha_{i,j}=\boldsymbol{C}_i\prod_{k=j+1}^{i}\overline{\boldsymbol{A}_k}\,\overline{\boldsymbol{B}_j} \tag{25}$$

根据 Mamba 的选择性,将参数 $\boldsymbol{B}$、$\boldsymbol{C}$ 和 $\boldsymbol{\Delta}$ 表示为跟输入 x_i 相关的形式:

$$\boldsymbol{B}_i=S_{\boldsymbol{B}}(x_i) \tag{26}$$

$$\boldsymbol{C}_i=S_{\boldsymbol{C}}(x_i) \tag{27}$$

$$\Delta_i=\mathrm{softplus}\left(S_{\Delta}(x_i)\right) \tag{28}$$

其中,$S_{\boldsymbol{B}}$、$S_{\boldsymbol{C}}$ 和 S_{Δ} 为线性投影,i 为序列中元素的索引。根据矩阵 $\boldsymbol{A}$ 的对角性质,式(25)可以视为逐元素乘法的求和:

$$\alpha_{i,j}=S_{\boldsymbol{C}}(x_i)\left(\exp\left(\sum_{k=j+1}^{i}\mathrm{softplus}\left(S_{\Delta}(x_k)\right)\boldsymbol{A}\right)\right)\cdot\mathrm{softplus}\left(S_{\Delta}(x_j)\right)S_{\boldsymbol{B}}(x_j) \tag{29}$$

对上式进行定义:

$$Q_i=S_{\boldsymbol{C}}(x_i) \tag{30}$$

$$K_j=\mathrm{softplus}\left(S_{\Delta}(x_j)\right)S_{\boldsymbol{B}}(x_j) \tag{31}$$

$$H_{i,j}=\exp\left(\sum_{k=j+1}^{i}\mathrm{softplus}\left(S_{\Delta}(x_k)\right)\boldsymbol{A}\right) \tag{32}$$

$$V_i=x_i \tag{33}$$

此时 Mamba 的输出可以表示为

$$y_i=\sum_{j=1}^{i}\alpha_{i,j}x_j=\sum_{j=1}^{i}Q_iK_jH_{i,j}V_j \tag{34}$$

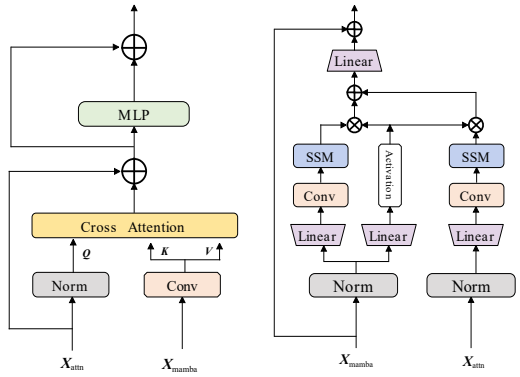
由此得出,Mamba 是一种特殊的注意力,两者在数学本质上是同源的,区别在于其 Mask($H_{i,j}$)不是静态的,而是由数据内容动态控制的。因此,Mamba 的特征和 Transformer 的特征实际上都是在计算某种形式的 $\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$ 交互。它们处于同一数学框架下,这为特征对齐提供了理论可行性,从而消除了异构的壁垒。

如图 6(a)所示,M2T-Bridge 将 Mamba 层输出

$\boldsymbol{X}_{\text{mamba}} \in \mathbb{R}^{N\times D}$ 作为键与值，而当前注意力层输入 $\boldsymbol{X}_{\text{attn}} \in \mathbb{R}^{N\times D}$ 经过归一化后作为查询，通过多头交叉注意力实现来自两种架构的双向信息对齐：

$$\boldsymbol{X}=\boldsymbol{X}+\text{CrossAttn}\left(\text{Norm}(\boldsymbol{X}_{\text{attn}}),\ \text{Conv1}(\boldsymbol{X}_{\text{mamba}}),\text{Conv1}(\boldsymbol{X}_{\text{mamba}})\right) \tag{35}$$

与直接残差注入信息不同，M2T-Bridge 通过显式的跨结构注意力机制，而非被动叠加。这种设计使得 Mamba 的局部顺序建模能力能够直接参与注意力权重分配，而注意力层也能在进入全局依赖建模之前接收更符合流量结构特点的表示，实现两类架构之间的功能互补。



(a) M2T-Bridge 模块 (b) T2M-Bridge 模块

(a) M2T-Bridge module (b) T2M-Bridge module

图 6 M2T-Bridge 和 T2M-Bridge 模块结构图

Figure 6 Architecture of the M2T-Bridge and T2M-Bridge

为了使注意力输出能够被有效转化到 Mamba 的状态建模空间，引入 T2M-Bridge 模块，用于在注意力层与 Mamba 层之间建立过渡结构，实现跨结构特征的融合。T2M-Bridge 模块设计思路与 M2T-Bridge 形成对偶关系，前者采用的是交叉注意力来实现特征融合，后者则是基于 Mamba 的计算方式实现两种特征的交叉融合。在 T2M-Bridge 中，当前时刻的 Mamba 隐藏状态 $\boldsymbol{X}_{\text{mamba}} \in \mathbb{R}^{N\times D}$ 作为主分支输入，来自注意力层的输出 $\boldsymbol{X}_{\text{attn}} \in \mathbb{R}^{N\times D}$ 经过归一化和卷积计算后在 SSM 计算中更新。最后两路特征采用门控机制实现跨模块特征的交互，鼓励互补特征学习，从而将注意力结构编码的全局依赖重新注入状态空间计算路径中。该模块的细节在算法 1 中进行了说明。

算法 1 T2M-Bridge Block (Attention → Mamba)

输入： attention feature $\boldsymbol{X}_{\text{attn}}:(N, D)$

mamba hidden state $\boldsymbol{X}_{\text{mamba}}:(N,D)$

输出： updated mamba feature $\boldsymbol{X}_{\text{out}}:(N,D)$

--- Branch 1: process attention feature ---

$\boldsymbol{X}'^{\text{attn}}_{l-1}:(B,N,D) \leftarrow \text{Norm}\left(\boldsymbol{X}^{\text{attn}}_{l-1}\right)$

$\boldsymbol{x}_{\text{attn}}:(B,N,P) \leftarrow \text{Linear}_{\text{attn}}\left(\boldsymbol{X}'^{\text{attn}}_{l-1}\right)$

$\boldsymbol{x}'_{\text{attn}}:(B,N,P) \leftarrow \text{SiLU}\left(\text{Conv1d}_{\text{attn}}\left(\boldsymbol{x}_{\text{attn}}\right)\right)$

$\boldsymbol{A}_{\text{attn}},\boldsymbol{B}_{\text{attn}},\boldsymbol{C}_{\text{attn}}:(B,N,P,K) \leftarrow \text{ParametersFunction}_{\text{attn}}\left(\boldsymbol{x}'_{\text{attn}}\right)$

$\boldsymbol{y}_{\text{attn}}:(B,N,P) \leftarrow \text{SSM}\left(\boldsymbol{A}_{\text{attn}},\boldsymbol{B}_{\text{attn}},\boldsymbol{C}_{\text{attn}}\right)\left(\boldsymbol{x}'_{\text{attn}}\right)$

--- Branch 2: process mamba hidden state ---

$\boldsymbol{X}'^{\text{mamba}}_{l-1}:(B,N,D) \leftarrow \text{Norm}\left(\boldsymbol{X}^{\text{mamba}}_{l-1}\right)$

$\boldsymbol{x}_{\text{mamba}}:(B,N,P) \leftarrow \text{Linear}_{\text{mamba}}\left(\boldsymbol{X}'^{\text{mamba}}_{l-1}\right)$

$\boldsymbol{x}'_{\text{mamba}}:(B,N,P) \leftarrow \text{SiLU}\left(\text{Conv1d}_{\text{mamba}}\left(\boldsymbol{x}_{\text{mamba}}\right)\right)$

$\boldsymbol{A}_{\text{mamba}},\boldsymbol{B}_{\text{mamba}},\boldsymbol{C}_{\text{mamba}}:$

$(B,N,P,K) \leftarrow \text{ParametersFunction}_{\text{mamba}}\left(\boldsymbol{x}'_{\text{mamba}}\right)$

$\boldsymbol{y}_{\text{mamba}}:(B,N,P) \leftarrow \text{SSM}\left(\boldsymbol{A}_{\text{mamba}},\boldsymbol{B}_{\text{mamba}},\boldsymbol{C}_{\text{mamba}}\right)\left(\boldsymbol{x}'_{\text{mamba}}\right)$

--- gating fusion between two branches ---

$z:(B,N,P) \leftarrow \text{Linear}\left(\boldsymbol{X}^{\text{mamba}}_{l-1}\right)$

$\boldsymbol{y}'_{\text{mamba}}:(B,N,P) \leftarrow \boldsymbol{y}_{\text{mamba}} \odot \text{SiLU}(z)$

$\boldsymbol{y}'_{\text{attn}}:(B,N,P) \leftarrow \boldsymbol{y}_{\text{attn}} \odot \text{SiLU}(z)$

