基于混合专家的流量分类基础模型

周嘉俊[1,2,3],孙长辉[1,2],何美静[1,2],俞山青[1,2,3*]
(1. 浙江工业大学网络空间安全研究院,浙江杭州 310023;2. 杭州市滨江区浙工大人工智能创新研究院,浙江杭州 310056;
3. 杭州数宇智汇科技发展有限责任公司,浙江杭州 310056)

摘 要: 随着网络通信技术的快速演进,攻击者广泛利用流量加密技术来隐匿恶意行为,导致基于端口匹配与深度包检测(Deep Packet Inspection,DPI)的传统流量分析技术的性能显著下降,网络安全防御边界日益模糊。尽管近年来基于深度学习与预训练技术的流量分类方法在提取网络流量深层特征方面取得了显著进展,但现有研究多采用密集型Transformer架构,模型推理时需激活全部参数,导致计算成本与模型规模紧密耦合,引发高昂的推理延迟与显存开销。在面对现代网络环境高吞吐量与实时检测的需求时,这种计算效率瓶颈极易形成防御漏洞,严重制约了大规模深度学习模型在实际网络防御场景中的部署与应用。为有效解决模型容量扩展与推理效率之间的矛盾,本文提出了一种专为异构流量分类设计的稀疏基础模型Traffic-MoE。该模型不仅沿用了"预训练-微调"范式以应对安全领域标注数据稀缺的挑战,更创新性地引入稀疏混合专家(Mixture-of-Experts,MoE)架构,实现对通用协议特征与特定领域行为的解耦建模。具体而言,本文首先设计了Traffic2Token异构流量表征方法,针对原始流量跨协议、跨设备的复杂特性,通过融合数据包关键特征与有效载荷,利用二元语法(bigram)分词技术构建细粒度Token序列,在保留字节级时序依赖关系的同时,有效抑制了底层噪声干扰。在此基础上,本文在Transformer架构中嵌入稀疏MoE模块以取代传统的密集前馈网络(Feed-Forward Network,FFN),利用可学习的门控网络实施动态路由策略,对于每个输入流量Token仅激活前k个最相关的特化专家,并保留共享专家以捕获通用的协议语法,从而在大幅扩展模型总容量的同时显著降低推理开销。依托自主构建的包含200万条会话流的无标签预训练语料库,模型通过自回归的"下一Token预测"任务习得网络协议的状态转换逻辑,随后仅需轻量级的监督微调便能快速适配下游任务。为了全面评估模型性能,本文在四个权威公开数据集上构建了六个典型的下游分类任务,涵盖物联网攻击检测、加密服务识别、匿名流量分析等多类场景。实验结果表明,相较于ET-BERT、NetGPT等现有先进基线方法,Traffic-MoE展现出更卓越的泛化能力与鲁棒性,整体检测性能平均提高了8.44%。更关键的是,得益于稀疏激活机制带来的计算优势,在同等参数规模下,Traffic-MoE相较于传统密集型架构实现了37.45%的吞吐量提升、27.25%的推理延迟缩减以及27.04%的GPU峰值显存消耗降低,为高效网络流量分析建立了一种新的范式。

关键词: 流量分类;基础模型;自监督预训练;混合专家;网络安全;流量表征

基金项目: 国家自然科学基金(No.62503423);国家重点研发计划(No.2025YFA1510900);浙江省自然科学基金联合重点(No.LBMHZ25F020002)

中图分类号: TP393.08 **文献标识码:** A **文章编号:** 0372-2112(2026)04-1460-21

电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20251026

A Foundation Model for Traffic Classification Based on Mixture-of-Experts

ZHOU Jiajun[1,2,3], SUN Changhui[1,2], HE Meijing[1,2], YU Shanqing[1,2,3*]
(1. *Institute of Cyberspace Security*, *Zhejiang University of Technology*, *Hangzhou*, *Zhejiang* 310023, *China*;
2. *Zhejiang University of Technology Artificial Intelligence Innovation Institute*, *Binjiang District*, *Hangzhou*, *Zhejiang* 310056, *China*;
3. *Soovar Technologies Co.*, *Ltd.*, *Hangzhou*, *Zhejiang* 310056, *China*)

Abstract: With the rapid evolution of network communication technologies, adversaries extensively leverage traffic encryption techniques to conceal malicious behaviors. Consequently, the performance of traditional traffic analysis methods based on port matching and deep packet inspection (DPI) has declined significantly, rendering network security defense boundaries increasingly blurred. Although traffic classification methods based on deep learning and pre-training techniques have made significant progress in extracting deep features of network traffic, existing studies predominantly adopt dense Transformer architectures. These models necessitate the activation of all parameters during inference, resulting in a tight coupling between computational cost and model scale, thereby incurring high inference latency and memory overhead. In

收稿日期:2026-01-13;录用日期:2026-02-05;责任编委:沈蒙
*通信作者:俞山青

the face of demands for high throughput and real-time detection in modern network environments, this computational efficiency bottleneck tends to create critical defense vulnerabilities, severely constraining the deployment and application of large-scale deep learning models in practical network defense scenarios. To effectively resolve the contradiction between model capacity expansion and inference efficiency, this paper proposes Traffic-MoE, a sparse foundation model designed specifically for heterogeneous traffic classification. This model not only follows the "pre-training and fine-tuning" paradigm to address the challenge of labeled data scarcity in the security domain but also innovatively introduces a sparse mixture-of-experts (MoE) architecture to achieve decoupled modeling of general protocol features and domain-specific behaviors. Specifically, we first design the Traffic2Token heterogeneous traffic representation method. Addressing the complex cross-protocol and cross-device characteristics of raw traffic, this method integrates critical packet features with payloads and utilizes bigram tokenization to construct fine-grained token sequences, effectively suppressing underlying noise interference while preserving byte-level temporal dependencies. On this basis, we embed sparse MoE modules into the Transformer architecture to replace traditional dense feed-forward networks (FFN). By leveraging a learnable gating network to implement a dynamic routing strategy, the model activates only the top-k most relevant specialized experts for each input traffic token while retaining a shared expert to capture general protocol syntax, thereby significantly reducing inference overhead while substantially expanding total model capacity. Leveraging a self-constructed unlabeled pre-training corpus containing 2 million session flows, the model learns the state transition logic of network protocols through an autoregressive "next-token prediction" task, subsequently requiring only lightweight supervised fine-tuning to rapidly adapt to downstream tasks. To comprehensively evaluate model performance, we construct six typical downstream classification tasks across four authoritative public datasets, covering diverse scenarios such as IoT (Internet of Things) attack detection, encrypted service identification, and anonymous traffic analysis. Experimental results demonstrate that compared to existing state-of-the-art baselines such as ET-BERT and NetGPT, Traffic-MoE exhibits superior generalization ability and robustness, with an average improvement of 8.44% in overall detection performance. Crucially, benefiting from the computational advantages brought by the sparse activation mechanism, Traffic-MoE achieves a 37.45% increase in throughput, a 27.25% reduction in inference latency, and a 27.04% decrease in peak GPU memory consumption compared to traditional dense architectures with equivalent parameter scales, establishing a new paradigm for efficient network traffic analysis.

Keywords: traffic classification; foundation model; self-supervised pre-training; mixture-of-experts; network security; traffic representation

Foundation Item(s): National Natural Science Foundation of China (No.62503423); National Key Research and Development Program of China (No.2025YFA1510900); Baima Lake Laboratory Joint Fund of Zhejiang Provincial Natural Science Foundation of China (No.LBMHZ25F020002)

0 引言

随着密态网络时代的到来,异构设备生态系统的形成,以及物联网(Internet of Things,IoT)和Web3等新兴范式的出现,网络通信环境发生了深刻变革[1-2]。当前,基于传输层安全性协议(Transport Layer Security,TLS)的服务、设备到云的遥测以及去中心化协议主导着互联网流量,使网络数据呈现显著的"黑盒"特性,传统的包检测方法难以穿透加密层识别应用程序语义。虽然这些进步提升了隐私性和可用性,但也带来了严峻的安全挑战:僵尸网络通信、凭证滥用等恶意活动可以深度隐匿在加密流量中,导致其在数据包层面与良性服务难以区分。

流量分类是密态网络防御的基础,支持入侵检测[3]、恶意软件分析[4]、高级持续性威胁(advanced persistent threat)检测[5]等任务。然而,传统方法已经难以适应当前日益复杂的异构网络环境。早期基于端口或静态规则的方法,已经因动态端口分配和协议伪装技术的普及而失效[6]。后续基于机器学习的方法[7-8],其性能高度依赖于人工设计的统计特征或协议签名,难以捕捉流量中蕴含的深层语义,在面对加密、混淆等场景时出现严重的性能下滑。为了克服这些局限性,最近的研究利用深度学习模型直接从原始数据中自动提取有效特征,从而在异构和加密流量分类中实现了更高的准确率[9-11]。但这些方法仍然面临一个核心瓶颈:对大规模、高质量标注数据的严重依赖,标注数据不足会引起模型性能的显著退化。而网络流量的标注极为复杂且成本高昂,并且在零日攻击、恶意软件变体等关键对象上存在典型的长尾分布现象[12],极大地限制了模型的实际部署效果。为突破数据标注瓶颈,研究界开始借鉴计算机视觉(computer vision)与自然语言处理(natural language processing)领域的成功经验[13],引入"预训练-微调"范式,构建更先进的预训练流量分类模型,以进一步提升在下游任务中的性能[14-18]。

尽管取得了这些进展,但大多数现有模型仍采用密集型(dense)架构,在推理过程中需激活全部参数,导致计算成本随模型规模的扩大而线性增加。随着模型容量的提升,推理延迟显著增加,不可避免地造成检测盲区,成为实时密态网络流量分析的关键瓶颈。虽然大型预训练模型能够捕捉丰富的协议语义,但其密集架构迫使网络管理员在检测深度和系统可用性之间做出权衡。此类密集模型在离线基准测试中可能表现良好,但难以适配高吞吐量、低延迟的实际网络防御场景。因此,实用的网络防御系统亟须兼具强大语义表征能力与高效推理性能的模型。

本文通过重构大规模流量分类模型的计算资源分配逻辑,解决安全性和检测效率之间的矛盾。研究发现,网络流量行为具有结构性异构特征而非均匀分布的,恶意软件信标、虚拟专用网络(Virtual Private Network,VPN)、Web3中继流量均表现出独特的字节级模式与时间动态特性。均匀密集架构需要低效地调动整个网络去处理所有流量模式,而混合专家架构(Mixture of Experts,MoE)[19]可以动态地仅激活与当前处理的流量域最相关的参数子空间,实现高效推理。这引出了本文的核心见解:网络流量建模的效率并非一定源于模型规模的缩减,还可以通过在推理过程中选择性地激活模型容量实现[20-22]。

针对现有问题,本文将预训练策略与混合专家架构深度融合,提出一种适用于通用网络流量分类的稀疏基础模型——Traffic-MoE。该模型不仅继承了预训练方法从海量无标签数据中学习通用知识的优势,还通过稀疏激活机制将总体检测能力与计算开销解耦,提升了模型的计算效率与灵活性。这种设计使得模型容量可大幅扩展,且不会成比例地增加推理成本,从而使大规模预训练流量分类模型在延迟敏感场景中的部署成为可能。为了适配异构网络流量,本文设计了一种新颖的流量表示方法Traffic2Token,将协议元数据与选择性有效载荷段相结合,使模型能更全面地捕获多层次信息。Traffic-MoE通过自监督方法在大规模无标签流量语料库上进行预训练,以捕获通用的协议语法和时序依赖关系。在下游自适应过程中,Traffic-MoE可以针对入侵检测、服务分类、加密流量分析等安全任务进行微调,同时保持高效推理,兼顾基准测试与实际部署需求。

本文的主要贡献总结如下:(1)提出一种基于混合专家的网络流量分类基础模型,该模型是首个专为流量建模设计的MoE架构,可在保持高效推理的同时实现大模型容量,弥补了当前密集型预训练模型的效率缺陷;(2)设计了一种支持跨流量语义专家特化的统一学习框架,本文通过Traffic2Token和均衡稀疏路由适配异构网络流,使专家自主专精于不同流量行为,提高面向密态网络安全的泛化能力;(3)通过广泛的评估验证了模型的优越性和高效率。在六个典型的下游流量分类任务中,Traffic-MoE的分类性能、少样本学习能力均显著超越了现有方法,同时保持了卓越的推理效率。

1 相关工作

1.1 统计机器学习方法

早期的流量分类方法主要依赖深度包检测(Deep Packet Inspection, DPI)技术,通过匹配数据包头或载荷中的预定义签名来识别应用程序[23]。然而,DPI技术不仅计算开销巨大,难以应对爆发式增长的网络流量规模,更因现代网络传输中普遍采用端到端加密和协议混淆技术,其分析能力受到严重限制。

为了克服流量加密导致的有效载荷不可见问题,基于统计特征的机器学习方法转而从流量中提取人工预先设计的时序或结构特征进行分类,此类模型通常结构轻量、参数规模较小。例如,Moore等人[24]提出一种基于内容特征的分类方法,降低了传统技术对端口号的依赖。Saber等人[25]则结合主成分分析(principal component analysis)与支持向量机(support vector machine),重点挖掘基于时间的统计模式进行分类。AppScanner[7]则从加密流量中提取54个统计特征,并训练随机森林(random forest)分类器来识别移动应用程序。此外,FlowPrint[8]通过聚类算法捕捉通信目的地之间的时序关联,生成特定于应用程序的流量指纹,从而实现对未知应用的识别。尽管这些方法计算效率较高,但其检测性能受限于特征工程的质量,难以捕捉流量数据中深层次的语义信息,在高动态加密或对抗性流量环境中表现不佳。