--- residual connection & projection ---

$\boldsymbol{F}^{\text{mamba}}_{l-1}:(B,C,N) \leftarrow \text{Reshape}\left(\text{Linear}^{\text{T}}\left(\boldsymbol{y}'_{\text{mamba}}+\boldsymbol{y}'_{\text{attn}}\right)+\boldsymbol{X}^{\text{mamba}}_{l-1}\right)$

$\boldsymbol{X}^{\text{mamba}}_{l}:(B,N,D) \leftarrow \text{Flatten}\left(\text{DWConv}\left(\boldsymbol{F}^{\text{mamba}}_{l-1}\right)+\boldsymbol{F}^{\text{mamba}}_{l-1}\right)$

Return: $\boldsymbol{X}^{\text{mamba}}_{l-1}$

3.4 预训练及微调策略

3.4.1 预训练

在预训练阶段采用 MAE 范式，并使用编码器-解码器架构。从整个流量嵌入中随机采样一部分 token 进行掩盖，仅将剩余的 token 输入模型，然后训练模型重建被掩蔽的块。具体来说，编码器从流量嵌入的剩余部分捕获特征及其之间的关系，并输出编码器 token；解码器则根据编码器标记重建被掩蔽的块。该过程由重建损失函数监督，即真实值 y_{gt} 与预测值 y_{rec} 之间的均方误差损失：

$$\mathcal{L}_{\text{rec}}=\text{MSE}\left(y_{\text{gt}},y_{\text{rec}}\right) \tag{36}$$

使用 MAE 预训练策略的目标是使模型学习到准确、统一的流量表示，以减少冗余信息，且预训练以无监督的方式进行，这使得可以使用大量无标签的流量数据进行预训练。

3.4.2 微调

在流量分类任务微调时，本文采用全量微调策略。具体而言，首先将加载预训练阶段获得的编码器参数作为骨干网络的初始化权重，并在其顶部添加一个随机初始化的任务特定 MLP 分类头。其次，在微调过程中，不冻结任何层，编码器与分类头的所有参数均通过反向传播算法进行端到端的联合更新。最后，将编码器获得的流量特征作为分类特征输入到 MLP 中，得到预测分布 $\hat{y} \in \mathbb{R}^{c}$，其中 c 表示流量数据的

类别数目。整个微调过程通过最小化真实标签y和预测分布$\hat{y}$之间的交叉熵损失来实现:

$$\mathcal{L}_{\mathrm{CE}}=\mathrm{CrossEntropy}\left(y,\hat{y}\right) \tag{37}$$

4　实验结果及分析

4.1　数据集及预处理

对于加密流量分类任务,在预训练阶段,以无监督的方式联合使用ISCXVPN 2016(虚拟专用网络与非虚拟专用网络流量分类数据集)[48]、ISCXTor 2016(Tor与非Tor网络流量分类数据集)[49]、CrossPlatform(Android)(跨平台(安卓端)流量数据集)[5]、CrossPlatform(iOS)(跨平台(苹果端)流量数据集)[5]、USTC-TFC数据集(中国科学技术大学网络流量分类数据集)[50]和CICIoT2022数据集(加拿大网络安全研究所物联网攻击数据集)[51]共六个数据集对模型进行预训练。在微调阶段,以监督的方式分别对以上数据集和CSTNET-TLS数据集[10]进行微调。每个数据集按8∶1∶1的比例分为训练集、验证集和测试集。这些数据集涵盖了各种网络应用场景和协议,例如使用超文本传输协议(HyperText Transfer Protocol,HTTP)的网页浏览、使用文件传输协议(File Transfer Protocol,FTP)的文件下载、使用简单邮件传输协议(Simple Mail Transfer Protocol,SMTP)的电子邮件和使用快速用户数据报协议网络连接(Quick User datagram protocol Internet Connections,QUIC)的视频流。

(1)ISCXVPN 2016是由加拿大网络安全部门加拿大网络安全研究所(Canadian Institute for Cybersecurity)在VPN和非VPN环境下捕获的通信应用程序构成。VPN通常用于绕过审查和通过协议混淆隐藏位置,由于其协议混淆,通过VPN进行通信的流量难以检测。该数据集包含来自VPN/非VPN两种类型的流量,分为7个通信类别的VPN流量数据,共包含16个应用程序的流量。

(2)ISCXTor 2016数据集包含利用Tor进行加密通信的应用程序流量,为通信增加了一层额外的混淆,包含来自8个通信类别的流量数据。这种流量通过分布式路由网络对发送方和接收方之间的通信进行混淆,进一步掩盖流量行为,流量模式的提取变得更加困难,使流量分类更具挑战性。

(3)USTC-TFC数据集包含由恶意软件和良性应用程序组成的加密流量,包括20种流量类型,其中包括10类良性流量和10类恶意流量,用于验证模型在网络安全管理场景下区分恶意与良性加密流量的能力。

(4)CrossPlatform(Android)和CrossPlatform(iOS)数据集包括来自美国、中国和印度的前100个Android和iOS应用程序生成的加密流量。CrossPlatform数据集的突发-静默模式明显[5],对于能够捕捉全局会话模式和统计特性建模更加友好。由于流量预处理后CrossPlatform(Android)和CrossPlatform(iOS)数据集存在少数类别样本数量非常稀少的情况,对模型的流量理解能力产生不利影响。因此,在不破坏数据集整体分布特点的前提下,决定丢弃数量少于50的类别,最终使用的类别数分别为181和124。

(5)CICIoT2022是一个从实验室网络收集的流量,旨在对各种物联网设备进行配置分析、行为分析和漏洞测试。选取了来自电力实验的良性流量以及包含拒绝服务(Denial of Service,DoS)攻击和暴力攻击的恶意流量分类数据集,包含6种主流的IoT恶意攻击类别。

(6)CSTNET-TLS数据集包含新加密协议TLS 1.3上的加密流量数据。TLS 1.3是目前互联网最前沿、应用最广且加密最彻底的传输协议标准,采集自真实的中国科技网(China Science and Technology NETwork,CSTNET)骨干网环境,包含120个应用程序。它高度贴近真实互联网服务提供商(Internet Service Provider,ISP)的现网流量场景,用于验证模型在面对现代强加密协议和真实骨干网高噪声环境下的分类极限界限。

4.2　实验设置及评价指标

在预训练阶段,总迭代次数为150 000次,批次大小为128。实验中应用线性学习率缩放规则,将基础学习率设置为1×10^{-3},并使用AdamW优化器,随机掩盖token的掩盖率设置为0.9。在微调阶段,使用AdamW优化器进行120轮的微调,基础学习率为2×10^{-3},批次大小为64。所提方法使用PyTorch 2.0.0实现,并在配备单个NVIDIA GeForce RTX3090 GPU的服务器上进行训练。

本文采用四个标准指标来评估和比较模型性能:准确率(ACcuracy,AC)、精确率(PRecision,PR)、召回率(ReCall,RC)和加权F_1分数(weighted F_1-Score,F_1)。这些指标基于混淆矩阵计算,其中TP表示将某类流量正确分类为该类的样本数,FP表示将其他类错误分类为该类的样本数,TN表示正确排除非该类流量的样本数,FN表示将该类流量错误分类为其他类的样本数。各指标的数学定义及在流量分类任务中的具体物理意义如下:

$$\mathrm{Accuracy}=\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \tag{38}$$

$$\mathrm{Precision}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \tag{39}$$

$$\mathrm{Recall}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \tag{40}$$

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{41}$$

$$\text{Weighted } F_1 = \sum_{i=1}^{K} \left(\frac{n_i}{N_{\text{total}}} \times F_{1\,i} \right) \tag{42}$$

其中，n_i表示第i类流量的实际样本数量，N_{total}表示样本的总数量，K表示类别数量。为了评估各消融变体在不同数据集上的综合泛化能力，本文还使用了宏平均（Macro Average）和平均名次（Rank）两个汇总指标，分别表示模型各项指标的算术平均值和模型在所有测试数据集上的性能排名均值。

4.3 主要结果

本文将ET-LDP与各种基准方法和最先进的方法进行了比较，具体包括以下几类。

（1）传统的机器学习方法，如AppScaner（基于加密网络流量的智能手机应用自动指纹识别系统）[4]和FlowPrint（加密网络流量上的半监督移动应用指纹识别方法）[5]，它们依赖于统计特征进行流量分类。

（2）基于CNN的流量分类方法，如基于流序列网络的加密流量分类方法（Flow Sequence Network for encrypted traffic classification，FS-Net）[6]利用原始数据包进行监督学习。

（3）基于Transformer的模型，如ET-BERT[10]、YaTC[23]、TrafficFormer[8]、FlowletFormer（网络行为语义感知的预训练流量分类模型）[52]和面向加密网络流量的多级预训练方法（Multi-Level pre-training for Encrypted Traffic Classification，MLETC）[53]。此类方法以图像或者自然语言的方式进行处理流量、捕获流量特征，并使用有限的标注数据针对特定任务进行微调。