1.2 深度学习方法

随着深度学习技术的兴起,端到端(end-to-end)学习范式已被广泛用于从原始流量数据中自动提取高维特征,从而摆脱了对手工特征的依赖。这类方法根据流量数据的不同特性,采用了多样化的数据表征方式和模型架构来处理流量。其模型复杂度相较于传统机器学习方法有所提升,通常属于为特定任务设计的、参数量适中的专用模型。在序列建模方面,FS-Net[9]采用基于双向门控循环单元(Bidirectional Gated Recurrent Unit,Bi-GRU)的编码器-解码器架构(encoder-decoder architecture),直接从加密流量中学习特征。在视觉化表征方面,ATVITSC[26]将流量的有效载荷转换为伪图像,并结合视觉Transformer与卷积-长短期记忆网络(convolutional long short-term memory)的混合

架构,联合捕获流量的全局与时空特征。在图结构建模方面,部分工作进一步将流量视为结构化的关系数据。其中,TFE-GNN[11]将数据包字节序列建模为图结构,并使用图神经网络(Graph Neural Network,GNN)分别对包头和有效载荷进行编码。而Yang等人[27-28]引入超图(hypergraph)结构,将独立的流视作图节点,并通过K近邻(K-nearest keighbors)算法生成超边(hyperedge)以反映流之间的高阶相关性。这些方法展示了深度神经网络在提取复杂语义结构方面的优势,超越了传统的手工特征方法。但它们仍然严重依赖于大规模标注数据集,即使是最先进的深度学习模型,在面对标注数据稀缺的实际场景时,性能也会出现显著下降。

1.3　大规模预训练方法

近年来,研究者将预训练范式引入流量分析领域,旨在构建具备更强表征能力的流量分类模型。诸如PERT[14]、ET-BERT[15]、NetGPT[16]、YaTC[17]和TrafficFormer[18]等工作,均针对流量特性设计了特定的预训练任务和模型架构,在大规模流量语料库上通过自监督学习获取可泛化的特征表示,进而通过微调适配下游任务。这些方法显著提高了模型在加密和有限监督条件下的性能,展现了"预训练-微调"范式的巨大潜力。然而,现有的主流预训练模型多采用密集型Transformer架构,在推理过程中需要激活全部参数用于处理不同的流量模式。正如已有研究所指出的,在密集架构的设计下,模型容量和推理成本紧密耦合,模型规模的扩大不可避免地导致推理延迟和资源开销的显著增加。这种计算特性大幅削弱了模型的实时检测能力,严重限制了其在密态网络防御场景中的实际部署。

2　本文方法

网络流量呈现出显著的异构行为模式,这要求模型具备差异化建模能力,而非依赖统一的密集计算范式。同时,面对海量数据,现实的网络监控系统对低延迟处理有着明确需求,以避免因处理滞后导致丢包和检测盲区,这使得计算成本高昂的大规模密集模型难以落地。这些挑战揭示了模型部署的核心矛盾:网络流量的深度语义建模日益有效,但在实际运行环境中却难以高效实现。

为了解决这些限制,本文设计了Traffic-MoE作为网络流量分类的基础模型,以兼顾语义表达能力和计算效率。Traffic-MoE并非通过缩减模型容量来换取计算速度,而是针对每个输入的流量Token仅激活最相关的模型参数子集,从而在不显著增加相应推理开销的前提下实现可扩展性。这种稀疏激活策略使计算路径能够动态适应特定的流量语义,实现跨异构流的领域专业化。图1展示了模型的整体工作流程。具体而言,Traffic-MoE由四个组件构成:(1)Traffic2Token将原始流量转换为结构化的Token序列,以保留关键元数据和结构模式;(2)多层Transformer骨干网络对数据包和会话流之间的时间依赖关系进行建模,以获得上下文相关的特征嵌入;(3)稀疏MoE层取代了密集的前馈网络(Feed-Forward Network,FFN)层,通过门控机制根据流量Token的语义仅激活最相关的k个专家,同时将共享专家与领域特化专家相结合;(4)预训练-微调流水线支持大规模自监督学习和下游任务自适应,可适配服务分类、恶意流量检测等多种场景。这些组件共同构成了一个统一架构,在保持深度语义建模能力的同时,确保模型在吞吐量敏感型环境中的高效性。

2.1　流量表示方法

从真实网络中捕获的原始流量包含跨应用、协议和设备的异构通信模式。直接对原始会话流进行建模通常效果不佳,因为完整的数据包头部包含大量噪声,而加密的有效载荷又限制了语义的可见性。因此,本文设计了一个Traffic2Token模块,将原始流量转换为结构化的Token序列,在保留字节级行为信号的同时,摆脱了对特定协议先验假设的依赖。

2.1.1　基于数据包的流序列化

本文首先通过五元组标识符(源IP地址、目的IP地址、源端口号、目的端口号和传输协议)从数据包追踪数据中重组会话级流,并移除内部数据包过少的短流,以减少统计方差并确保模型学习到可靠的行为模式。由此产生的会话流构成了Traffic2Token的基本处理单元。

为了避免引入随机初始化或与传输无关的数据包头部字段所带来的噪声及模型偏见,本文将每个数据包都转换为紧凑且稳定的表示形式。具体而言,本文提取了六个反映传输行为和时间动态的包级属性:数据包长度、传输方向、TCP标志位、包间隔时间、IP协议类型和有效载荷长度,如图2所示。这些属性被序列化为固定长度的字节序列。同时,本文对数据包有效载荷部分进行截断采样,以控制流序列长度,同时保留具有代表性的内容模式。随后,将元数据字节和采样的有效载荷字节拼接,形成包级十六进制序列。对于每条会话流,本文选择前K个数据包作为模型输入,并将这些数据包的十六进制表示按时间顺序拼接,构建出流级十六进制序列。这种基于数据包的序列化策略有效保留了通信的方向性、突发性和时序结构,而这些信息在纯字节流中通常容易丢失。

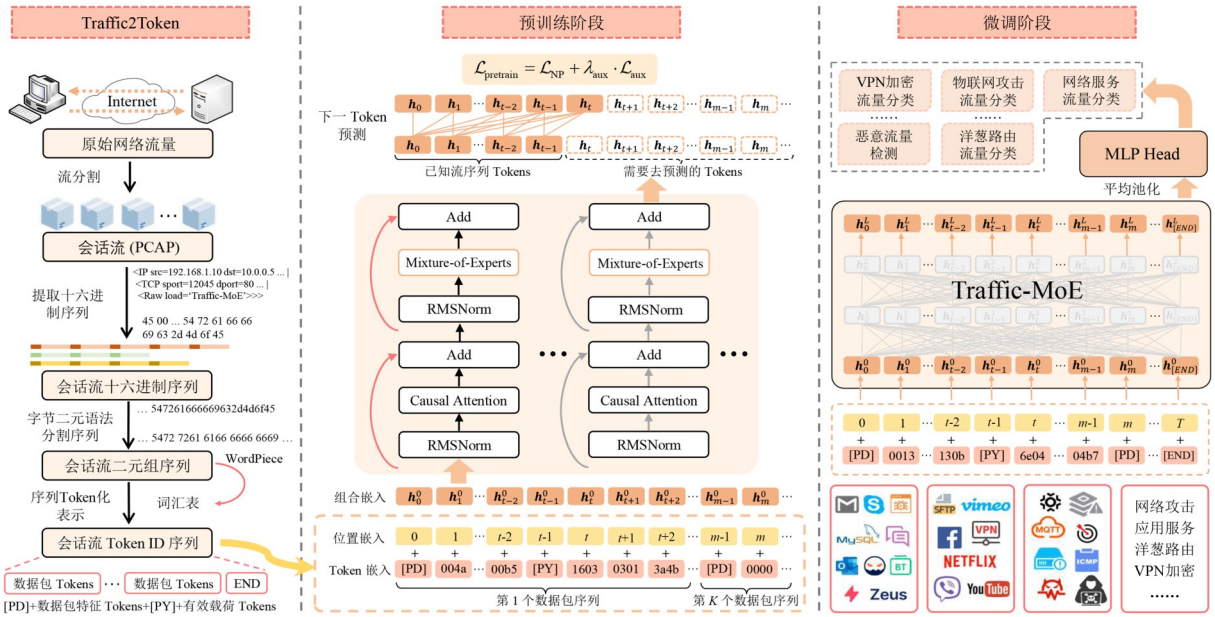

图 1 Traffic-MoE 整体框架

Figure 1 The overall framework of Traffic-MoE

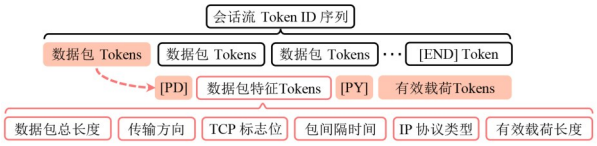

图 2 会话流 Token 序列的结构

Figure 2 Structure of session flow token sequences

2.1.2 分词与嵌入

网络协议通常表现出一些局部结构规律，例如协议特定的头部字段、帧模式和数据边界。为了揭示这些结构关联性，本文采用“二元语法”(bigram)的策略，通过对流级十六进制序列应用滑动窗口，将相邻的字节对转换为字节二元语法。这种二元语法转换生成了一种细粒度的流级序列表示，突出了局部结构线索，且无需保留完整的数据包信息。随后，本文将所有流的二元语法序列视为一个统一的语料库，采用 WordPiece[29] 算法构建一个亚字节级词汇表 $\mathcal{V}$。WordPiece 算法可自适应地识别具有统计显著性的二元语法组合(例如 TLS 记录头部)，并将它们标记为 Token 单元。这种数据驱动的词汇表构建方式能够有效捕获重复出现的流量模式，同时避免词汇表规模过度膨胀，为后续模型高效学习奠定基础。

此外，为了增强模型对会话流层级结构的感知，本文在流序列中还引入了具有特殊功能的结构化标记：[PD]用于标记每个数据包序列的开始；[PY]用于分隔数据包序列中的元数据字段和有效载荷字节；[PAD]用于序列填充以确保输入长度一致；[END]用于标记整条会话流序列的结束；[UNK]表示在词汇表之外的低频字节模式。这些结构标记确保模型在处理长序列时能够清晰地识别数据包边界，从而防止扁平化字节流中流信息的语义坍塌。最终，流级序列中的二元组 Token 和特殊标记通过学习到的词汇表映射到 Token ID。每个 Token ID 都被嵌入到一个稠密向量中，并引入位置编码，通过对 Token ID 嵌入和位置嵌入求和，可以生成流级 Token 序列表示：$\boldsymbol{X}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_T)$，其中 $\boldsymbol{x}_t\in\boldsymbol{R}^d$ 是位置 t 处的初始 Token 嵌入，T 是序列长度，d 是嵌入维度。这种统一的嵌入表示既支持局部字节级模式建模，也支持全局数据包级时序推理，从而实现鲁棒的训练和稀疏专家路由。

2.2 具有稀疏专家层的骨干网络

图 3 展示了 Traffic-MoE 的整体骨干网架构，其中深度集成了因果掩码自注意力机制(causal self-attention)[30]和稀疏 MoE 前馈层。这种设计源于网络流量的双重特性：一方面，数据包序列展现出由协议状态机驱动的时序依赖性；另一方面，流量中包含由各种应用和攻击者产生的、高度异构且特定于领域的行为模式。为了在一个统一的模型中同时捕获这些底层序列结构和高层行为变化，模型骨干网由多个堆叠的自回归 Transformer 层组成。其中，每个模块首先执行因果自注意力机制来建模流量 Token 的有序演化，随后应用稀疏激活的专家层，动态调整计算路径以适配当前的流量语义。由此产生的混合架构成功将模型表征能力与推理成本解耦，使 Traffic-MoE 能够在保持实时推理效率的前提下，扩展到大规模的参数量级。

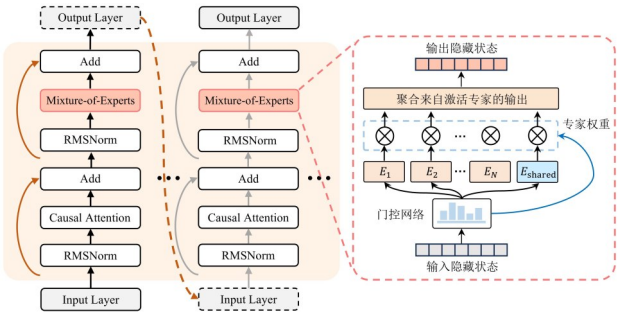

图3　模型骨干网架构

Figure 3　Architecture of the model backbone

2.2.1　因果掩码自注意力机制

网络流量的演化遵循严格的协议状态机逻辑,例如握手交换、隧道封装、重传等。这些协议阶段在字节级Token序列中表现为时序依赖结构。为了满足现实世界网络安全监控的约束,即模型检测时无法获取未到达数据包的信息,同时有效建模这些时序依赖关系,Traffic-MoE采用了一种因果掩码多头自注意力机制。

给定第$(l-1)$层中流序列的隐藏状态$\boldsymbol{H}^{(l-1)}\in\boldsymbol{R}^{T\times d}$,模型首先应用均方根层归一化(Root Mean Square Layer Normalization,RMSNorm)[31]来消除异构流段之间的特征尺度差异,从而提高训练稳定性。随后,为了准确编码数据包之间的时间偏移,例如突发间隔或时序偏移,本文将旋转位置嵌入(Rotary Positional Embeddings,RoPE)[32]融入注意力机制的计算中。与绝对位置编码(absolute positional encoding)不同,RoPE通过在共享频率空间中旋转特征向量,以数学方式注入相对位置信息,能够更好地应对变长序列。第j个注意力头的查询矩阵$\boldsymbol{Q}$、键矩阵$\boldsymbol{K}$和值矩阵$\boldsymbol{V}$的计算如下:

$$\boldsymbol{Q}_j^{(l)}=\mathrm{RoPE}\left(\mathrm{RMSNorm}\left(\boldsymbol{H}^{(l-1)}\right)\cdot\boldsymbol{W}_{Q,j}^{(l)}\right)\tag{1}$$

$$\boldsymbol{K}_j^{(l)}=\mathrm{RoPE}\left(\mathrm{RMSNorm}\left(\boldsymbol{H}^{(l-1)}\right)\cdot\boldsymbol{W}_{K,j}^{(l)}\right)\tag{2}$$

$$\boldsymbol{V}_j^{(l)}=\mathrm{RMSNorm}\left(\boldsymbol{H}^{(l-1)}\right)\cdot\boldsymbol{W}_{V,j}^{(l)}\tag{3}$$