（4）基于Mamba的模型，如NetMamba[26]是基于纯Mamba的预训练模型，将流量视为一维序列数据进行分块处理。

与其他方法进行对比的结果如表1、表2和表3所示，其中除了AppScaner、FlowPrint和FS-Net外，其他方法均为预训练模型。需要注意的是，TrafficFormer与FlowletFormer的预训练数据集采用的是ISCX-VPN2016（加拿大网络安全研究所虚拟专用网络与非虚拟专用网络流量分类数据集（2016））、CICIDS2017（加拿大网络安全研究所入侵检测系统数据集（2017））和宽带集成分布式环境（Widely Integrated Distributed Environment，WIDE）数据集。MLETC的预训练数据集使用了未公开的自行收集的数据，该数据集包含了真实网络环境中各种协议的流量。

传统的机器学习方法，如AppScaner和FlowPrint，能够利用统计特征获得一定的流量特征。在类别数量较少的数据集（如CICIoT2022）上，该类方法能够取得良好的性能。然而，由于其统计学习策略的不足，在类别数量较多的复杂数据集，例如CrossPlatform（Android）和Cross-Platform（iOS）上表现不佳。

ET-BERT、TrafficFormer、FlowletFormer和MLETC作为基于大语言模型的Transformer网络流量分类模型，在多个数据集上均表现良好，但在新加密协议的CSTN-TLS1.3数据集上的性能不高，且TrafficFormer在两个跨平台数据集上表现差距较大。说明基于自然语言方式处理的大型Transformer架构在流量分类任务中仍然存在适配性问题，导致稳定性和泛化性未能达到最优，进一步显示出混合架构ET-LDP的流量分类模型在性能和稳定性方面的优势。表1显示，在CrossPlatform（Android）和CrossPlatform（iOS）两个数据集中，ET-LDP比现有方法有了明显的改进。具体来说，在CrossPlatform（Android）上相较于大型的Transformer模型，ET-BERT的准确率和F_1分别提高了0.052 6和0.051 1。表2展示了在加密VPN加密流量和恶意流量分类任务的方法对比，可以观察到这些方法更擅长应对此类环境下的流量分类。推测可能是因为基于BURST级别的词表的嵌入方式有利于学习多层加密的数据包之间的关系。而ET-LDP在不依赖词表的情况下取得了SOTA的性能，体现了本文方法采用的混合结构能更加有效地利用流量的整体和微观信息。

YaTC模型同样基于Transformer模型，并借鉴了计算机视觉中的MAE方法，然而该方法将流量数据视为图像，可能引入不必要的空间信息干扰。相比之下，ET-LDP采用序列建模的思路，结合Mamba和Transformer的混合架构，引入Agent注意力排除干扰性信息，其性能在所有数据集中均达到了最优性能。在三个加密数据集ISCXTor2016、ISCXVPN2016和USTC-TFC2016上的表现均优于YaTC，尽管提升有限，但分类准确率均超过0.995 0，表明了ET-LDP强大的混合机制在加密流量解析中的有效性。

引入Mamba进行序列建模的NetMamba方法，在多个类别数量较多的复杂数据集上达到了次优的效果，这说明Mamba对于流量分类任务的提升效果显著，展示了合理性。然而，局限于纯Mamba的模型结构，依然与本文的ET-LDP有所差距。尤其是在CSTNET-TLS数据集上，准确率和F_1得分方面分别提高了0.012 3和0.013 2。如表3所示，ET-LDP在CSTN-TLS1.3数据集上达到了0.938 1的准确率，这说明ET-LDP在面对协议的变体时依然能够保留强大的分类能力。

4.4 消融研究

为验证ET-LDP模型各组件设计对整体性能的贡献，并进一步探究不同混合方法对模型性能的影响，在CrossPlatform（Android）、ISCXVPN2016和CSTNET-TLS

表1　在CrossPlatform(Android)、CrossPlatform(iOS)和CICIoT2022数据集上与其他方法的比较

Table 1　Comparisons of various methods on CrossPlatform(Android、iOS) and CICIoT2022 datasets

方法	CrossPlatform(Android)				CrossPlatform(iOS)				CICIoT2022			
	AC	PR	RC	F_1	AC	PR	RC	F_1	AC	PR	RC	F_1
AppScanner	0.162 6	0.164 6	0.145 6	0.141 3	0.171 8	0.140 0	0.144 0	0.128 3	0.755 6	0.809 3	0.724 4	0.693 8
FlowPrint	0.873 9	0.894 1	0.873 9	0.870 0	0.871 2	0.868 7	0.871 2	0.860 3	0.582 0	0.416 4	0.582 0	0.464 3
FS-Net	0.014 7	0.002 3	0.014 7	0.003 4	0.029 3	0.001 4	0.029 3	0.002 5	0.574 7	0.380 0	0.574 7	0.421 6
ET-BERT	0.938 6	0.945 1	0.938 6	0.940 1	0.910 5	0.880 9	0.910 5	0.885 0	0.993 7	0.993 8	0.993 7	0.993 7
YaTC	0.904 2	0.908 1	0.904 2	0.904 2	0.931 0	0.930 7	0.931 0	0.929 5	0.995 9	0.995 9	0.995 9	0.995 9
TrafficFormer	0.766 4	0.643 5	0.620 4	0.616 7	0.567 9	0.496 6	0.469 7	0.468 9	0.872 5	0.848 7	0.834 3	0.828 8
FlowletFormer	—	—	—	—	—	—	—	—	0.910 9	0.890 5	0.886 6	0.885 9
NetMamba	0.986 9	0.987 1	0.986 9	0.986 4	0.988 1	0.988 5	0.988 1	0.988 1	0.998 5	0.998 5	0.998 5	0.998 5
ET-LDP	**0.991 2**	**0.991 2**	**0.991 2**	**0.991 2**	**0.992 6**	**0.989 7**	**0.992 6**	**0.990 8**	**0.999 5**	**0.999 5**	**0.999 5**	**0.999 5**

注:最佳结果以粗体显示,下同。

表2　在ISCXTor2016、ISCXVPN2016和USTC-TFC2016数据集上与其他方法的比较

Table 2　Comparisons of various methods on ISCXTor2016, ISCXVPN2016 and USTC-TFC2016 datasets

方法	ISCXTor2016				ISCXVPN2016				USTC-TFC2016			
	AC	PR	RC	F_1	AC	PR	RC	F_1	AC	PR	RC	F_1
AppScanner	0.403 4	0.285 0	0.214 9	0.211 3	0.764 3	0.804 7	0.704 5	0.725 6	0.699 8	0.859 1	0.606 2	0.663 3
FlowPrint	0.131 6	0.017 3	0.131 6	0.030 6	0.966 6	0.973 3	0.966 6	0.968 1	0.799 2	0.774 5	0.799 2	0.775 5
FS-Net	0.702 0	0.701 0	0.702 0	0.699 9	0.702 3	0.748 7	0.702 3	0.666 0	0.438 1	0.201 1	0.438 1	0.267 2
ET-BERT	0.998 0	0.998 1	0.998 0	0.998 0	0.956 6	0.956 6	0.956 6	0.956 5	0.991 0	0.991 1	0.991 0	0.991 0
YaTC	0.995 9	0.995 9	0.995 9	0.995 9	0.981 9	0.982 0	0.981 9	0.981 9	0.994 7	0.974 9	0.974 7	0.973 4
TrafficFormer	0.866 9	0.754 5	0.746 0	0.747 2	0.853 3	0.844 5	0.834 8	0.827 9	0.975 0	0.978 9	0.975 0	0.974 6
TrafficLLM	—	0.981 0	0.987 1	0.981 0	—	0.996 0	0.997 0	0.996 0	—	—	—	—
FlowletFormer	0.921 5	0.926 3	0.904 3	0.911 6	0.940 0	0.947 1	0.927 7	0.936 4	0.965 0	0.968 9	0.965 0	0.964 8
MLETC	0.994 8	0.977 8	0.995 4	0.986 5	0.987 5	0.984 7	0.984 7	0.988 0	0.990 2	0.993 5	0.991 8	0.992 6
NetMamba	0.999 3	0.999 3	0.999 3	0.999 3	0.989 9	0.989 9	0.989 9	0.989 9	0.999 0	0.999 1	0.999 0	0.999 0
ET-LDP	**0.999 8**	**0.999 8**	**0.999 8**	**0.999 8**	**0.998 6**	**0.998 6**	**0.998 6**	**0.998 6**	**1.000 0**	**1.000 0**	**1.000 0**	**1.000 0**

表3　在CSTNET-TLS数据集上与其他方法的比较

Table 3　Comparisons of various methods on CSTNET-TLS 1.3 datasets

方法	CSTNET-TLS			
	AC	PR	RC	F_1
AppScanner	0.744 1	0.723 2	0.696 3	0.702 3
FS-Net	0.781 4	0.767 0	0.731 6	0.731 1
ET-BERT	0.799 3	0.783 2	0.768 9	0.770 0
YaTC	0.839 1	0.836 4	0.810 1	0.814 0
TrafficFormer	0.798 2	0.788 3	0.773 6	0.770 4
FlowletFormer	0.860 5	0.857 8	0.844 5	0.847 3
NetMamba	0.925 8	0.927 1	0.925 8	0.924 9
ET-LDP	**0.938 1**	**0.939 3**	**0.938 1**	**0.938 1**

上进行了各项消融实验,综合展示各组件在应用分类、加密VPN流量和新协议流量场景下的表现。

4.4.1　混合架构的影响研究

为探究各阶段Mamba与Transformer两种模块的顺序关系,本文进行了多种排列实验,如表4所示。其中,S表示softmax注意力层,A表示Agent注意力层,M表示Mamba层。为探究第一阶段模块的影响,在保持后续阶段结构一致的前提下,将所提方法的编码器第一阶段的注意力模块分别替换为Mamba、Agent注意力和softmax自注意力,进行消融实验对比分析。从M MA MMAS MMMMAASS、S MA MMAS MMMMAASS和A MA MMAS MMMMAASS这三种模式结果看出,使用Mamba作为首层时,模型在整体的准确率均表现稳定,优于其他设置。而将softmax自注意力或Agent注意力置于首层时,模型在CrossPlatform(Android)和CSTNET-TLS两个数据集上存在性能波动,F_1差距最大达到了0.070 5。分析认为,这可能是由于softmax自注意力和Agent注意力机制无法捕获全面的流量token的顺序关系,它们在捕捉局部关键特征方面相对不足,使得在不同结构或分布的数据集中表现出不稳定性。而Mamba结合位置编码的方法在获取全面的流量细粒度特征和流量模式方面更具优势。因此,在最终架构中,将单层Mamba作为模型的初始阶段,以保证模型在起始阶段能够提供稳定而充分的上

下文捕捉能力。除了第一阶段外，后续模块均采用混合 Mamba-Transformer 架构，进一步探讨了两种模块的顺序关系。在保持混合注意力层内部顺序不变的情况下，仅改变混合注意力层与 Mamba 的顺序。实验结果显示，在混合模块的中后段，Mamba-Transformer 顺序的调整对最终性能影响相对有限。通过观察模式 M AM ASMM AASSMMMM 和 M MA MMAS MMMMAASS（本文）的结果，所提方法关于编码器的中后阶段“优先引入 Mamba 用以捕捉足够的局部特征与更广泛的流模式”的设计逻辑，在三个数据集中均成立，但其性能优势并非绝对，例如二者均在数据集 ISCXVPN2016 上实现了 0.998 6 的精度。这可能受到具体数据结构和任务类型的影响，特别是在网络流量数据中，上下文顺序的弱相关性导致不同模块对时序建模的敏感程度存在差异。综合来看，推荐采用默认结构，即模式 M MA MMAS MMMMAASS，其在多个任务上表现更为均衡，且更契合所提出的流量表示建模逻辑。

4.4.2 混合注意力机制的影响研究

本文采用了混合注意力机制，对 Agent 注意力和 softmax 注意力在编码器中的关系进行消融研究。如表 4 所示，以最优设置，即 M MA MMAS MMMMAASS 模式（本文）为基准，调整混合注意力层内部的顺序。

表 4 混合架构的排列顺序消融实验

Table 4 Ablation study on the arrangement order of hybrid architectures

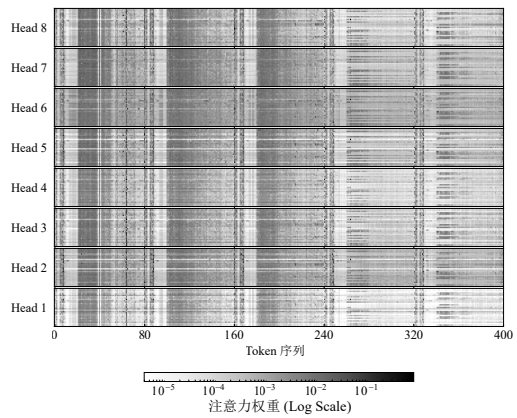
混合模式	CrossPlatform(Android)				ISCXVPN2016				CSTNET-TLS			
	AC	PR	RC	F_1	AC	PR	RC	F_1	AC	PR	RC	F_1
S MA MMAS MMMMAASS	0.990 3	0.990 3	0.990 3	0.989 8	0.997 8	0.997 8	0.997 8	0.997 8	0.932 5	0.933 4	0.932 5	0.932 1
A MA MMAS MMMMAASS	0.991 0	0.990 9	0.991 0	0.990 5	0.997 8	0.997 8	0.997 8	0.990 4	0.920 4	0.921 6	0.920 4	0.920 0
M AM ASMM AASSMMMM	0.989 0	0.989 2	0.989 0	0.988 9	0.998 6	0.998 6	0.998 6	0.998 6	0.920 4	0.926 6	0.920 4	0.921 0
M MS MMSA MMMMSSAA	0.990 8	0.990 6	0.990 8	0.990 3	0.997 8	0.997 8	0.997 8	0.997 8	0.934 2	0.936 7	0.934 2	0.934 0
M MA MMAA MMMMAAAA	0.989 5	0.989 4	0.989 5	0.989 0	0.996 4	0.996 4	0.996 4	0.996 4	0.935 7	0.937 7	0.935 7	0.935 7
M MS MMSS MMMMSSSS	0.990 1	0.990 4	0.990 1	0.989 9	0.996 4	0.996 4	0.996 4	0.996 4	0.921 3	0.923 7	0.921 3	0.921 1
M MA MMAS MMMMAASS(本文)	**0.991 2**	**0.991 2**	**0.991 2**	**0.991 2**	**0.998 6**	**0.998 6**	**0.998 6**	**0.998 6**	**0.938 1**	**0.939 3**	**0.938 1**	**0.938 1**

结果显示“softmax 注意力优先，Agent 靠后”的排列模式即 M MS MMSA MMMMSSAA，其整体性能略低。这一结果印证了方法设计中的假设：Agent 注意力能够先聚合冗余信息、简化 token 关系，再由 softmax 注意力完成更精细的全局建模，从而更适应结构重复、不均衡的网络流量数据特性，提升建模完整性和分类效果。

为了证明混合注意力机制的有效性，选择单一 Agent 注意力或者 softmax 注意力进行实验。具体来说，将除第一阶段外所有的 softmax 自注意力替换为 Agent 注意力，即模式 M MA MMAA MMMMAAAA，其整体性能略低于本文模型混合结构（M MA MMAS MMMMAASS），将所有的注意力层都选择 softmax 自注意力层时，即模式 M MS MMSS MMMMSSSS，在 CrossPlatform（Android）数据集上达到了 0.990 1 的准确率，且在剩余两个数据集上的表现远不如 ET-LDP 的设置。值得注意的是，由于 TLS 1.3 加密更彻底，CSTNET-TLS 对字节语义更敏感[10]，此时 M MA MMAA MMMMAAAA 的结果明显优于 M MS MMSS MMMMSSSS。结果表明，Agent 注意力在处理加密流量的冗余结构和局部聚合方面的优秀能力在流量分类任务中更占优势，但在 CrossPlatform（Android）上略有差距，这也反映出 Agent 注意力在全局建模能力上相较于 softmax 注意力仍存在一定差距。因此，只需引入少量 softmax 注意力便可实现在不显著增加复杂度的情况下提升性能，从而印证了混合机制的可行性。

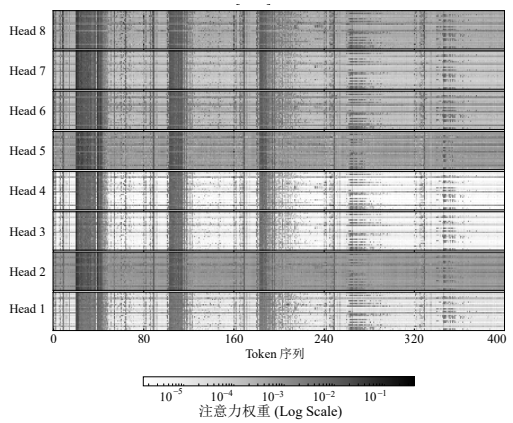
为了更加清晰地对比采用 Agent 注意力的模型和仅使用 softmax 注意力的模型之间的差别，实验展示了基于 CSTNET-TLS 数据集的最终层中[CLS]标记的注意力图。图 7（a）和图 7（b）中分别展示了模式 M MA MMAS MMMMAASS 和模式 M MS MMSS MMMMSSSS 在 CSTNET-TLS 1.3 数据集上最后一层[CLS]标记的注意力。横轴表示 5 个数据包，每个间隔包含 80 个 token（320 字节）的数据。纵轴显示了 8 个注意力头，每个间隔内是一个批量数据，批量大小为 32。采用 Agent Attention 的模型 M MA MMAS MMMMAASS 的注意力稀疏性和特征显著性方面明显优于仅使用 softmax 注意力的模型 M MS MMSS MMMMSSSS。首先，从共性特征来看，两组热力图均在 X 轴的特定位置呈现出显著的深色垂直条纹，且这些条纹的分布随数据包位置的变化而动态调整。这表明两种模型结构均具备捕捉加密流量中的强语义特征（如协议头关键字段）的能力，且能够反映出数据流前段信息密度较高的特性。然而，在注意力分布的质量与抗噪性上，两者存在差异。图 7（a）展现了较高的信噪比与全局一致性。其垂直条纹边缘锐利，覆盖区域更广且过渡平滑，条纹间的背景区域几乎纯净，表明模型对关键特征的关注高度集中。此外，所有注意力头的表现高度协同，均清