其中,$\boldsymbol{W}_{\{Q,K,V\},j}^{(l)}\in\boldsymbol{R}^{d\times d_m}$是可学习的投影矩阵。这种计算方式使得注意力机制能够捕捉流Token之间的相对序列距离,这对于识别与绝对序列位置无关的流量模式(例如,可能出现在流序列中任意位置的攻击特征)至关重要。为了遵循网络流量的物理约束,即未到达的数据包不能影响当前流量状态的表征,本文引入了一个严格的下三角因果掩码$\boldsymbol{M}\in\boldsymbol{R}^{T\times T}$。注意力输出的计算如下所示:

$$\boldsymbol{O}_j^{(l)}=\mathrm{Softmax}\left(\frac{\boldsymbol{Q}_j^{(l)}\left(\boldsymbol{K}_j^{(l)}\right)^{\mathrm{T}}}{\sqrt{d_m}}+\boldsymbol{M}\right)\cdot\boldsymbol{V}_j^{(l)}\tag{4}$$

$$\boldsymbol{M}_{tp}=\begin{cases}0, & p\leqslant t\\ -\infty, & p>t\end{cases}\tag{5}$$

此因果掩码确保处于位置t的Token只能关注自身及其前面位置的信息。最后,将所有m个注意力头的输出拼接并通过一个输出投影矩阵进行融合,形成残差更新:

$$\tilde{\boldsymbol{H}}^{(l)}=\boldsymbol{H}^{(l-1)}+\left[\boldsymbol{O}_1^{(l)}\|\boldsymbol{O}_2^{(l)}\|\cdots\|\boldsymbol{O}_m^{(l)}\right]\cdot\boldsymbol{W}_O^{(l)}\tag{6}$$

其中,$\boldsymbol{W}_O^{(l)}\in\boldsymbol{R}^{d\times d}$是输出投影矩阵,$\tilde{\boldsymbol{H}}^{(l)}$是中间隐藏状态。该设计确保模型能够通过自回归任务有效学习底层流量语法,从而对网络协议的状态转换逻辑进行精准建模。

2.2.2　带有稀疏路由的混合专家层

在经过因果掩码自注意力模块处理后,中间隐藏状态$\tilde{\boldsymbol{H}}^{(l)}$将输入MoE层。标准Transformer通常采用密集型FFN,对所有输入数据应用同一组参数进行处理。然而,网络流量同时包含通用的协议规则(如传输头结构、标准握手模式)和高度异构的行为模式(如特定应用程序的流量指纹、恶意软件的命令与控制流),使用统一参数空间去处理这两种截然不同的模式并不理想。为了在适应这种异构性的同时避免计算成本的线性增长,Traffic-MoE使用由“共享专家”和多个“特化专家”组成的混合MoE层替换了传统的密集FFN层。

为了实现通用协议知识与特定流量行为的解耦,本文设计了一种双路径计算机制。首先,对输入的$\tilde{\boldsymbol{H}}^{(l)}$进行RMSNorm归一化,得到$\boldsymbol{Z}^{(l)}=\mathrm{RMSNorm}\left(\tilde{\boldsymbol{H}}^{(l)}\right)$。随后,将经过归一化后的隐状态路由到两个不同的专家组并行处理。

(1)用于捕捉通用协议语义的共享专家。为了捕获在各种流量场景中保持不变的基础结构规律,本文设计了一个持续激活的共享专家E_{shared}。与稀疏特化专家间的竞争性路由不同,共享专家采用独立的门控机制,自适应地调节通用流量知识的注入强度:

$$\boldsymbol{O}_{\text{shared}}^{(l)}=\mathrm{Sigmoid}\left(\boldsymbol{Z}^{(l)}\boldsymbol{w}_{\Delta}\right)\odot E_{\text{shared}}\left(\boldsymbol{Z}^{(l)}\right)\tag{7}$$

其中,$\boldsymbol{w}_{\Delta}\in\boldsymbol{R}^{d}$是投影向量,$\odot$表示在特征维度上进行的逐元素乘法运算。

(2)用于处理异构流量模式的特化专家。同时,本文构建了一个包含N个专家的专家库$\{E_e\}_{e=1}^{N}$,旨在专门处理多样化的流量模式。路由网络计算这些专家的激活概率分布,并选择前k个最相关的专家来处理当前输入Token。路由得分$\boldsymbol{S}^{(l)}$和稀疏路由权重$\tilde{\boldsymbol{S}}^{(l)}$的计算如下:

$$\boldsymbol{S}^{(l)}=\mathrm{Softmax}\left(\boldsymbol{Z}^{(l)}\boldsymbol{W}_{*}\right)\tag{8}$$

$$\tilde{\boldsymbol{S}}_{ue}^{(l)}=\begin{cases}\boldsymbol{S}_{ue}^{(l)}, & e\in \mathrm{Top}_k\left(\boldsymbol{S}_u^{(l)}\right)\\ 0, & \text{otherwise}\end{cases} \tag{9}$$

其中，$\boldsymbol{W}_*\in\boldsymbol{R}^{d\times N}$将输入投影到专家嵌入空间，$\boldsymbol{S}^{(l)}\in\boldsymbol{R}^{T\times N}$包含专家对每个Token的Logits，而$\tilde{\boldsymbol{S}}^{(l)}\in\boldsymbol{R}^{T\times N}$是稀疏路由矩阵，其中对于每个流量Token，模型仅保留前k个最大的路由得分，而其余专家的权重被设置为0。这种稀疏路由机制确保了对于每个输入的流量Token，模型仅激活与其语义最匹配的专家子集。稀疏分支的总输出是所有被激活专家输出的加权和：

$$\boldsymbol{O}_{\text{special}}^{(l)}=\sum_{e=1}^{N}\tilde{\boldsymbol{S}}_{:,e}^{(l)}\odot E_e\left(\boldsymbol{Z}^{(l)}\right) \tag{10}$$

其中，$\tilde{\boldsymbol{S}}_{:,e}^{(l)}$表示专家$E_e$在所有Token上的路由权重，乘法$\odot$沿特征维度进行广播。通过设置$k\ll N$，模型能够选择性地激活与当前正在处理的特定攻击或协议阶段最相关的参数子空间。

为了构建高效的专家系统，共享专家和特化专家均采用基于SwiGLU的门控前馈设计。给定输入$\boldsymbol{Z}$，由参数$\left\{\boldsymbol{W}_{\text{gate}},\boldsymbol{W}_{\text{up}},\boldsymbol{W}_{\text{down}}\right\}$定义的专家网络计算流程如下：

$$E(\boldsymbol{Z})=\left(\mathrm{SiLU}\left(\boldsymbol{Z}\boldsymbol{W}_{\text{gate}}\right)\odot\left(\boldsymbol{Z}\boldsymbol{W}_{\text{up}}\right)\right)\boldsymbol{W}_{\text{down}} \tag{11}$$

其中，SiLU(·)是Sigmoid线性单元。值得注意的是，共享专家保留了完整的中间维度$d'\gg d$以充分捕获复杂的通用协议模式（即$\boldsymbol{W}_{\{\text{gate, up}\}}\in\boldsymbol{R}^{d\times d'}$和$\boldsymbol{W}_{\text{down}}\in\boldsymbol{R}^{d'\times d}$）；而稀疏特化专家则使用更小的中间维度$d'/k$以平衡专家数量和参数规模（即$\boldsymbol{W}_{\{\text{gate, up}\}}\in\boldsymbol{R}^{d\times\frac{d'}{k}}$和$\boldsymbol{W}_{\text{down}}\in\boldsymbol{R}^{\frac{d'}{k}\times d}$）。不同特化专家之间的权重相互独立。

MoE层的最终输出融合了来自共享专家的协议不变特征和来自特化专家的领域特定特征，并添加到残差流中，这部分计算过程表示如下：

$$\boldsymbol{H}^{(l)}=\tilde{\boldsymbol{H}}^{(l)}+\left(\boldsymbol{O}_{\text{shared}}^{(l)}+\boldsymbol{O}_{\text{special}}^{(l)}\right) \tag{12}$$

为了防止路由崩溃，即路由网络过度依赖少数特定专家而导致其他专家训练不足，本文引入了负载均衡辅助损失$\mathcal{L}_{\text{aux}}$。对于第l层，基于专家激活的统计数据计算损失。$\mathrm{Load}_e^{(l)}$表示当前批次（batch）中分配给专家e的Token比例，$\mathrm{Prob}_e^{(l)}$表示专家e在该批次中的平均路由概率，则第l层的负载均衡损失定义为

$$\mathcal{L}_{\text{aux}}^{(l)}=N\sum_{e=1}^{N}\mathrm{Load}_e^{(l)}\cdot\mathrm{Prob}_e^{(l)} \tag{13}$$

$$\mathcal{L}_{\text{aux}}=\frac{1}{L}\sum_{l=1}^{L}\mathcal{L}_{\text{aux}}^{(l)} \tag{14}$$

总辅助损失$\mathcal{L}_{\text{aux}}$是所有L层上负载均衡损失的平均值。该约束确保专家在批次级别上被均匀利用，最大化模型容量的有效利用率，同时保持稀疏推理的高效性。尽管MoE架构在训练阶段需要加载所有专家参数，使得显存开销和优化难度有所增加，但通过负载均衡机制和并行优化策略，模型有效规避了训练不稳定的风险。更重要的是，这种设计体现了网络安全场景中“计算资源置换”理念，即通过增加离线训练时的资源投入，换取在线部署阶段的极致推理速度与低延迟响应，从而完美契合实时防御的需求。

2.3 预训练和微调

为了赋予模型广泛的流量理解能力和对特定任务的判别能力，本文设计了两阶段训练框架：大规模自监督预训练和轻量级少样本（few-shot）微调。预训练阶段将骨干网络暴露于海量未标注的流量数据中，通过自回归式的下一Token预测任务和均衡的专家激活机制，使模型学习网络流量的通用特征表示。微调阶段则采用时序数据增强、流级语义池化和分层优化策略，将模型习得的通用表示适配到下游流量分类任务。两阶段协同作用，确保模型在各类流量安全场景下保持高效和鲁棒，同时便于扩展容量。

2.3.1 自回归预训练：下一Token预测

现实网络持续产生海量未标注数据，但现有的经过标注的网络入侵或应用程序数据集却十分有限。因此，利用这些丰富的未标注流量进行预训练，对于模型学习可泛化的流量表示至关重要。本文采用自监督的“下一Token预测”的目标函数来利用这些未标注数据。由于网络通信本质上是顺序的，且受协议驱动的状态转换逻辑控制，因此基于历史上下文预测后续Token能够使模型学习流量的时间依赖性和结构规律，这一特性在加密流中同样适用。该过程使模型实现对网络行为的统一理解，从而在下游任务微调过程中有效地进行参数调整。图1展示了完整预训练流程，其中Traffic2Token模块、因果注意力机制和稀疏专家模型协同工作，形成一个可扩展的流量分类基础模型。

给定一个流Token序列$\boldsymbol{X}=\left\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_T\right\}$，模型在任意位置$t$的目标是基于其历史上下文$\boldsymbol{X}_{<t}=\left\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_{t-1}\right\}$最大化真实Token$\boldsymbol{x}_t$的条件概率。具体而言，对于上下文序列$\boldsymbol{X}_{<t}$，Traffic-MoE的$L$层骨干网络对其进行编码，生成一组高维隐状态向量$\boldsymbol{H}_{<t}^{(L)}=\left\{\boldsymbol{h}_1^{(L)},\boldsymbol{h}_2^{(L)},\cdots,\boldsymbol{h}_{t-1}^{(L)}\right\}$。随后，利用聚合了当前位置完整上下文信息的隐状态$\boldsymbol{h}_{t-1}^{(L)}\in\boldsymbol{R}^d$，通过将其投影到词汇空间以获得下一个Token的预测概率分布：

$$p_\theta\left(\boldsymbol{x}_t|\boldsymbol{X}_{<t}\right)=\mathrm{Softmax}\left(\boldsymbol{h}_{t-1}^{(L)}\cdot\boldsymbol{W}_{\mathrm{vocab}}\right)\tag{15}$$

其中,$\boldsymbol{W}_{\mathrm{vocab}}\in\boldsymbol{R}^{d\times|\mathcal{V}|}$是一个投影矩阵,用于将隐藏状态映射到词汇Logits。为了优化模型参数θ,本文采用负对数似然损失函数$\mathcal{L}_{\mathrm{NP}}$。对于每个批次,该损失函数计算其中所有有效Token的平均损失:

$$\mathcal{L}_{\mathrm{NP}}=-\frac{1}{B\cdot T}\sum_{b=1}^{B}\sum_{t=1}^{T}\log p_\theta\left(\boldsymbol{x}_t^{(b)}|\boldsymbol{X}_{<t}^{(b)}\right)\tag{16}$$

其中,B表示该批次中包含的流序列数量。最后,由于MoE仅为每个Token激活一小部分专家,因此均衡的专家利用成为预训练的关键要求。本文引入负载均衡项$\mathcal{L}_{\mathrm{aux}}$来防止路由崩溃,并促进跨异构流量行为的专家特化(expert specialization)。因此,预训练阶段的总体目标是:

$$\mathcal{L}_{\mathrm{pretrain}}=\mathcal{L}_{\mathrm{NP}}+\lambda_{\mathrm{aux}}\cdot\mathcal{L}_{\mathrm{aux}}\tag{17}$$

其中,λ_{aux}控制负载均衡正则化的强度。通过最小化此复合损失,模型能够习得丰富的、协议感知的流量表示,同时鼓励高效且多样化的专家激活。因此,预训练骨干网既能捕捉通用的流量模式,又能初步捕捉特定领域的细微差别,为包括入侵检测、服务分类和加密流量分析在内的下游任务提供了坚实的基础。

2.3.2　少量数据微调:流量分类

下游流量分类任务,例如恶意软件识别、服务分类和攻击检测,通常面临标签稀缺和类别分布高度不平衡的问题。为了使模型有效适应这些数据受限的场景,本文设计了一种专为少样本学习设计的微调策略。该策略集成了时间感知数据增强、流级表示聚合和分层优化,从而在有限监督下实现稳健的性能。

网络会话的持续时间差异极大,域名系统(Domain Name System, DNS)查询可能在几毫秒内完成,而视频流或VPN隧道可能持续数十分钟。这种差异导致流长度呈典型的长尾分布,并且某些类别的样本数量极其有限。为了缓解这种不平衡现象,本文采用了基于时间切片的方法,将长会话流分割成多个固定持续时间的子流,这些子流继承了原始标签。由此产生的增强数据集提高了特定类别的样本数量,使模型能够捕捉到更精细的时间动态,例如周期性信标、空闲间隔和突发传输,而这些动态在使用完整会话流表示时往往会被稀释。这种增强方法最大限度地利用了稀疏的会话级样本,这在少样本场景中尤为有效。

对于监督自适应,本文将最终骨干网络层输出的Token表示聚合为统一的流级嵌入。本文不依赖于单个特殊Token表示(例如[PD]或[END]),因为这些Token嵌入可能会忽略会话中期的结构语义。相反,本文对所有有效Token表示应用均值池化聚合,并通过基于多层感知器(Multi-Layer Perceptron, MLP)的分类头计算类别概率:

$$p_\theta\left(y|\boldsymbol{X}\right)=\mathrm{Softmax}\left(\mathrm{MLP}\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{h}_t^{(L)}\right)\right)\tag{18}$$

对于包含B个流Token序列$\{\boldsymbol{X}^{(b)}\}_{b=1}^{B}$及其标签$\{y_b\}_{b=1}^{B}$的小批量数据,监督损失函数为平均交叉熵:

$$\mathcal{L}_{\mathrm{TC}}=-\frac{1}{B}\sum_{b=1}^{B}\log p_\theta\left(y_b|\boldsymbol{X}^{(b)}\right)\tag{19}$$

为了促进对模型容量的充分利用,确保微调过程中所有MoE专家得到充分更新,本文保留了MoE负载均衡损失$\mathcal{L}_{\mathrm{aux}}$,从而得到微调阶段的整体目标函数:

$$\mathcal{L}_{\mathrm{finetune}}=\mathcal{L}_{\mathrm{TC}}+\lambda_{\mathrm{aux}}\cdot\mathcal{L}_{\mathrm{aux}}\tag{20}$$

通过最小化此综合损失,Traffic-MoE能够将预训练期间学习到的通用流量知识精确地应用于下游分类任务,同时高效利用其强大的MoE架构,持续提升模型在特定流量安全任务中的分类性能。

最后,微调过程中还需考虑避免预训练知识的灾难性遗忘。因此,本文采用了一种与Traffic-MoE主干网层级结构相匹配的逐层学习率衰减(Layer-wise Learning Rate Decay, LLRD)策略。具体而言,编码通用字节级模式和数据包结构规律的浅层采用较小的学习率进行保守更新,而深层和分类头则使用较大的学习率以更快地适应特定任务的语义。对于具有L层且基础学习率为η_0的主干网络,第l层的学习率为

$$\eta_l=\xi^{L-l}\cdot\eta_0\tag{21}$$

其中,$\xi<1$为衰减因子。这种训练策略有效保留了预训练期间习得的通用流量知识,同时使更深层的网络能够快速适应下游任务的语义和分布特征。

通过结合时间切片、流级语义池化、稀疏专家正则化和分层优化等机制,Traffic-MoE在数据受限的流量安全场景中实现了高样本效率和鲁棒的泛化能力。即使标注样本有限,该模型也能有效地适应新的领域和不断演变的威胁环境,同时保留预训练期间学习到的广泛流量知识。

3　实验

为全面评估Traffic-MoE在异构网络环境下的流量分类性能与泛化能力,本文基于多源异构数据构建了一套系统的评估基准。

3.1　数据集与下游分类任务

3.1.1　预训练语料库

为促进在异构网络环境中学习通用且鲁棒的流量表示,本文聚合了CICIoT2023[33]、CICIoMT2024[34]、

USTC-TFC2016[35]、ISCXVPN2016[36]和UNSW-NB15[37]五个权威公开数据集，构建了一个包含200万条未标注会话流的预训练语料库。如表1所示，该语料库涵盖物联网、医疗物联网（Internet of Medical Things，IoMT）、网络入侵检测以及恶意软件分析等多个领域的流量，涉及TCP、UDP、HTTP、HTTPS、MQTT、DNS、FTP等主流网络协议。这种跨域与跨协议的数据组合，旨在驱动Traffic-MoE学习网络流量中通用的时序关联与内在语义，摆脱对特定数据集特征的依赖。

表1 预训练语料库包含的数据集

Table 1 Datasets in the pre-training corpus

数据集	领域	类别数	使用规模/会话流	关键协议
USTC-TFC2016	恶意软件分析	20	203.2×10^3	FTP, SMB, HTTP, HTTPS, SMTP, DNS, MySQL, BitTorrent, etc
UNSW-NB15	网络入侵检测	10	828.3×10^3	HTTP, ICMP, FTP, SSH, DNS, SMTP, etc
CICIoT2023	IoT安全	34	892.5×10^3	MQTT, HTTP, HTTPS, ICMP, DNS, SSH, ARP, etc
CICIoMT2024	IoMT安全	19	439.4×10^3	Wi-Fi, MQTT, HTTP, HTTPS, DNS, SSH, etc
ISCXVPN2016	加密流量识别	14	160.2×10^3	OpenVPN, HTTPS, SFTP, FTPS, SMTPS, POP3S, BitTorrent, etc

3.1.2 微调数据集与分类任务

在微调阶段，本文选用四个公开数据集，设计了六个具有代表性的下游分类任务，旨在从攻击检测敏感度、加密服务识别准确率等多个维度系统性评估模型在不同场景下的有效性。

（1）IoT场景下的攻击检测：基于CICIoT2023[33]数据集构建。该场景模拟了由105个真实IoT设备组成的复杂网络环境，涵盖拒绝服务攻击、暴力破解（brute force）、Web攻击和Mirai僵尸网络等7大类共33种细粒度攻击行为。此任务旨在评估模型在海量背景流量中对多样化攻击向量的精确识别与异常检测能力。

（2）IoMT场景下的攻击检测：基于CICIoMT2024[34]数据集构建。该数据集源自包含40台真实及模拟医疗设备的测试平台，覆盖Wi-Fi、MQTT等多种协议下的18种攻击。考虑到医疗环境的高度敏感性，此任务重点考察模型在特定垂直领域中对利用协议漏洞实施攻击的检测性能。

（3）VPN/NonVPN场景下的服务分类：基于ISCXVPN2016[36]数据集构建。由于VPN技术引入了多层加密与隧道封装，显著改变了流量的统计特征，该任务极具挑战性。本文构建了“非VPN场景服务分类”与“混合流量（VPN与NonVPN）服务分类”两个子任务，旨在验证模型在面对协议混淆与强加密干扰时，能否有效识别网络流量中隐藏的原始服务语义。

（4）Tor/NonTor场景下的服务分类：基于ISCX-Tor2016[38]数据集构建。该数据集包含经洋葱路由（The onion router，Tor）技术高度匿名化的流量，本文将其划分为“Tor流量”与“NonTor流量”两个子集，分别对其中的8种应用服务进行分类。此任务旨在检验模型在极端匿名场景下，对微弱流级特征与应用层指纹的深度捕捉能力。

3.2 实验设置

3.2.1 实现细节与参数设置

本文所提模型基于PyTorch 2.2框架实现。所有实验均在配备Intel Xeon Gold 5218R CPU（32核）和NVIDIA A100 GPU（40 GB VRAM）的高性能集群上执行，运行环境为Ubuntu 22.04 LTS和CUDA 11.7。

在数据处理阶段，本文首先使用SplitCap工具将原始PCAP文件分割为独立会话流。为减少统计噪声，本文过滤掉少于3个数据包的极短流，因为这些样本无法承载足够的信息。但对于样本极度稀缺的流量类别，本文保留所有会话流以避免进一步加剧样本不足的问题，确保模型能够充分训练。随后，本文采用Traffic2Token方法将会话流转换成Token序列表示。具体而言，本文对每个数据包提取6个包级特征和前40字节的有效载荷，并将每个会话截断为前10个数据包，以平衡模型计算效率和早期检测能力。基于流序列训练得到的词汇表大小为65 541。

在训练过程中，本文采用AdamW优化器[39]，并启用Flash Attention[40]机制优化Transformer结构的计算效率，batch size设置为32。在预训练阶段，学习率设置为3.0×10^{-4}，总共训练8个Epochs。损失函数结合了下一Token预测任务损失和MoE负载均衡辅助损失，权重系数λ_{aux}设置为0.02。在微调阶段，数据集按8∶1∶1的比例划分为训练集、验证集与测试集。同时，本文采用LLRD策略以保留预训练知识，设置层间衰减率$\xi=0.9$，初始学习率$\eta_0=5.0\times10^{-5}$。为防止过拟合，本文引入了基于验证集上宏观F1分数的早停（early stopping）机制，最大Epochs为40，若连续5个Epoch性能无提升，则提前终止训练。

3.2.2 评估指标

鉴于网络流量数据固有的长尾分布与类别不平衡特性，标准的准确率无法客观反映模型性能。因此，本文选取宏观精确率（Macro-Precision，M-PR）、宏

观召回率(Macro-Recall, M-RC)和宏观F1分数(Macro-F1 score, M-F1)作为核心评估指标。宏观指标通过独立计算各类别指标后再取算术平均得到,能够平等地衡量模型在少数类与多数类上的综合表现,避免评估结果被多数类主导。

对于每个给定的类别c和类别总数C,该类别各基础指标的定义如下:

$$\text{Precision}_c=\frac{\text{TP}_c}{\text{TP}_c+\text{FP}_c},\text{Recall}_c=\frac{\text{TP}_c}{\text{TP}_c+\text{FN}_c} \tag{22}$$

$$\text{F1}_c=2\cdot\frac{\text{Precision}_c\times\text{Recall}_c}{\text{Precision}_c+\text{Recall}_c} \tag{23}$$

其中,TP_c、FP_c、FN_c分别代表类别c的真阳性、假阳性和假阴性计数。最终的宏观平均指标通过下式计算,并以M-F1分数作为关键评判依据:

$$\text{Macro}-\text{Metric}=\frac{1}{C}\sum_{c=1}^{C}\text{Metric}_c \tag{24}$$

3.2.3 对比方法

为对Traffic-MoE的性能进行精确的定位与评估,本文选取了3种主流的基于"预训练-微调"范式的前沿模型ET-BERT[15]、NetGPT[16]和TrafficFormer[18]作为核心基线,同时为了验证预训练策略的有效性,本文还引入了2种机器学习方法AppScanner[7]、FlowPrint[8]和1种深度学习方法FS-Net[9]参与比较。为保证实验公平性,所有基线模型均基于官方开源代码复现,预训练方法统一使用本文构建的预训练和微调数据集,并在微调时采用相同的早停策略以提高训练效率,而机器学习或深度学习方法则统一使用本文构建的微调数据集。

(1)ET-BERT。基于BERT设计的加密流量分类方法,引入掩码突发模型(masked burst model)与同源突发预测(same-origin burst prediction)两大预训练任务,从海量未标注流量数据中学习上下文依赖关系。

(2)NetGPT。首次将生成式预训练Transformer应用于流量分析领域,将流量视为十六进制字节序列,统一了流量理解与生成任务,通过Prompt微调适配不同的下游任务。

(3)Trafficformer。在ET-BERT基础上提出"同源-同向-同流"(same origin-direction-flow)多分类预测任务,以更有效地捕获细粒度的数据包方向与顺序关系。

(4)AppScanner。一种经典的流量指纹识别方法,通过提取加密流量的数据包长度、到达间隔时间等统计特征,训练随机森林分类器,而无需解析加密载荷。

(5)FlowPrint。一种半监督的流量分类方法,侧重于挖掘流量的时间相关性(temporal correlations),通过构建流量指纹库实现对未知应用程序的识别。

(6)FS-Net。专为流量序列建模设计的端到端深度学习架构,在编码器-解码器结构中引入Bi-GRU,直接从原始数据包长度序列中学习流量的分类特征。

为了在统一的数据基准上公平对比各模型的表征能力,避免数据工程技巧的影响,本文对实验设置进行标准化,移除原始论文中使用的特定数据增强策略,并在相同的原始数据集上对所有模型进行微调。

3.3 与先进方法的比较

本节深入剖析Traffic-MoE在六个细粒度流量分类任务上的性能表现,并与多种基线模型进行横向对比,详细结果如表2所示。整体而言,Traffic-MoE在所有评估任务中均展现出显著的优越性,在最具挑战性的场景下,其M-F1分数相较于次优基线模型最高提升12.38%,从而确立了新的性能基准。

通过横向对比不同训练范式,预训练策略的优势得以充分显现。实验数据显示,传统机器学习方法与标准深度学习模型在多数任务上的性能普遍落后于采用"预训练-微调"范式的模型。这种性能差距的根本原因在于:在标注数据受限且分布长尾的情况下,直接监督学习难以提取具有泛化能力的特征表达,更无法准确推断复杂的协议状态转换逻辑和时间突发模式。相比之下,Traffic-MoE凭借在大规模未标注流量数据上的自监督预训练,成功习得并内化网络协议的通用语法,为下游任务提供了鲁棒的特征初始化,极大地降低了模型对特定任务标注数据的依赖。