晰地聚焦于语义关键区,未出现发散现象。相比之下,图7(b)的注意力分布虽然部分位置的响应强度极高,但纹理边缘存在突兀的跳变,背景区域充斥大量噪点。更为显著的是,部分注意力头呈现出明显的横向带状分布,甚至在整个流区域内权重趋于均一,表明这些头部未能提取到有效特征,出现了注意力坍塌或冗余。



(a) M MA MMAS MMMMAASS最终层中[CLS]标记的注意力图

(a) Attention map marked with [CLS] in the final layer ofMMA MMAS MMMMAASS



(b) M MS MMSS MMMMSSSS最终层中[CLS]标记的注意力图

(b) Attention map with [CLS] marker in the final layer of M MS MMSS MMMMSSSS

图7　Agent注意力与softmax注意力在CSTNET-TLS数据集上最终层中[CLS]标记的注意力图对比

Figure 7　Comparison of Agent attention and softmax attention heatmaps at the final layer [CLS] token on the CSTNET-TLS 1.3 dataset

上述现象说明,Agent注意力增强了全局语义聚合,通过Agent token的中转,模型能够更有效地捕捉长距离依赖,将分散的字节级特征聚合成强语义的流级特征。这种机制天然具备去噪能力,有效抑制了加密流量中高熵负载带来的随机干扰。而标准softmax注意力在处理流量序列时容易受到噪声影响,这可能会导致模型在部分场景下的泛化能力变弱,或者被某些样本的噪声主导。

4.4.3　模型组件分析

为系统地评估所提出的混合Mamba-Transformer架构中各个关键组件的有效性,本文进行了一系列详尽的消融实验。ET-LDP整合了三个核心组件:(1)在Mamba模块的最后一层引入MoE[Mamba-MoE(Last)];(2)从Mamba层到注意力层的跨结构过渡M2T-Bridge;(3)从注意力层到Mamba层的跨结构过渡T2M-Bridge。实验以一个不包含上述任何组件的基线模型作为基准,并在CrossPlatform(Android)、ISCXVPN2016和CSTNET-TLS数据集上进行评估。

评估MoE层的引入对模型性能的影响。如表5所示,与基线相比,仅在Mamba模块的最后一层添加MoE,在所有三个数据集上均带来了性能提升。尤其是在最具挑战性的CSTNET-TLS数据集上,F_1分数由0.929 3显著提升至0.937 0。这证明了MoE结构通过其专家路由机制,提升了字节/字段级差异可分性。相比之下,在所有Mamba层中都使用MoE[Mamba-MoE(All)]的效果并不理想。虽然它在CrossPlatform(Android)上达到了0.991 5,但在ISCXVPN2016上的表现甚至低于基线模型。这表明,过多地使用MoE可能会增加模型优化的难度,或导致对特定数据分布的过拟合,从而损害了模型的泛化性。为此,本文尝试引入负载均衡损失[54]来减轻这个问题,权重为0.01。结果如表6所示,Mamba-MoE(Last)添加负载均衡后的整体性能回落。推测是因为在流量任务中,最后一层的MoE层本就需要非均匀的路由,使得某些专家专注少数难类或长尾模式以应对数据不平衡的情况,且专家数量较少。在此情况下,负载均衡倾向把路由推向均匀使用,削弱了这种必要的不均匀性,导致能力下降。在全层MoE的设置中,由于路由在多层叠加,小幅的负载均衡可抑制层间路由误差累积。例如,在ISCXVPN/TLS这类协议和行为模式较固定数据上带来有限的正向增益,但是提升幅度较小。

为进一步展示MoE层引入对模型的影响,图8和图9分别展示了Mamba-MoE(Last)和Mamba-MoE(All)两种情况下在CSTNET-TLS 1.3数据集上专家激活分布的热力图。图中横轴对应5个连续数据包(每包涵盖80个token),纵轴代表4个专家,色彩亮度反映了专家在一个批量(32个样本)上的平均激活概率。首先,图8展示了Mamba-MoE(Last)在无显式负载均衡约束下的激活分布。可以观察到,在同一个批量数据中,每个专家热力图呈现出显著且清晰的垂直

表5 混合架构的排列顺序消融实验

Table 5 Ablation study of each component

模式	CrossPlatform(Android)				ISCXVPN2016				CSTNET-TLS				AVG	
	AC	PR	RC	F_1	AC	PR	RC	F_1	AC	PR	RC	F_1	Macro Average	Rank
基线模型	0.990 1	0.990 3	0.990 1	0.989 8	0.996 4	0.996 4	0.996 4	0.996 4	0.929 3	0.931 5	0.929 3	0.929 3	0.972 1	6.46
+ Mamba-MoE(Last)	0.990 1	0.990 1	0.990 1	0.989 8	0.997 8	0.997 8	0.997 8	0.997 8	0.937 0	0.939 2	0.937 0	0.937 0	0.975 1	4.33
+ Mamba-MoE(All)	**0.991 8**	**0.991 7**	**0.991 8**	**0.991 5**	0.995 7	0.995 7	0.995 7	0.995 6	0.935 1	0.936 7	0.935 1	0.934 7	0.974 3	4.00
+ M2T-Bridge	0.990 8	0.990 7	0.990 8	0.990 3	0.998 6	0.998 6	0.998 6	0.998 6	0.932 5	0.934 0	0.932 5	0.932 3	0.974 0	4.08
+T2M-Bridge	0.990 8	0.990 8	0.990 8	0.990 5	0.997 1	0.997 1	0.997 1	0.997 1	0.937 2	0.939 1	0.937 2	0.936 9	0.975 2	3.79
+ M2T/T2M	0.991 4	0.991 3	0.991 4	0.990 9	0.997 8	0.997 9	0.997 8	0.997 8	0.932 7	0.935 7	0.935 7	0.932 8	0.974 2	3.58
ET-LDP	0.991 2	0.991 2	0.991 2	0.991 2	**0.998 6**	**0.998 6**	**0.998 6**	**0.998 6**	**0.938 1**	**0.939 3**	**0.938 1**	**0.938 1**	**0.976 1**	**1.75**

表6 引入负载均衡损失的混合架构的排列顺序消融实验

Table 6 Ablation study of the arrangement order for the hybrid architecture incorporating load balancing loss

模式	CrossPlatform(Android)				ISCXVPN2016				CSTNET-TLS			
	AC	PR	RC	F_1	AC	PR	RC	F_1	AC	PR	RC	F_1
基线模型	0.990 1	0.990 3	0.990 1	0.989 8	0.996 4	0.996 4	0.996 4	0.996 4	0.929 3	0.931 5	0.929 3	0.929 3
+ Mamba-MoE(Last)	0.990 1	0.990 1	0.990 1	0.989 8	**0.997 8**	**0.997 8**	**0.997 8**	**0.997 8**	**0.937 0**	**0.939 2**	**0.937 0**	**0.937 0**
+ Mamba-MoE(Last) + 负载均衡损失	0.990 3	0.990 3	0.990 3	0.989 8	0.997 1	0.997 1	0.997 1	0.997 1	0.929 0	0.931 1	0.929 0	0.928 9
+ Mamba-MoE(All)	**0.991 8**	**0.991 7**	**0.991 8**	**0.991 5**	0.995 7	0.995 7	0.995 7	0.995 6	0.935 1	0.936 7	0.935 1	0.934 7
+ Mamba-MoE(All) + 负载均衡损失	0.989 9	0.990 1	0.989 9	0.989 6	0.997 1	0.997 1	0.997 1	0.995 6	0.935 3	0.936 6	0.935 3	0.934 8

条纹特征，表明专家之间形成了明确的语义分工。Expert 2（倒数第二行）其高亮区域集中在特定的X轴区间。结合流量结构分析，这些位置往往是每个包的头部字段或特定字段，如“TLS证书片段”或“协议头部字段”等特定语义。相比之下，其余专家呈现出一种类似弥散的背景分布，覆盖了Expert 2未激活的区域，且在处理相同的数据片段时，激活强度存在明显差异，形成了一种交错覆盖的互补状态。这表明模型在处理无明显结构的加密负载时，依然学习到了潜在的细粒度特征分布，形成了分而治之的策略机制，而非简单地将其视为随机噪声。这与加密流量的物理特性高度吻合，即特定专家（Expert 2）负责处理结构化的明文或半明文头部，而其余专家负责处理高熵的加密负载。此外，细致观察热力图可以发现，由于不同数据包的报头长度、选项字段或应用层协议存在差异，专家激活的边界在不同样本间存在微小、动态的锯齿状交错偏移。这证明了路由决策是动态依赖于当前token实际内容的，而非绝对位置死记。同时也证明了MoE（Last）模式能够有效地将不同语义的token路由给最擅长的专家。相反，图9揭示了Mamba-MoE（All）模式中存在的专家坍塌现象。如图9（b）所示，Expert 2在最后一个阶段表现为横贯整个流序列的高频激活带，主导了几乎所有token的处理，而剩余专家处于抑制状态。这种分布意味着路由丧失了对不同语义区域的辨别能力。随着网络层数的加深，全层MoE结构在浅层引入的路由噪声逐渐累积，导致特征表征在深层趋于同质化。这种误差累积削弱了模型对特定语义结构的敏感度，最终导致最后一层的专家分工失效。

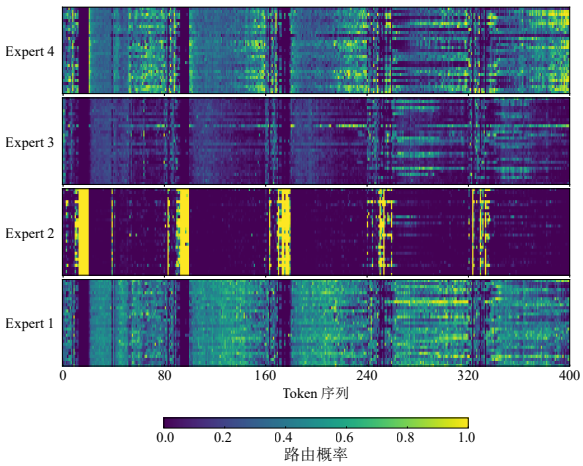
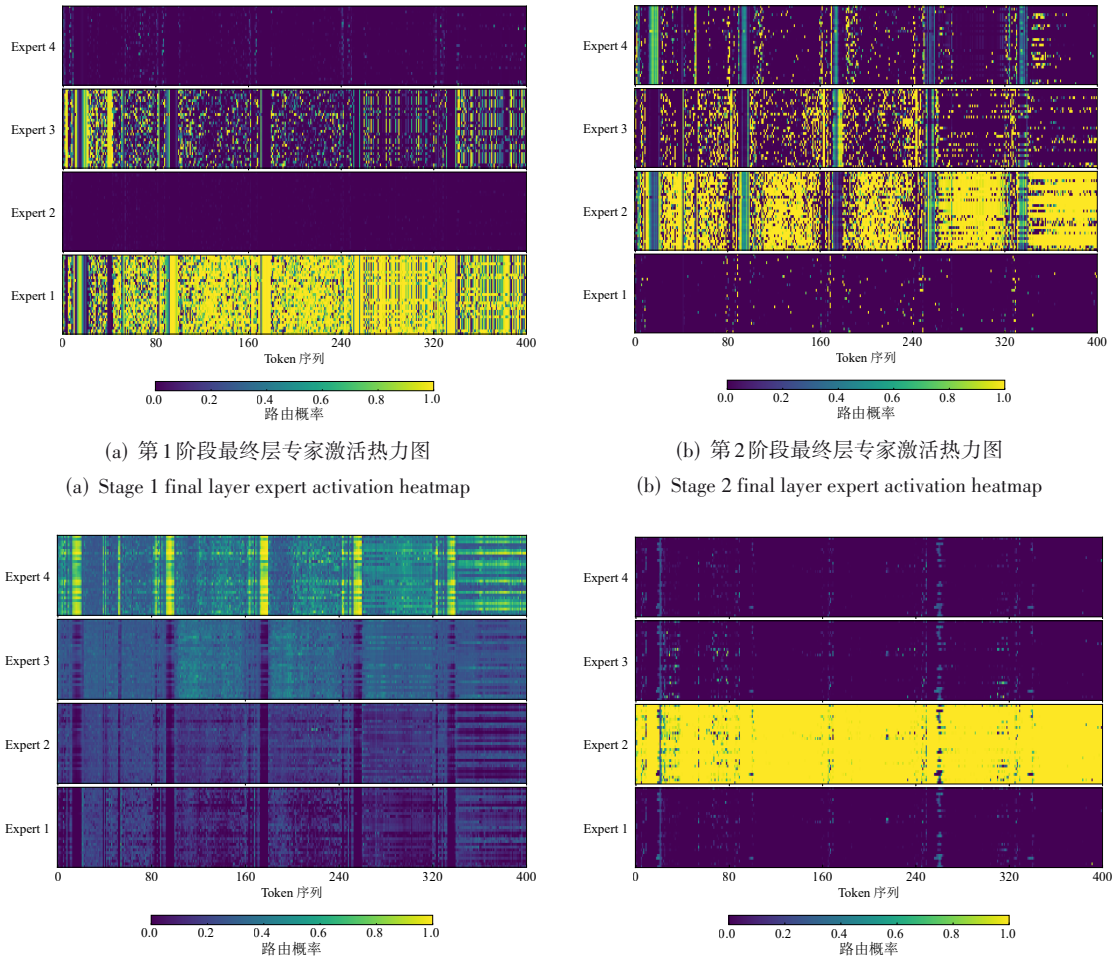


图8 MoE(Last)在CSTNET-TLS 1.3数据集上最终层专家激活热力图

Figure 8 The final layer expert activation heatmap of MoE (Last) on CSTNET-TLS 1.3 dataset

在推理效率方面，如表7所示，在批大小为64的条件下，比较了Mamba-MoE（Last）与Mamba-MoE（All）的推理效率。实验统计了模型参数量、推理吞吐率



(a) 第1阶段最终层专家激活热力图

(a) Stage 1 final layer expert activation heatmap

(b) 第2阶段最终层专家激活热力图

(b) Stage 2 final layer expert activation heatmap

(c) 第3阶段最终层专家激活热力图

(c) Stage 3 final layer expert activation heatmap

(d) 第4阶段最终层专家激活热力图

(d) Stage 4 final layer expert activation heatmap

图9　MoE(All)在CSTNET-TLS 1.3数据集上各阶段最终层专家激活热力图

Figure 9　The final layer expert activation heatmaps of MoE (All) mode at each stage on the CSTNET-TLS 1.3 dataset

(样本/秒)以及显存峰值。由结果可知,随着MoE层数的增加,模型参数量仅从52.17 × 10^6增加至53.76 × 10^6(约3.1%),显存峰值也仅提升1.1%。然而推理速度下降显著,Mamba-MoE(All)的吞吐率为632样本/秒,比Mamba-MoE(Last)的909样本/秒降低约30.5%,相当于推理延迟增加约44%。这说明MoE层数量并不是影响显存的主要因素,而是导致计算效率显著下降的关键瓶颈。综合效率与精度消融结果,Mamba-MoE(Last)在保证精度提升的同时,显著降低了推理代价。其仅在最后一层引入专家结构,既能充分发挥MoE面对流量数据的字节/字段级建模优势,又避免了多层路由带来的重复计算与带宽开销。因此,本文的方法选择仅在Mamba模块的最后一层引入MoE作为默认配置,以实现精度和效率的最优平衡。

跨架构桥接模块的有效性。如表5所示,单独添加M2T-Bridge或T2M-Bridge都能在几乎所有数据集上超越基线模型,这有力地证明了在Mamba和Transformer之间进行信息过渡是至关重要的。同时,观察到了一个特化现象:单独的M2T-Bridge在ISCXVPN2016上取得了0.998 6的高分,而单独的T2M-Bridge则在

表7　批大小 = 64下的Mamba-MoE组件推理效率评估

Table 7　Efficiency evaluation of the Mamba-MoE component with batch size = 64

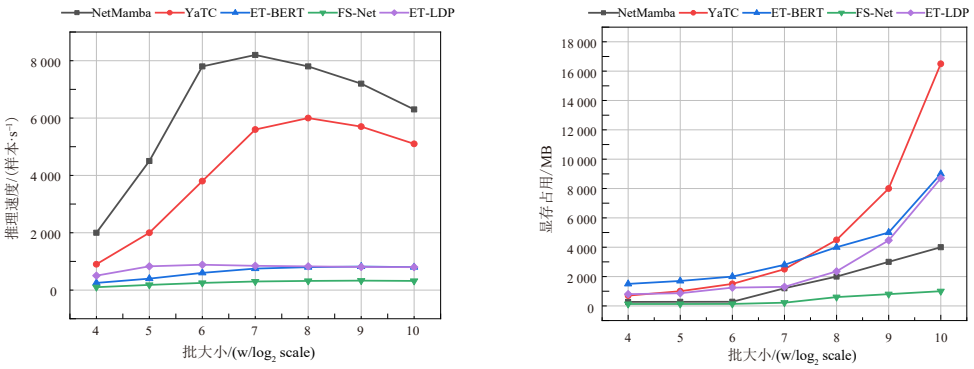
模式	参数量/×10^6		推理速度/(样本·s^{-1})	显存占用峰值/×10^6
	预训练	微调		
基线模型	19.51	19.23	1 678.460	672.369
+Mamba-MoE (Last)	52.45	52.17	909.955	801.046
+Mamba-MoE (All)	54.04	53.76	632.396	810.098

CSTNET-TLS上表现突出(0.936 9),甚至优于两者共存的M2T/T2M-Bridge配置。这种现象表明,不同的数据集可能对特定方向的信息流有不同的偏好。例如,ISCXVPN2016的流量特性可能更依赖Mamba提取的广泛流模式与全局依赖向Transformer的传递;而TLS类的CSTNET-TLS数据集对字节语义更敏感,则更需要Transformer的全局上下文反馈给Mamba。然而,这种特化的泛化能力是受限的。例如,M2T-Bridge在CSTNET-TLS上相对于基线模型的增益最低,准确率和F_1增益分别为0.003 2和0.003 0。