在高度混淆的Tor加密场景中,Traffic-MoE表现出卓越的鲁棒性。在基于ISCXTor2016数据集的两个子任务中,Traffic-MoE的M-F1分数分别达到0.890 0和0.807 2,相较于最强基线分别提升了9.42%和11.28%。值得注意的是,ET-BERT、TrafficFormer等现有的强基线模型在Tor加密流量分类任务中均出现了明显的性能衰减。这种性能下降可归因于Tor协议的多层加密和固定大小的单元填充机制(fixed-size cell padding),这些机制使数据包长度趋于一致并引入了显著的结构噪声,进而干扰了对所有流量Token进行统一处理的密集型架构。Traffic-MoE则通过其稀疏MoE架构克服了这一挑战,其动态路由机制允许特定专家专注于解耦有效行为指纹与填充噪声,从而使模型在极端匿名性约束下仍能保持高判别力。

在ISCXVPN2016数据集上,Traffic-MoE同样保持领先优势。特别是在非VPN流量的子任务中,Traffic-MoE相较于最佳非预训练方法的提升超过16%;即便是对比生成式预训练模型NetGPT,其在M-F1分数上也实现了7.35%的显著增长。在更复杂的"混合流量"分类任务中,AppScanner等传统方法由于VPN隧道掩盖了表面头部特征而难以有效工作,但Traffic-MoE仍然在准确率和M-F1分数上领先次优模型

表2 在六个流量分类任务上与先进方法的对比

Table 2 Performance comparison with state-of-the-art methods across six traffic classification tasks

对比方法		ISCXTor2016(NonTor)				ISCXTor2016(Tor)				ISCXVPN2016(NonVPN)			
		准确率	精确率	召回率	F1分数	准确率	精确率	召回率	F1分数	准确率	精确率	召回率	F1分数
机器学习	AppScanner	0.9468	0.8587	0.6934	0.7436	0.8690	0.7921	0.7054	0.7254	0.5794	0.6756	0.6782	0.6588
	FlowPrint	0.9213	0.7408	0.7245	0.7138	0.4127	0.1907	0.2595	0.1738	0.5459	0.6283	0.5620	0.5617
深度学习	FS-Net	0.9406	0.7841	0.6518	0.6972	0.6679	0.2008	0.2527	0.2161	0.5093	0.4952	0.5021	0.4971
预训练	ET-BERT	0.9653	0.8322	0.7706	0.7960	0.7932	0.4093	0.4677	0.4303	0.5502	0.6670	0.6852	0.6755
	NetGPT	0.9655	0.8479	0.7911	0.8134	0.8506	0.7342	0.6805	0.6583	0.6618	0.7405	0.7511	0.7457
	TrafficFormer	0.9554	0.7663	0.7596	0.7418	0.7975	0.6083	0.5984	0.5428	0.6045	0.6831	0.6970	0.6842
	Traffic-MoE	**0.9827**	**0.9222**	**0.8707**	**0.8900**	**0.9089**	**0.8942**	**0.7879**	**0.8072**	**0.7613**	**0.7866**	**0.8158**	**0.8005**
对比方法		ISCXVPN2016(Mixed)				CICIoMT2024				CICIoT2023			
		准确率	精确率	召回率	F1分数	准确率	精确率	召回率	F1分数	准确率	精确率	召回率	F1分数
机器学习	AppScanner	0.6249	0.7414	0.7355	0.7248	0.6929	0.7473	0.6783	0.6397	0.5266	0.5535	0.4471	0.4619
	FlowPrint	0.5659	0.6033	0.6431	0.5952	0.0132	0.0373	0.1364	0.0501	0.4578	0.3166	0.2048	0.1989
深度学习	FS-Net	0.5357	0.4617	0.4681	0.4631	0.5788	0.5319	0.5050	0.4533	0.4903	0.5265	0.4046	0.4094
预训练	ET-BERT	0.6106	0.7503	0.7549	0.7521	**0.9769**	0.5620	0.5271	0.5255	0.7455	0.5416	0.5172	0.5217
	NetGPT	0.6973	0.8003	0.8072	0.8034	0.8958	0.8326	0.8335	0.8300	0.7684	0.7756	0.6709	0.6962
	TrafficFormer	0.6373	0.7689	0.7285	0.7422	0.8912	0.7572	0.7835	0.7530	0.7636	0.7172	0.6411	0.6594
	Traffic-MoE	**0.7679**	**0.8306**	**0.8377**	**0.8332**	**0.9769**	**0.8849**	**0.8860**	**0.8839**	**0.8588**	**0.8007**	**0.7701**	**0.7824**

注:加粗数据为最优结果,下划线数据为次优结果。

10.12%和3.71%,这表明模型能够有效克服协议混淆并准确识别底层服务。该优势源于模型的自注意力机制,其能够捕获在隧道封装下保持不变的内在流量演化模式,从而抵抗外部协议头的干扰。

攻击检测场景中,Traffic-MoE在海量背景流量下识别恶意活动的优势尤为显著。在CICIoMT2024数据集上,Traffic-MoE相较于次优模型NetGPT在各项指标上实现了6.28%~9.05%的全面提升。而在CICIoT2023数据集上,Traffic-MoE的准确率相比次优模型提高了11.76%,更关键的是,其M-RC提升了14.79%。对于入侵检测系统而言,召回率的显著提升意味着漏报率(false negative rate)的大幅降低,这对于及时发现隐蔽攻击、保障网络基础设施安全具有重要意义。

同时,实验观察到,由于严重的类别不平衡和攻击变种的多样性,AppScanner、FlowPrint等传统方法在CICIoT2023数据集上的性能出现剧烈下滑,这主要因为基于统计特征或固定指纹的方法难以覆盖长尾分布中的攻击变种。而Traffic-MoE通过在预训练阶段内化异常模式来解决这一问题,使模型能够从常见攻击泛化到新型变种,从而展现出稳健的性能。值得注意的是,FlowPrint在此类任务中的表现最差,进一步证实了在面对高动态、高变异的现代网络攻击时,依靠流量指纹库的方法已难以满足安全需求,而基于深度表征学习的Traffic-MoE则提供了更优的应对方案。

3.4 各流量场景下模型能力分析

为了探究Traffic-MoE在实际网络场景中的性能边界,本节通过混淆矩阵直观展示不同流量分类任务中模型在细粒度类别间的混淆情况,具体如图4所示,据此对模型能力展开深入分析。

3.4.1 强匿名环境下的鲁棒性

通过对比图4中模型在ISCXTor2016数据集构建的两个子任务上的表现,可以清晰地观察到匿名化机制引发的“特征侵蚀”(feature erosion)效应。NonTor场景下表现优异的Audio类别,在Tor场景下准确率骤降至0.56,且有34%的样本被误判为Browsing类别。这一现象的根本原因在于,语音流量的识别高度依赖数据包到达时间间隔与包长分布作为指纹,而Tor协议的随机路由延迟与单元批处理机制彻底破坏了这些细粒度特征,导致其统计分布向通用Web浏览流量坍缩。Email流量的情况类似,因其握手特征被加密协议层层包裹,准确率降至0.30,同样大量混淆于Browsing流量。

与之相反,模型对P2P、VoIP等流量的识别在Tor加密场景中展现出强大的鲁棒性,两者准确率分别达到0.99和1.00。尽管浅层的包级特征因流量加密而丢失,但Traffic-MoE成功捕获了这些协议特有的流级行为指纹,例如P2P协议中的去中心化节点发现行为。这一结果表明,模型能够跨越底层加密噪声,聚焦于学习强匿名环境下仍保持有效的协议语法。

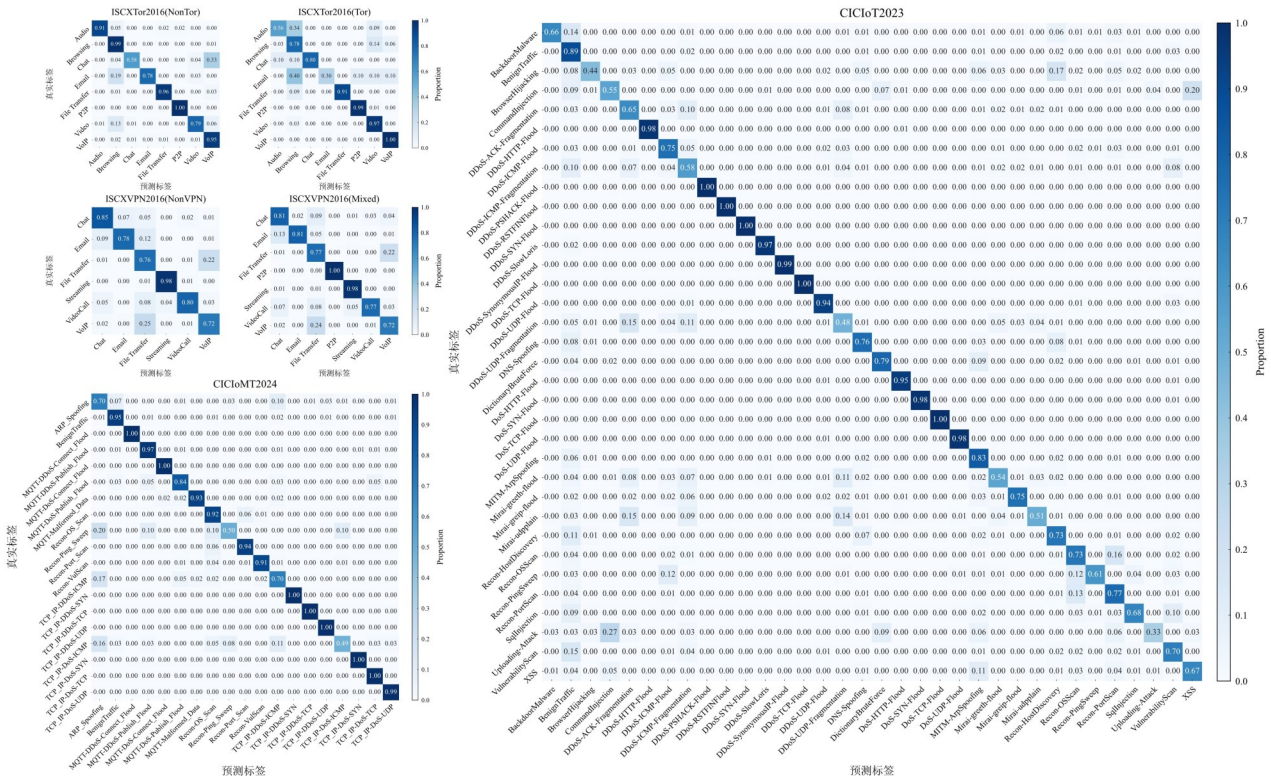

图 4　Traffic-MoE 在不同下游任务中的性能混淆矩阵

Figure 4　Confusion matrices of Traffic-MoE across different downstream tasks

3.4.2　VPN 隧道封装下的不变性学习

与 Tor 协议的强力整形不同,VPN 协议主要通过隧道封装引入额外的头部开销与加密载荷,即便如此,模型在此场景下依旧表现出稳健性能。从图 4 中可观察到,在非 VPN 和混合场景下,模型的类别混淆情况保持了高度一致,典型表现为 VoIP 和 File Transfer 之间的相互混淆,这种误分类源于两类流量固有的特征相似性,而非 VPN 加密引入的噪声所致。关键在于,Traffic-MoE 在面对 VPN 加密流量时并未丢失对原始流特征的判别力,证明其未被外层的 VPN 协议封装头误导。此外,对于 Streaming 和 P2P 等复杂类别,模型在混合场景下仍能保持近乎完美的识别准确率。这表明模型使用的自注意力机制能够有效突破外部 VPN 协议封装,捕获长距离流依赖性,进而解耦有效载荷行为与传输封装的语义。

3.4.3　入侵检测场景中强大的识别能力

通过观察模型在 CICIoMT2024 与 CICIoT2023 数据集上的分类表现,可以发现绝大多数误判主要集中在同一攻击家族内部,例如 Web 攻击之间的相互混淆,而“攻击与正常流量”等跨家族类别的区分边界始终清晰。这一特性和实际入侵检测系统的需求高度契合,防御者通常更关注“攻击家族”层面的识别准确性,因为同类攻击行为的应对策略总体相似。此外,Traffic-MoE 对于不同攻击行为的高识别准确率意味着其能够成功拦截绝大多数威胁,在防御渗透方面表现出色。同时,模型对于良性流量的误报率维持在一个较低水平,表明模型实际部署后对正常业务流量的影响较小,不会显著增加正常流量的延迟。

3.5　少样本场景下的性能对比

在实际的网络安全运营中,获取大规模的高质量标注数据通常成本高昂且耗时长。为评估模型在“标注稀缺”场景下的鲁棒性,本文设计了少样本评估实验。实验选取了包括 ISCXTor2016(NonTor)、ISCXVPN2016(Mixed)和 CICIoMT2024 在内的三个代表性数据集,分别采样原训练集 5%、10%、20%、40%和 100% 的数据构建差异化训练子集,并在统一的测试集上评估各模型性能。各数据集上的实验结果如图 5、图 6 和图 7 所示。

整体而言,随着标注样本量的缩减,所有模型的性能均呈现不同程度的衰减。通过横向对比,实验结果揭示了三种不同的性能演变模式。以 FS-Net 为代表的从头训练的深度学习方法表现出较差的鲁棒性,由于缺乏先验知识,其深层参数在极少样本下难以收敛,导致性能显著下降。例如,在 CICIoMT2024 数据集上,其 M-F1 分数的最大跌幅超过 85%。相比之下,以 AppScanner 为代表的统计机器学习方法表现出较