消融实验清晰地表明,尽管单一组件可能在特定数据集上展现出优势,然而最终模型ET-LDP综合表现最好。从跨数据集的宏平均与平均名次看,ET-LDP在总体上也体现出更加稳健的特点。实验结果证明了所提出的各个组件对于构建一个强大的网络流量分类模型均为不可或缺的,它们之间存在着积极的协同效应。

4.4.4 推理效率评估

为了评估ET-LDP的推理效率,本文针对速度和GPU内存消耗与现有的学习方法进行了比较实验。如图10(a)所示,ET-LDP的速度曲线较平滑,处于中间水平。观察到在批大小由64~1 024过程中,吞吐量仅小幅下降,这说明在ET-LDP采用的混合架构中,Mamba与混合注意力模块已较好地并行化,在批量放大时没有出现明显的内存和带宽消耗。其他方法中,NetMamba与YaTC在大多数批大小下保持最高吞吐,而ET-BERT与FS-Net明显偏慢。图10(b)为显存占用方面结果对比。在16~256批大小下,ET-LDP的显存处于中等水平,显著低于标准自注意力模型ET-BERT和YaTC,但高于极度轻量的RNN系列模型FS-Net与高度优化的SSM模型NetMamba。例如,在批大小为64时,ET-LDP显存大约仅为ET-BERT的0.6倍,YaTC的0.8倍。随着批量大小继续增大,所有模型显存都近似按批大小线性增长。此时,ET-LDP的增长斜率仍可控,符合混合架构的计算复杂度。尽管ET-BERT在速度与显存上都不是极端最优,但曲线更平稳与均衡,受批量大小影响较小,为后续的扩展提供了可行的资源空间。



(a) 推理速度对比 (b) 显存占用对比

(a) Inference speed comparison (b) Memory usage comparison

图10 与其他模型的推理效率比较

Figure 10 Comparison of inference efficiency with other models

为更加直观体现各方法的综合性能,进一步在批大小为64的统一设置下比较单点效率与性能。如图11所示,横轴为显存,纵轴为速度,颜色与数值为CSTNET-TLS数据集的加权F_1。从效率与准确率的综合效用考虑,ET-LDP在保持低资源的前提下,取得了最高F_1,是更均衡稳妥的流量分类模型方案。这与本文架构取舍一致,用Mamba和近线性注意力的Agent保证整体效率,结合少量softmax注意力补齐全局上下文信息,并将MoE控制在Mamba末层,实现性能与效率的均衡。

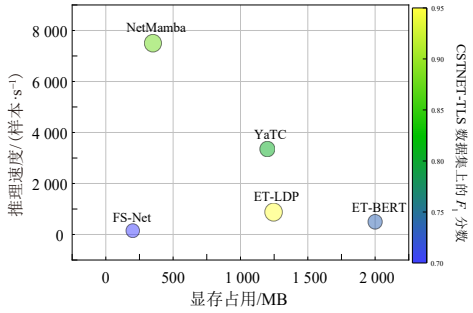


图11 批量64时的推理效率比较

Figure 11 Comparison of inference efficiency when batch size is 64

5　结论

本文针对现有网络流量分类模型在效率与性能之间存在的矛盾问题,摒弃了传统的单一架构或简单的模块拼接,提出了一种ET-LDP方法。首先,提出状态空间与注意力的跨机制协同范式,通过合理串联Mamba模块、Agent注意力与Softmax注意力,有效发挥了不同机制在结构建模与远程依赖捕捉方面的优势互补。其次,针对加密流量多粒度语义并存的异构特性,在Mamba层中引入MoE进行动态路由,在提升模型对复杂载荷的局部动态感知能力的同时,保持了推理阶段的参数高效性。最后,基于自监督预训练与微调的范式,使得模型在应对大规模未标记流量数据和下游任务时具备更强的适应能力。在7个公开数据集上的实验表明,ET-LDP在微调阶段达到了最佳性能,并且在推理效率上依然保留较高的竞争力,为利用混合架构进行网络流量分类开辟了新的途径。尽管如此,真实的网络环境是开放且动态变化的,常常伴随着未知应用、新协议或零日攻击的出现。后续工作将结合开集识别与持续学习技术,以应对网络流量的持续演进。除了经典的流量分类任务外,未来计划将ET-LDP的应用范围扩展至服务质量预测、网络性能评估及异常流量检测等更多网络管理与安全任务中,并需进一步提升模型的计算效率。

参考文献

[1] Bujlow T, Carela-Español V, Barlet-Ros P. Independent comparison of popular DPI tools for traffic classification[J]. Computer Networks, 2015, 76: 75-89.

[2] 任杰, 王洪超, 王钦定, 等. 服务感知网络[J]. 电子学报, 2025, 53(2): 371-384.
Ren Jie, Wang Hongchao, Wang Qinding, et al. Service aware network[J]. Acta Electronica Sinica, 2025, 53(2): 371-384. (in Chinese)

[3] 徐鹏, 林森. 基于C4.5决策树的流量分类方法[J]. 软件学报, 2009, 20(10): 2692-2704.
Xu Pen, Lin Sen. Internet traffic classification using C4.5 decision tree[J]. Journal of Software, 2009, 20(10): 2692-2704. (in Chinese)

[4] Taylor V F, Spolaor R, Conti M, et al. Robust smartphone app identification via encrypted network traffic analysis[J]. IEEE Transactions on Information Forensics and Security, 2018, 13(1): 63-78.

[5] Van Ede T, Bortolameotti R, Continella A, et al. FlowPrint: Semi-supervised mobile-app fingerprinting on encrypted network traffic[C/OL]//Proceedings of the 27th Annual Network and Distributed System Security Symposium, 2020. https://dblp.org/db/conf/ndss/ndss2020.html#conf/ndss/EdeBCRDLCSP20.

[6] Liu Chang, He Longtao, Xiong Gang, et al. FS-Net: A flow sequence network for encrypted traffic classification[C]// Proceedings of the IEEE Conference on Computer Communications. Piscataway: IEEE, 2019: 1171-1179.

[7] 张小莉, 程光, 张慰慈. 基于改进深度卷积神经网络的网络流量分类方法[J]. 中国科学: 信息科学, 2021, 51(1): 56-74.
Zhang Xiaoli, Cheng Guang, Zhang Weici. Network traffic classification method based on improved deep convolutional neural network[J]. Scientia Sinica Informationis, 2021, 51(1): 56-74. (in Chinese)

[8] Zhou Guangmeng, Guo Xiongwen, Liu Zhuotao, et al. TrafficFormer: An efficient pre-trained model for traffic data[C]// 2025 IEEE Symposium on Security and Privacy (SP). Piscataway: IEEE, 2025: 1844-1860.

[9] Cui Tianyu, Lin Xinjie, Li Sijia, et al. TrafficLLM: Enhancing large language models for network traffic analysis with generic traffic representation[PP/OL]. V2. arXiv (2025-04-15) [2026-02-25]. https://arxiv.org/abs/2504.04222v2.

[10] Lin Xinjie, Xiong Gang, Gou Gaopeng, et al. ET-BERT: A contextualized datagram representation with pre-training transformers for encrypted traffic classification[C]// Proceedings of the ACM Web Conference 2022. New York: ACM, 2022: 633-642.

[11] Gu A, Dao T. Mamba: Linear-time sequence modeling with selective state spaces[PP/OL]. V2. arXiv (2024-05-31) [2025-11-13]. https://arxiv.org/abs/2312.00752v2.

[12] Zhao Ruijie, Zhan Mingwei, Deng Xianwen, et al. A novel self-supervised framework based on masked autoencoder for traffic classification[J]. IEEE/ACM Transactions on Networking, 2024, 32(3): 2012-2025.

[13] Wang Qineng, Qian Chen, Li Xiaochang, et al. Lens: A foundation model for network traffic in cybersecurity[PP/OL]. V3. arXiv (2024-03-29) [2025-11-13]. https://arxiv.org/abs/2402.03646v3.

[14] 何帅, 张京超, 徐笛, 等. 基于对比学习和预训练Transformer的流量隐匿数据检测方法[J]. 通信学报, 2025, 46(3): 221-233.
He Shuai, Zhang Jingchao, Xu Di, et al. Traffic concealed data detection method based on contrastive learning and pre-trained Transformer[J]. Journal on Communications, 2025, 46(3): 221-233. (in Chinese)

[15] Park J, Park J, Xiong Zheyang, et al. Can mamba learn how to learn: A comparative study on in-context learning tasks[C/OL]//Proceedings of the 41st International Conference on Machine Learning, 2024: 39793-39812. https://dblp.uni-trier.de/db/conf/icml/icml2024.html#conf/icml/ParkPXLCO0P24.

[16] Lieber O, Lenz B, Bata H, et al. Jamba: A hybrid transformer-mamba language model[PP/OL]. V2. arXiv (2024-07-03) [2025-11-13]. https://arxiv.org/abs/2403.19887v2.

[17] Hatamizadeh A, Kautz J. MambaVision: A hybrid mamba-transformer vision backbone[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2025: 25261-25270.

[18] Shi Jingze, Xie Ting, Wu Bingheng, et al. OTCE: Hybrid SSM and attention with cross domain mixture of experts to construct observer-thinker-conceiver-expresser[PP/OL]. V3. arXiv (2024-07-20) [2025-11-13]. https://arxiv.org/abs/2406.16495v3.

[19] Su Jinzhao, Huang Zhenhua, Wang Changdong, et al. MLSA4Rec: Mamba combined with low-rank decomposed self-attention for sequential recommendation[J]. IEEE Transactions on Computational Social Systems, 2026, 13(2): 1819-1834.

[20] Han Dongchen, Ye Tianzhu, Han Yizeng, et al. Agent attention: On the integration of softmax and linear attention[C]//Proceedings of the 18th European Conference on Computer Vision. Heidelberg: Springer, 2025: 124-140.

[21] He Hongye, Yang Zhiguo, Chen Xiangning. PERT: Payload encoding representation from transformer for encrypted traffic classification[C]//Proceedings of the 2020 ITU Kaleidoscope: Industry-Driven Digital Transformation (ITU K). Piscataway: IEEE, 2020: 9303204.

[22] Meng Xuying, Lin Chungang, Wang Yequan, et al. NetGPT: Generative pretrained transformer for network traffic[PP/OL]. V3. arXiv (2025-08-28) [2025-11-13]. https://arxiv.org/abs/2304.09513v3.

[23] Zhao Ruijie, Zhan Mingwei, Deng Xianwen, et al. Yet another traffic classifier: A masked autoencoder based traffic transformer with multi-level flow representation[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2023, 37(4): 5420-5427.

[24] Liu Xiao, Zhang Chenxu, Huang Fuxiang, et al. Vision mamba: A comprehensive survey and taxonomy[J]. IEEE Transactions on Neural Networks and Learning Systems, 2026, 37(2): 505-525.

[25] Waleffe R, Byeon W, Riach D, et al. An empirical study of mamba-based language models[PP/OL]. V1. arXiv (2024-06-12) [2025-11-13]. https://arxiv.org/abs/2406.07887v1.

[26] Wang T, Xie X, Wang W, et al. Netmamba: Efficient network traffic classification via pre-training unidirectional mamba[C]//2024 IEEE 32nd International Conference on Network Protocols. Piscataway: IEEE, 2024: 10858569.

[27] Katharopoulos A, Vyas A, Pappas N, et al. Transformers are RNNs: Fast autoregressive transformers with linear attention[C/OL]//Proceedings of the 37th International Conference on Machine Learning, 2020: 5156-5165. https://dblp.org/db/conf/icml/icml2020.html#conf/icml/KatharopoulosV020.

[28] Qin Zhen, Sun Weixuan, Deng Hui, et al. cosFormer: Rethinking Softmax in attention[C/OL]//Proceedings of the 10th International Conference on Learning Representations, 2022: 8304-8318. https://dblp.org/db/conf/iclr/iclr2022.html#conf/iclr/QinSDLWLYKZ22.

[29] Choromanski K M, Likhosherstov V, Dohan D, et al. Rethinking attention with performers[C/OL]//Proceedings of the 9th International Conference on Learning Representations, 2021: 1060-1097. https://dblp.org/db/conf/iclr/iclr2021.html#conf/iclr/ChoromanskiLDSG21.

[30] Shen Zhuoran, Zhang Mingyuan, Zhao Haiyu, et al. Efficient attention: Attention with linear complexities[C]//Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. Piscataway: IEEE, 2021: 3530-3538.

[31] Xiong Yunyang, Zeng Zhanpeng, Chakraborty R, et al. Nyströmformer: A nyström-based algorithm for approximating self-attention[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35(16): 14138-14148.

[32] Lu Jiachen, Yao Jinghan, Zhang Junge, et al. SOFT: Softmax-free transformer with linear complexity[C]//Proceedings of the 35th International Conference on Neural Information Processing Systems. New York: ACM, 2021: 1629.

[33] Han Dongchen, Pan Xuran, Han Yizeng, et al. FLatten transformer: Vision transformer using focused linear attention[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway: IEEE, 2023: 5938-5948.

[34] Jacobs R A, Jordan M I, Nowlan S J, et al. Adaptive mixtures of local experts[J]. Neural Computation, 1991, 3(1):

79-87.

[35] Jordan M I, Jacobs R A. Hierarchical mixtures of experts and the EM algorithm[J]. Neural Computation, 1994, 6(2): 181-214.

[36] Shazeer N, Mirhoseini A, Maziarz K, et al. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer[C/OL]//Proceedings of the 5th International Conference on Learning Representations, 2017: 878-896. https://dblp.uni-trier.de/db/conf/iclr/iclr2017.html#conf/iclr/ShazeerMMDLHD17.

[37] Cai Weilin, Jiang Juyong, Wang Fan, et al. A survey on mixture of experts in large language models[J]. IEEE Transactions on Knowledge and Data Engineering, 2025, 37(7): 3896-3915.

[38] Riquelme C, Puigcerver J, Mustafa B, et al. Scaling vision with sparse mixture of experts[C]//Proceedings of the 35th International Conference on Neural Information Processing Systems. New York: ACM, 2021: 657.

[39] Kudugunta S, Huang Yanping, Bapna A, et al. Beyond distillation: Task-level mixture-of-experts for efficient inference[C]//Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2021. Stroudsburg: ACL, 2021: 3577-3599.

[40] Lin Junyang, Rui Men, Yang An, et al. M6: A Chinese multimodal pretrainer[PP/OL]. V4. arXiv (2021-05-29) [2025-11-13]. https://arxiv.org/abs/2103.00823v4.

[41] Fedus W, Dean J, Zoph B. A review of sparse expert models in deep learning[PP/OL]. V1. arXiv (2022-09-04) [2025-11-13]. https://arxiv.org/abs/2209.01667v1.

[42] Wang Wenhai, Xie Enze, Li Xiang, et al. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway: IEEE, 2021: 548-558.

[43] Liu Ze, Lin Yutong, Cao Yue, et al. Swin transformer: Hierarchical vision transformer using shifted windows[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway: IEEE, 2021: 9992-10002.

[44] Alwhbi I A, Zou C C, Alharbi R N. Encrypted network traffic analysis and classification utilizing machine learning[J]. Sensors, 2024, 24(11): 3509.

[45] Devlin J, Chang Mingwei, Lee K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding[C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg: ACL, 2019: 4171-4186.

[46] Ucar I, Morato D, Magaña E, et al. Duplicate detection methodology for IP network traffic analysis[C]//Proceedings of the 2013 IEEE International Workshop on Measurements & Networking (M&N). Piscataway: IEEE, 2013: 161-166.

[47] Ali Ali A, Zimerman I, Wolf L. The hidden attention of mamba models[C]//Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics. Stroudsburg: ACL, 2025: 1516-1534.

[48] Draper-Gil G, Lashkari A H, Mamun M S I, et al. Characterization of encrypted and VPN traffic using time-related features[C]//Proceedings of the 2nd International Conference on Information Systems Security and Privacy. Setúbal: SciTePress, 2016: 407-414.

[49] Habibi Lashkari A, Draper Gil G, Mamun M S I, et al. Characterization of tor traffic using time based features [C]//International Conference on Information Systems Security and Privacy. Setúbal: SciTePress, 2017, 2: 253-262.

[50] Wang Wei, Zhu Ming, Zeng Xuewen, et al. Malware traffic classification using convolutional neural network for representation learning[C]//2017 International Conference on Information Networking (ICOIN). Piscataway: IEEE, 2017: 712-717.

[51] Dadkhah S, Mahdikhani H, Danso P K, et al. Towards the development of a realistic multidimensional IoT profiling dataset[C]//Proceedings of the 2022 19th Annual International Conference on Privacy, Security & Trust (PST). Piscataway: IEEE, 2022: 9851966.

[52] Liu Liming, Li Ruoyu, Li Qing, et al. FlowletFormer: Network behavioral semantic aware pre-training model for traffic classification[PP/OL]. V1. arXiv (2025-08-27) [2025-11-13]. https://arxiv.org/abs/2508.19924v1.

[53] Park J T, Choi Y S, Cho B S, et al. Multi-level pre-training for encrypted network traffic classification[J]. IEEE Access, 2025, 13: 68643-68659.

[54] Fedus W, Zoph B, Shazeer N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity[J]. The Journal of Machine Learning Research, 2022, 23(1): 120.

作者简介

张 磊 男,1987年10月出生于江苏省徐州市。现为重庆大学微电子与通信工程学院教授、博士生导师。获重庆市自然科学奖、吴文俊人工智能自然科学奖等奖项6项。在国内外发表学术论文150余篇。中国电子学会会员编号:E190072830S。

E-mail: leizhang@cqu.edu.cn

刘 晓 男,2001年5月出生于甘肃省庆阳市。现为重庆大学微电子与通信工程学院硕士研究生。主要研究方向为深度学习、计算机视觉、网络流量分类。

E-mail: liuxiao@stu.cqu.edu.cn

张龙港 男,1999年10月出生于重庆市。现为重庆大学微电子与通信工程学院硕士研究生。主要研究方向为深度学习、计算机视觉、网络流量分类。

E-mail: zlg502361@gmail.com

何 清 男,2000年3月出生于云南省丽江市。现为重庆大学微电子与通信工程学院硕士研究生。主要研究方向为深度学习、计算机视觉、网络流量分类。

E-mail: qinghe@cqu.edu.cn