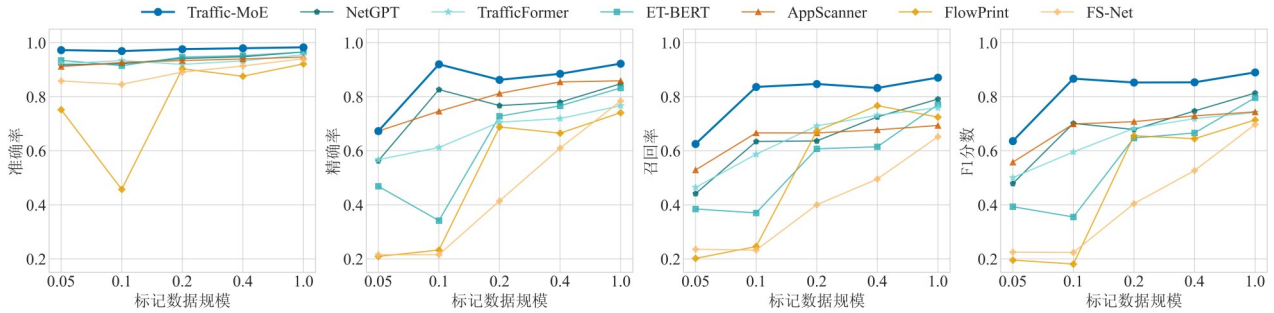

图5 在ISCXTor2016(NonTor)数据集上标注数据规模对各方法性能的影响

Figure 5 Impact of labeled data scale on performance for ISCXTor2016 (NonTor)

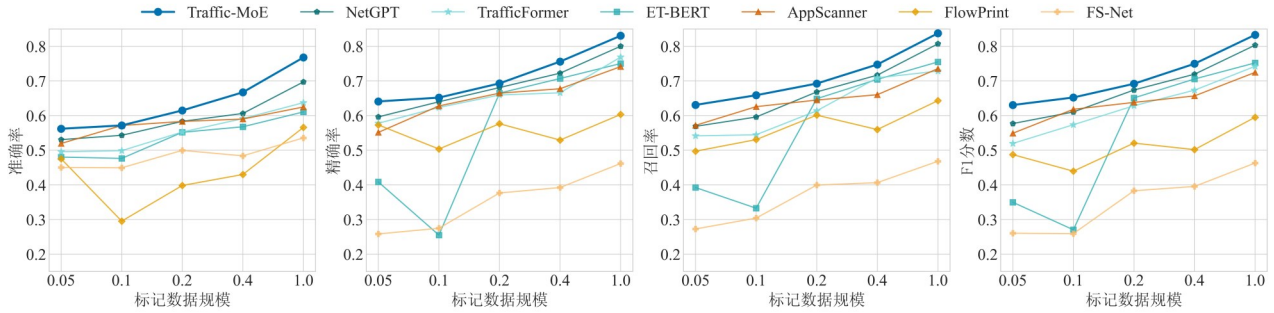

图6 在ISCXVPN2016(Mixed)数据集上标注数据规模对各方法性能的影响

Figure 6 Impact of labeled data scale on performance for ISCXVPN2016 (Mixed)

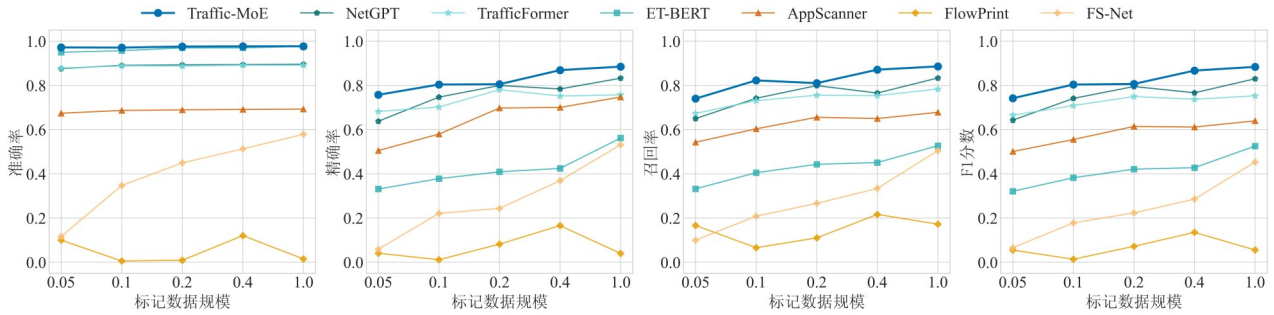

图7 在CICIoMT2024数据集上标注数据规模对各方法性能的影响

Figure 7 Impact of labeled data scale on performance for CICIoMT2024

强的稳定性,但受限于手工特征的表达能力,其性能存在明显的瓶颈,无法通过增加数据规模实现进一步的性能突破。而以Traffic-MoE为代表的预训练模型展现出最佳样本效率。得益于预训练阶段内化的通用流量语义,Traffic-MoE即使在5%的极端数据规模下,其性能依然显著优于其他基线方法。尤为关键的是,在ISCXTor2016(NonTor)和CICIoMT2024数据集上,Traffic-MoE仅需10%的标注数据微调,即可达到甚至超越ET-BERT、TrafficFormer等多数基线模型在100%数据量下的性能水平,这一结果有力验证了预训练MoE架构成功内化了可用于迁移的通用流量语义。

在ISCXTor2016(NonTor)数据集上的服务分类任务中(图5),AppScanner方法表现出强大的鲁棒性,在多个采样点上超越了NetGPT等部分预训练模型。这表明在非加密的Tor流量中,统计指纹依然是有效的判别特征。然而,Traffic-MoE实现了性能与稳定性的最佳平衡,它不仅在所有数据规模下均保持M-F1>0.60的绝对领先优势,而且性能衰减曲线相比其他预训练模型更为平缓,这证明了MoE架构在特征稀疏场景中具有更好的泛化能力。

在ISCXVPN2016(Mixed)数据集上的混合流量分类任务中(图6),实验结果呈现出一个明显的性能分水岭。当标注数据比例低于20%时,ET-BERT等密集预训练模型的性能出现显著下降,甚至被传统机器学习方法AppScanner反超。这表明,当微调数据量不足

以约束庞大的参数空间时,密集型架构容易出现过拟合和表征崩溃问题。相比之下,Traffic-MoE在全数据比例区间(5%~100%)内始终保持最优性能。有力证明了本文所提模型可有效缓解少样本条件下的过拟合风险,确立了其在复杂加密场景下的实用价值。

在CICIoMT2024数据集上的攻击检测任务中(图7),预训练模型的优势进一步凸显。与服务分类不同,攻击流量通常包含极微弱的异常模式。FlowPrint的彻底失效和FS-Net的性能严重下滑,证明无先验知识的模型很难从少量样本中捕捉此类微弱特征。而Traffic-MoE凭借预训练阶段习得的上下文理解能力,成功实现了异常模式的快速迁移,在5%标注数据量下依然维持了0.742 0的M-F1分数,展现了在新型攻击早期检测任务中的应用潜力。

3.6　消融分析

为量化模型各关键组件对分类性能的贡献,验证架构设计的合理性,本文基于表3汇总的实验结果,在六个代表性下游任务上展开了多维度的消融分析。本节将依次探讨MoE架构的有效性、预训练策略的必要性、辅助损失函数的作用以及输入表征的互补性。

表3　在不同流量场景下的消融实验

Table 3　Ablation study across diverse traffic scenarios

模型版本	ISCXTor2016(NonTor)				ISCXTor2016(Tor)				ISCXVPN2016(NonVPN)			
	准确率	精确率	召回率	F1分数	准确率	精确率	召回率	F1分数	准确率	精确率	召回率	F1分数
Traffic-MoE(Ours)	0.9827	0.9222	**0.8707**	0.8900	**0.9089**	**0.8942**	**0.7879**	**0.8072**	**0.7613**	0.7866	**0.8158**	**0.8005**
MoE→Dense	**0.9831**	**0.9229**	0.8655	**0.8904**	0.9038	0.8656	0.7638	0.7767	0.7178	0.7570	0.7900	0.7713
w/o PT	0.9643	0.7753	0.7271	0.7481	0.8304	0.7071	0.6392	0.6311	0.6062	0.6327	0.6095	0.6153
(MoE→Dense) w/o PT	0.9674	0.8064	0.8054	0.7938	0.8506	0.7056	0.6541	0.6564	0.6074	0.6334	0.6533	0.6389
w/o Aux Loss	0.9800	0.8786	0.8658	0.8718	0.8962	0.8734	0.7683	0.7913	0.7370	**0.7900**	0.7990	0.7929
w/ Header Only	0.9750	0.8776	0.7983	0.8290	0.8759	0.7721	0.6385	0.6492	0.7308	0.7665	0.7846	0.7715
w/ Payload Only	0.9517	0.8935	0.7509	0.8055	0.8127	0.7597	0.7106	0.6996	0.4967	0.6436	0.4450	0.4472
模型版本	ISCXVPN2016(Mixed)				CICIoMT2024				CICIoT2023			
	准确率	精确率	召回率	F1分数	准确率	精确率	召回率	F1分数	准确率	精确率	召回率	F1分数
Traffic-MoE(Ours)	**0.7679**	**0.8306**	**0.8377**	**0.8332**	0.9769	0.8849	0.8860	0.8839	0.8588	0.8007	0.7701	0.7824
MoE→Dense	0.7600	0.8101	0.8302	0.8193	**0.9790**	**0.8957**	**0.8947**	**0.8946**	**0.8594**	0.8090	**0.7765**	**0.7894**
w/o PT	0.6255	0.6883	0.6865	0.6854	0.9685	0.8110	0.8236	0.8154	0.7833	0.6757	0.6584	0.6640
(MoE→Dense) w/o PT	0.6111	0.6958	0.6680	0.6770	0.9728	0.8350	0.8346	0.8294	0.7867	0.7054	0.6586	0.6741
w/o Aux Loss	0.7353	0.8064	0.8205	0.8107	0.9767	0.8674	0.8725	0.8692	0.8575	**0.8116**	0.7630	0.7819
w/ Header Only	0.7425	0.8050	0.8150	0.8089	0.9263	0.8444	0.8539	0.8481	0.8395	0.7679	0.7211	0.7358
w/ Payload Only	0.5199	0.7147	0.5382	0.5648	0.4879	0.4603	0.4459	0.4210	0.3965	0.5782	0.3314	0.3710

注:加粗数据为最优结果,下划线数据为次优结果。

3.6.1　MoE架构与稀疏激活的有效性

为验证MoE架构的性能增益,本文构建了参数总量相当的密集型基线模型MoE→Dense,即将原模型中的MoE层替换为标准的FFN层,并在相同条件下进行训练。实验结果表明,Traffic-MoE(Ours)在所有任务上均取得了与MoE→Dense相近的性能。尤为关键的是,MoE模型在推理过程中仅稀疏激活了部分专家参数,却能达到全参数激活模型的性能水准,证明了该架构优越的参数效率(parameter efficiency)。而在ISCXTor2016(Tor)和ISCXVPN2016(Mixed)等高混淆、高复杂度的场景中,Traffic-MoE展现出明显的性能优势,其M-F1分数分别超越Dense架构3.93%和1.69%。这表明,在处理异构且特征重叠严重的流量时,MoE的"专家特化"机制发挥了关键作用,通过动态路由将不同模式的混淆流量分配至特定的专家网络,模型可更精细地解耦复杂的流量行为,而这是共享参数的Dense架构难以实现的。

3.6.2　预训练策略的必要性

为探究预训练对模型表征能力的贡献,本文移除预训练阶段,直接基于下游任务标注数据从零训练,构建了w/o PT版本模型。实验结果充分证明了"预训练-微调"范式的必要性,移除预训练后,MoE和Dense两种架构的检测性能均出现显著下滑。其中,Traffic-MoE(Ours)在ISCXVPN2016(NonVPN)数据集上的M-F1分数下降23.14%,证实了通过无监督预训练习得的通用流量语法是下游任务实现高性能的关键初始化基础。值得关注的是,Traffic-MoE(Ours)缺失预训练时的性能降幅普遍大于MoE→Dense。例如,在Tor场景中移除预训练后,MoE架构的性能下降21.82%,而Dense架构的性能下降15.49%。这揭示了MoE架构对先验知识的强依赖性,MoE的核心在于路由网络的决策能力,若缺乏预训练提供的良好特征空间,随

机初始化的路由网络难以在有限的标注数据下习得有效的专家路由策略，导致模型难以收敛至最优解。

3.6.3 负载均衡损失的作用

为验证负载均衡辅助损失在稳定MoE训练中的作用，本文评估了在训练中移除该损失项的w/o Aux Loss版本模型。实验数据显示，去除负载均衡约束后，模型在各项任务上的M-F1分数普遍下降了0.06%~2.70%，验证了辅助损失在防止路由崩溃方面的积极作用。若缺乏该损失项的正则化约束，门控网络容易陷入局部最优，导致模型退化为容量受限的Dense架构。而辅助损失的使用可以促进计算负载的均匀分配，最大化模型整体容量的利用率。

3.6.4 输入表征的贡献分析

本文所提模型默认采用数据包的“头部特征”和“部分载荷”构建流序列。为解构两类特征的贡献，本文对比了仅使用头部特征（w/ Header Only）和仅使用载荷（w/ Payload Only）的模型变体。

实验结果表明，头部特征在流量分类中占据主导地位。尤其在CICIoMT2024和CICIoT2023构建的攻击检测场景中，w/ Header Only的性能优势最为显著，在CICIoMT2024数据集上，其M-F1分数相较于w/ Payload Only实现翻倍提升。这是因为网络攻击本质上表现为异常流量统计行为，如特定的包长序列与到达间隔模式，这些侧信道信息完整保留于数据包头部。而攻击流量的载荷多由随机填充数据或重复指令构成，缺乏判别性语义信息，导致w/ Payload Only在攻击检测任务中表现不佳。

但在Tor场景中，w/ Payload Only的M-F1分数反而优于w/ Header Only。这一反直觉的现象可归因于Tor协议独特的固定单元填充机制。Tor强制将数据包整形为统一的512字节单元，该“流量整形”操作使头部特征中关键的“包长度”信息严重同质化，大幅削弱了头部特征的判别能力。在此特定约束下，载荷序列中隐含的应用程序级交互模式成为更有效的判别依据。

尽管单一模态在特定场景下存在局限性，包括Tor场景下的头部特征混淆和攻击检测中载荷的语义稀疏，但完整模型Traffic-MoE（Ours）始终保持最佳性能。例如在头部特征混淆的Tor场景中，完整模型通过融合两类特征，将M-F1分数提升至0.807 2，相较于w/ Payload Only提升15.38%。这充分证明了头部特征与数据载荷在信息空间中的互补性，模型能够动态地从更可靠的信息源提取特征，通过多视角的决策融合弥补单一视角的局限性，进而在各类异构网络环境中实现稳健的分类精度。

3.7 模型推理效率分析

本文的核心创新点之一是引入MoE架构，旨在通过“稀疏激活机制”突破传统深度模型的“性能-效率”权衡。本节通过严格的推理基准测试，验证Traffic-MoE在保持卓越分类性能的同时，实现计算开销显著降低的可行性。具体而言，本文在CICIoMT2024数据集上构建标准化的推理测试流程，对比对象包括Traffic-MoE（Ours）和密集架构变体MoE→Dense。测试覆盖不同负载场景（batch size $B \in \{8, 16, 32, 64\}$），在GPU充分预热（warm-up）后，记录吞吐量（throughput）、平均延迟（average latency）以及峰值显存消耗（peak GPU memory consumption）三个关键指标的统计平均值。实验结果如图8所示。

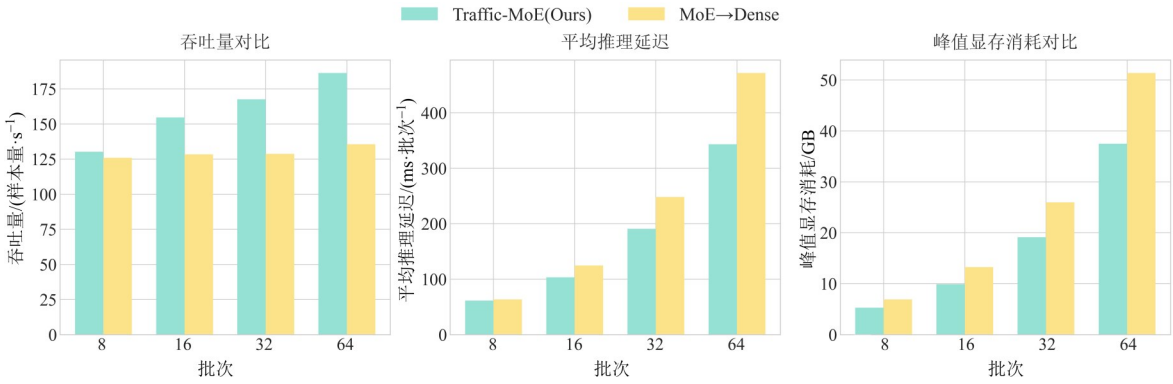

图8 引入MoE架构后对模型推理效率的影响

Figure 8 Impact of the MoE architecture on inference efficiency

整体而言，Traffic-MoE（Ours）在推理效率上具备显著优势，且该优势随着负载的增加呈现非线性放大的趋势。在 $B=64$ 的高负载场景下，Traffic-MoE（Ours）相较于同构的MoE→Dense模型，吞吐量提升37.45%，平均延迟降低27.25%，峰值显存节省27.04%。这一结果直接证实了MoE架构可以在不牺牲模型容量的前提下，有效卸载约40%的冗余计算负担。本文分析认为，这种显著的效率提升源于MoE

架构设计的两大核心特性。

(1)条件计算带来的浮点运算量(Floating Point Operations,FLOPs)缩减。密集模型对每个输入Token均需激活全量参数,计算复杂度为$\mathcal{O}(N_{\text{total}})$。而Traffic-MoE引入了动态路由机制,对于每个Token仅激活最相关的k个专家(本实验中$k=2$),使推理过程的FLOPs与总参数量解耦,实际计算复杂度降至$\mathcal{O}(N_{\text{active}})$,其中$N_{\text{active}} \ll N_{\text{total}}$。因此,模型在保持大容量表征能力的同时,实现了推理速度的倍增。

(2)动态激活显存的优化。在深度学习推理中,显存瓶颈多源于存储中间层激活值而非静态权重。虽然对比模型的总参数量相当,但由于MoE架构前向传播路径的稀疏性,使得仅被选中的专家才会产生中间激活状态,这种"激活稀疏性"大幅降低了动态显存的峰值占用,使得Traffic-MoE可在更低的硬件资源条件下部署于实际环境,或支持更大的批处理大小。

3.8　专家激活可视化

为了解析稀疏MoE模块的内部决策逻辑,本节对不同训练阶段结束后的模型在执行推理时各层专家的激活概率进行可视化分析,并绘制热力图,结果如图9和图10所示。图中颜色的深浅代表模型骨干网中特定层级中某个专家的激活频率。

从预训练到微调的路由模式演变,为知识迁移和专家特化提供了经验证据。在预训练阶段,虽然测试数据集涉及不同流量场景,但其对应的激活热力图呈现出高度的"模式同质性",表明模型主要依赖于从大规模流量语料库习得的通用语法规则进行推理。此时专家网络尚未完全实现特化,而是通过协同工作构建鲁棒的通用特征空间,为后续任务提供普适性的表征基础。然而,经过微调后,专家激活模式发生了显著的任务驱动型重构,在特定任务梯度的引导下,路由策略表现出明显的聚类转变,将流量导向特定的专家子集。这一现象证明,通过监督微调,预训练的专家已成功转型为专注于处理特定下游任务特征的领域专家。该演化轨迹表明,Traffic-MoE可以有效利用通用先验知识作为初始化基础,并通过动态调整路由逻辑实现计算资源的专门分配,从而在保持泛化能力的同时,最大化下游任务的分类准确率。

此外,专家激活模式在不同网络深度上呈现差异化的分布特征。浅层专家在所有下游任务中均表现出高度一致的激活模式,表明其作为通用语法解析器,专注于提取任务无关的协议特征,包括TCP/IP报头字段、TLS握手序列等。而深层专家仅在语义相关的任务之间表现出相似的激活模式,例如VPN和Tor等加密隧道相关场景,说明其已趋向于抽象的、高层次的行为语义建模。中层专家的激活在不同任务之间仍存在显著的分布差异,反映模型正在经历严格的语义转换和特征解耦过程。

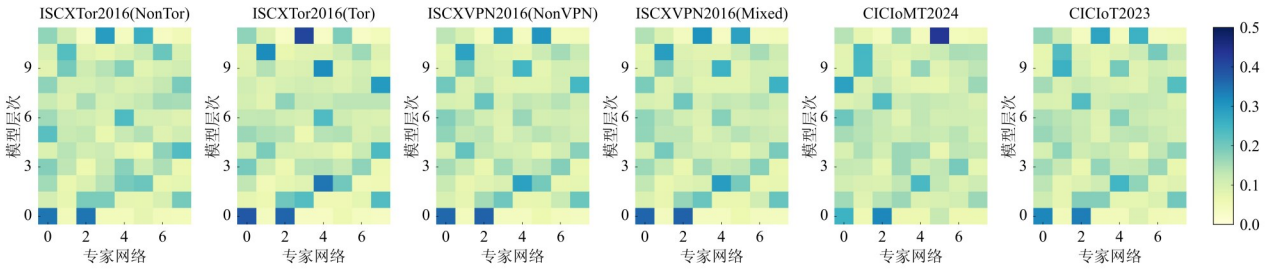

图9　预训练阶段模型各层专家的激活情况

Figure 9　Expert activation patterns across layers during the pre-training phase

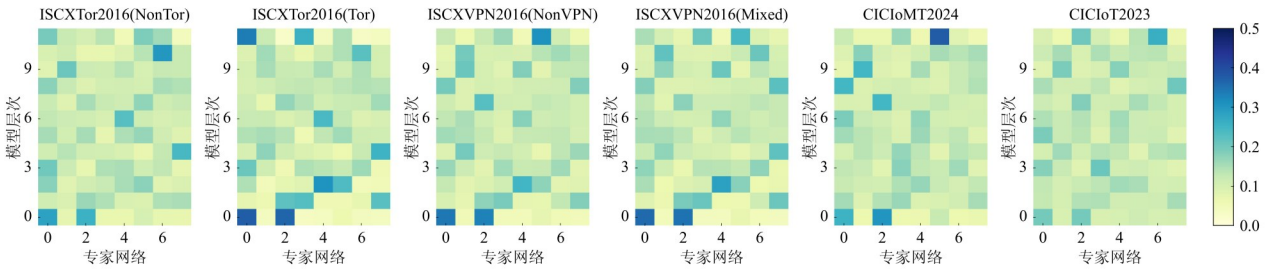

图10　微调阶段模型各层专家的激活情况

Figure 10　Expert activation patterns across layers during the fine-tuning phase

3.9 Top-k 参数实验

MoE 架构的性能对激活专家数 k 高度敏感。为了在控制推理成本的前提下探究最优的模型容量分配策略，本文在各层 MoE 总激活参数量固定的条件下，评估活跃专家数对模型性能的影响。本节实验的核心目标在于深入探索“专家个体粒度”与“路由多样性”之间的最佳平衡点。

实验在六个数据子集上开展，保持总专家池大小 $N=8$ 不变，依次设置激活专家数 $k\in\{1,2,3,4,6\}$ 进行对比。结果如表 4 所示，w/{Top$k$} 表示在模型 MoE 架构中激活专家数为 k，其中 w/{Top2}（本文默认配置）在绝大多数任务中展现出最优的性能与鲁棒性。具体而言，w/{Top2} 在大部分数据集上均实现了最高的 M-F1 分数，相较于次优配置最大提升了 3.01%。即便在 CICIoMT2024 非最优场景中，其与最优版本的差距也极小，充分证明了该配置的显著优势。本文将 $k=2$ 的优越性归因于其在“专家能力”与“路由灵活性”之间达成了最佳折中。

（1）优于 $k=1$ 的单专家硬路由。当 $k=1$ 时，虽然单个专家的容量最大且表征能力最强，但迫使路由网络进行“非此即彼”的硬决策。这种机制缺乏容错性，一旦路由网络将 Token 误分配给不擅长的专家，由于缺乏辅助专家的修正，模型的分类性能将显著下降。表 4 数据显示，w/{Top1} 在 ISCXVPN2016（Mixed）等复杂场景中的表现普遍劣于 w/{Top2}，证实了路由瓶颈的存在。

（2）优于 $k>2$ 的多专家泛路由。随着 k 值的增大（$k=4,6$），单个专家的容量大幅减小，导致其处理复杂特征的能力减弱。更关键的是，过高 k 值会导致“专家同质化”，若每个 Token 都激活绝大部分专家，MoE 架构将实际退化为次优的密集型网络，从而丧失稀疏特化的优势。实验中 Top6 在多个任务上的性能回落，验证了过度分散计算资源会抑制专家特化的形成。

总之，$k=2$ 的配置实现了“最佳平衡”，既保留了足够大的专家容量以建模复杂流量模式，又通过双专家协作机制提供必要的路由容错性与表征平滑度，最终在多个下游任务中达到最佳性能。

表 4 不同 Top-k 参数对模型性能的影响

Table 4 Impact of the Top-k parameter on model performance

Traffic-MoE	ISCXTor2016(NonTor)				ISCXTor2016(Tor)				ISCXVPN2016(NonVPN)			
	准确率	精确率	召回率	F1 分数	准确率	精确率	召回率	F1 分数	准确率	精确率	召回率	F1 分数
w/ {Top1}	0.9812	0.8906	0.8383	0.8597	0.8987	0.8700	0.7785	0.7958	0.7483	0.7722	0.7850	0.7771
w/ {Top2}	0.9827	**0.9222**	**0.8707**	**0.8900**	**0.9089**	**0.8942**	**0.7879**	**0.8072**	**0.7613**	**0.7866**	**0.8158**	**0.8005**
w/ {Top3}	0.9818	0.9142	0.8516	0.8782	0.8734	0.7886	0.7226	0.7138	0.7166	0.7422	0.7742	0.7569
w/ {Top4}	0.9814	0.8798	0.8652	0.8714	0.9038	0.8626	0.7815	0.7948	0.7111	0.7608	0.7446	0.7483
w/ {Top6}	**0.9837**	0.9056	0.8691	0.8845	0.8886	0.8395	0.7637	0.7729	0.6885	0.7280	0.7624	0.7418
Traffic-MoE	ISCXVPN2016(Mixed)				CICIoMT2024				CICIoT2023			
	准确率	精确率	召回率	F1 分数	准确率	精确率	召回率	F1 分数	准确率	精确率	召回率	F1 分数
w/ {Top1}	0.7421	0.8198	0.8210	0.8177	0.9773	0.8796	0.8848	0.8812	0.8555	0.7876	0.7632	0.7735
w/ {Top2}	**0.7679**	**0.8306**	**0.8377**	**0.8332**	0.9769	**0.8849**	0.8860	0.8839	**0.8588**	**0.8007**	**0.7701**	**0.7824**
w/ {Top3}	0.7322	0.8090	0.7993	0.8001	**0.9780**	0.8719	0.8766	0.8727	0.8537	0.7918	0.7652	0.7760
w/ {Top4}	0.7338	0.8276	0.8106	0.8185	0.9777	0.8809	**0.8887**	**0.8846**	0.8527	0.7988	0.7633	0.7780
w/ {Top6}	0.7562	0.8220	0.8191	0.8184	0.9768	0.8695	0.8771	0.8710	0.8514	0.7841	0.7680	0.7731

注：加粗数据为最优结果，下划线数据为次优结果。

3.10 流量序列参数分析

流量序列的构造方式直接决定模型输入的信息密度与感受野。本文设计的流量序列构造涉及两个关键维度：单包载荷截断长度和数据包数量。为确定这两个超参数在“表征能力”与“计算效率”之间的最优平衡点，本节设计了两组详尽的控制变量实验。需强调的是，为了消除词表不匹配或流序列截断带来的潜在偏差，本文采取严谨的实验方案。针对每组参数配置（如 10 pkt/60 B 或 20 pkt/40 B），均重新执行全流程训练，包括从头构建预训练语料库、重新生成词表、从零开始预训练以及最终的微调，确保所有对比均在公平且最优的模型状态下进行。

3.10.1 载荷截断长度的影响

固定序列长度为 10 个数据包，探究不同载荷截断阈值对模型性能的影响，实验结果如图 11 所示。实验数据呈现出清晰的“先升后降”的趋势：当载荷长度从 10 字节增加到 40 字节时，模型在所有数据集上的性能持续提升并达到峰值；当载荷长度进一步超过 40 字节时，性能提升趋于平缓，在部分加密流量任务中甚至出现性能下降。该现象凸显了网络流量中固有的头部语义丰富性，数据包载荷的初始段聚集了 TLS 记录头部、HTTP 请求头等关键检测特征；超过此

阈值的尾部有效载荷则主要由高熵加密数据或随机填充组成,此类数据不仅缺乏区分性语义,还会作为随机噪声稀释关键特征的注意力权重。因此,本文将载荷截断长度设置为40字节,既能有效包含绝大多数协议的关键头部信息,又能在保留高密度区分性信息的同时,最大限度地缩短流量序列长度以降低推理开销。

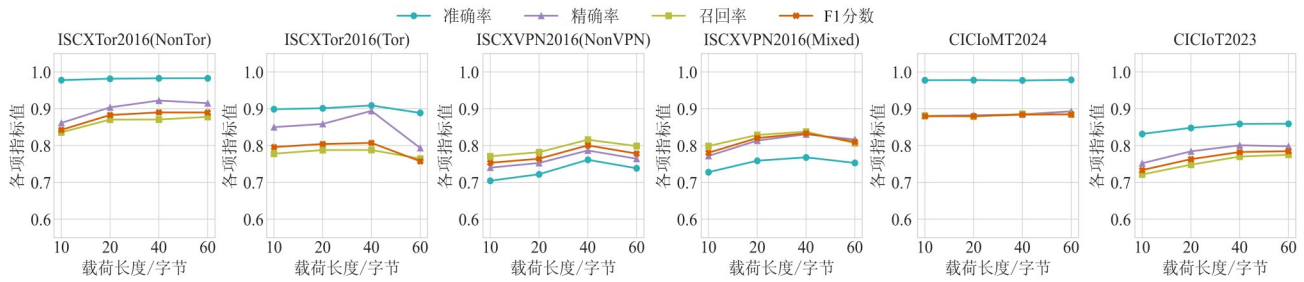

图11　有效载荷长度对模型性能的影响

Figure 11　Impact of payload length on model performance

3.10.2　流序列中数据包数量的影响

固定载荷长度为40字节,探究流序列中数据包数量的影响,实验结果如图12所示,从中可以观察到类似的边际效应递减规律。模型性能在$K=3$到$K=10$的区间内显著提升,但当$K>10$时,性能出现停滞甚至下降。该趋势与网络协议的基本交互机制相符,主要的区分信号在初始连接建立阶段即被编码到“协议语法”中,包括TCP三次握手、TLS密钥交换以及证书协商等过程。此阶段的交互模式具有高度确定性且与协议类别强相关,因此随着序列包数增加,能够覆盖更完整的行为签名,进而带来性能提升。当连接进入后续的批量数据传输阶段,流量行为趋于同质化,主要特征为持续的有效载荷传输,提供的增量信息极少,同时包数增加会引入更多的内容噪声,影响流量分类的准确性。因此,选取前10个数据包作为输入,足以捕获完整的会话建立逻辑,同时有效避免长序列处理带来的计算复杂度二次增长,实现长流的有效表示。

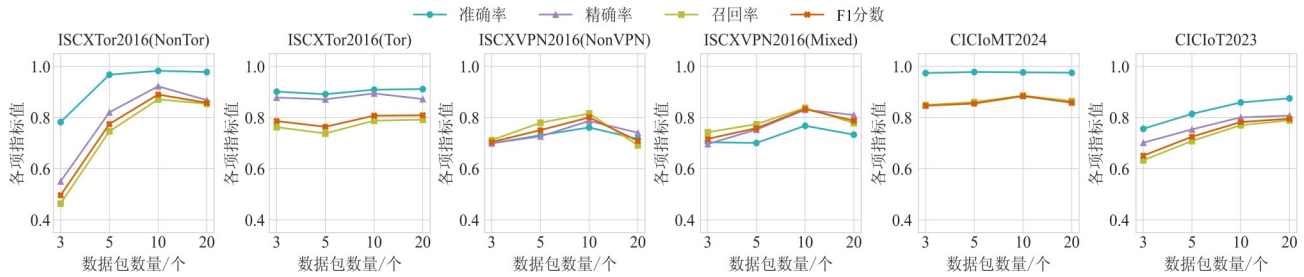

图12　流序列中数据包数量对模型性能的影响

Figure 12　Impact of the number of packets per flow on model performance

4　结束语

针对网络流量分析领域长期面临的高质量标注数据稀缺与大型模型推理开销高昂两大挑战,本文提出了一种高效的流量分类基础模型——Traffic-MoE。为突破标注数据稀缺瓶颈,本文采用“预训练-微调”范式,利用自监督任务从大规模无标注语料库中内化通用的语义表征与时序关联,从而仅需少量下游任务标注数据便可实现高效适配。为解决模型推理开销难题,本文创新性地将MoE架构引入流量分类领域,通过稀疏激活机制,在推理时仅动态调用少部分参数,从而在维持高模型容量的同时,大幅降低了计算资源消耗。为全面评估模型性能,本文基于四个公开数据集构建了六个分类任务,覆盖物联网攻击检测、加密流量分析和服务分类等多种场景。实验结果表明,Traffic-MoE在所有测试任务上均达到了当前最佳性能,显著超越了现有的预训练基线模型,并在少样本学习和跨域泛化能力上展现出显著优势。

尽管Traffic-MoE表现出色,但仍存在可优化之处。目前模型依赖单一视角建模,且在极端少样本学习和开集识别等特定挑战上仍有提升空间。因此,未来的工作将聚焦于:探索多视角融合的流量建模方法以捕获更全面的语义信息;引入先进的元学习或开集识别机制,增强模型的自适应能力。本文相信,这些探索将进一步提升模型在复杂多变、持续演进的真实

网络环境中的性能与适用性。

参考文献

[1] Wazid M, Das A K, Shetty S, et al. Security in 5G-enabled internet of things communication: Issues, challenges, and future research roadmap[J]. IEEE Access, 2021, 9: 4466-4489.

[2] Buczak A L, Guven E. A survey of data mining and machine learning methods for cyber security intrusion detection[J]. IEEE Communications Surveys & Tutorials, 2016, 18(2): 1153-1176.

[3] Zeng Yi, Gu Huaxi, Wei Wenting, et al. Deep-Full-Range: A deep learning based network encrypted traffic classification and intrusion detection framework[J]. IEEE Access, 2019, 7: 45182-45190.

[4] Bekerman D, Shapira B, Rokach L, et al. Unknown malware detection using network traffic classification[C]// 2015 IEEE Conference on Communications and Network Security (CNS). Piscataway: IEEE, 2015: 134-142.

[5] Xuan C D. Detecting APT attacks based on network traffic using machine learning[J]. Journal of Web Engineering, 2021, 20(1): 171-190.

[6] Azab A, Khasawneh M, Alrabaee S, et al. Network traffic classification: Techniques, datasets, and challenges[J]. Digital Communications and Networks, 2024, 10(3): 676-692.

[7] 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.

[8] Van Ede T, Bortolameotti R, Continella A, et al. FlowPrint: Semi-supervised mobile-app fingerprinting on encrypted network traffic[C]//27th Annual Network and Distributed System Security Symposium (NDSS). Reston: The Internet Society, 2020: 24412.

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

[10] Shen Meng, Zhang Jinpeng, Zhu Liehuang, et al. Accurate decentralized application identification via encrypted traffic analysis using graph neural networks[J]. IEEE Transactions on Information Forensics and Security, 2021, 16: 2367-2380.

[11] Zhang Haozhen, Yu Le, Xiao Xi, et al. TFE-GNN: A temporal fusion encoder using graph neural networks for fine-grained encrypted traffic classification[C]//Proceedings of the ACM Web Conference 2023. New York: ACM, 2023: 2066-2075.

[12] Guo Chaoqun, Wang Nan, Sun Yuanlin, et al. DTC: Addressing the long-tailed problem in intrusion detection through the divide-then-conquer paradigm[C]//2023 IEEE 29th International Conference on Parallel and Distributed Systems (ICPADS). Piscataway: IEEE, 2023: 1319-1326.

[13] Gui Jie, Chen Tuo, Zhang Jing, et al. A survey on self-supervised learning: Algorithms, applications, and future trends[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, 46(12): 9052-9071.

[14] He Hongye, Yang Zhiguo, Chen Xiangning. Payload encoding representation from transformer for encrypted traffic classification[J]. ZTE Communications, 2021, 19(4): 90-97.

[15] 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.

[16] Meng Xuying, Lin Chungang, Wang Yequan, et al. NetGPT: Generative pretrained transformer for network traffic[PP/OL]. V2. arXiv (2023-05-17)[2025-09-01]. https://arxiv.org/abs/2304.09513v2.

[17] 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.

[18] 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.

[19] 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: 1-19. https://openreview.net/forum?id=B1ckMDqlg.

[20] 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): 5232-5270.

[21] Jiang A Q, Sablayrolles A, Roux A, et al. Mixtral of experts[PP/OL]. V1. arXiv (2024-01-08)[2025-09-01]. https:

//arxiv.org/abs/2401.04088.

[22] 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: Curran Associates Inc., 2021: 657.

[23] Wei Guanglu, Wang Zhonghua. Adoption and realization of deep learning in network traffic anomaly detection device design[J]. Soft Computing, 2021, 25(2): 1147-1158.

[24] Moore A W, Papagiannaki K. Toward the accurate identification of network applications[C]//6th International Workshop on Passive and Active Network Measurement. Berlin: Springer, 2005: 41-54.

[25] Saber A, Fergani B, Abbas M. Encrypted traffic classification: Combining over-and under-sampling through a PCA-SVM[C]//2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS). Piscataway: IEEE, 2018: 8598480.

[26] Liu Ya, Wang Xiao, Qu Bo, et al. ATVITSC: A novel encrypted traffic classification method based on deep learning[J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 9374-9389.

[27] Yang Zhe, Ma Zitong, Zhao Wenbo, et al. HRNN: Hypergraph recurrent neural network for network intrusion detection[J]. Journal of Grid Computing, 2024, 22(2): 52.

[28] 赵文博, 马紫彤, 杨哲. 基于超图神经网络的恶意流量分类模型[J]. 网络与信息安全学报, 2023, 9(5): 166-177.
Zhao Wenbo, Ma Zitong, Yang Zhe. Model of the malicious traffic classification based on hypergraph neural network[J]. Chinese Journal of Network and Information Security, 2023, 9(5): 166-177. (in Chinese)

[29] Wu Yonghui, Schuster M, Chen Zhifeng, et al. Google's neural machine translation system: Bridging the gap between human and machine translation[EB/OL]. V2.arXiv (2016-10-08)[2025-09-01]. https://arxiv.org/abs/1609.08144v2.

[30] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. New York: Curran Associates Inc., 2017: 6000-6010.

[31] Zhang Biao, Sennrich R. Root mean square layer normalization[C]//Proceedings of the 33rd International Conference on Neural Information Processing Systems. New York: Curran Associates Inc., 2019: 1110.

[32] Su Jianlin, Ahmed M, Lu Yu, et al. RoFormer: Enhanced transformer with rotary position embedding[J]. Neurocomputing, 2024, 568: 127063.

[33] Neto E C P, Dadkhah S, Ferreira R, et al. CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment[J]. Sensors, 2023, 23(13): 5941.

[34] Dadkhah S, Neto E C P, Ferreira R, et al. CICIoMT2024: A benchmark dataset for multi-protocol security assessment in IoMT[J]. Internet of Things, 2024, 28: 101351.

[35] 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.

[36] 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 (ICISSP 2016). Setúbal: SciTePress, 2016: 407-414.

[37] Moustafa N, Slay J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set) [C]//2015 Military Communications and Information Systems Conference (MilCIS). Piscataway: IEEE, 2015: 7348942.

[38] Lashkari A H, Draper-Gil G, Mamun M S I, et al. Characterization of tor traffic using time based features[C]//Proceedings of the 3rd International Conference on Information Systems Security and Privacy. Setúbal: SciTePress, 2017: 253-262.

[39] Loshchilov I, Hutter F. Decoupled weight decay regularization[C/OL]//Proceedings of the 7th International Conference on Learning Representations, 2019: 1-18. https://openreview.net/forum?id=Bkg6RiCqY7.

[40] Dao T, Fu D Y, Ermon S, et al. FlashAttention: Fast and memory-efficient exact attention with IO-awareness[C]//Proceedings of the 36th International Conference on Neural Information Processing Systems. New York: Curran Associates Inc., 2022: 068431-1189.

作者简介

周嘉俊 男,1995 年 12 月出生于浙江省杭州市。现为浙江工业大学网络空间安全研究院特聘副研究员。主要研究方向为互联网安全、大模型安全、图机器学习。

E-mail: jjzhou@zjut.edu.cn

何美静 女,2004 年 11 月出生于陕西省西安市。现为浙江工业大学网络空间安全研究院硕士研究生。主要研究方向为网络流量分析、恶意流量检测。

E-mail: mjhe@zjut.edu.cn

孙长辉 男,2002 年 4 月出生于江西省赣州市。现为浙江工业大学网络空间安全研究院硕士研究生。主要研究方向为网络流量分析、恶意流量检测。

E-mail: sunchanghui@zjut.edu.cn

俞山青 女,1984 年 2 月出生于浙江省杭州市。现为浙江工业大学网络空间安全研究院副教授。主要研究方向为互联网安全、推荐系统安全。

E-mail: yushanqing@zjut.edu.cn