引用格式：尹鑫宇，施凡，许成喜，等. 基于分组级流量语义特征增强的 LLM IoT 设备识别方法[J]. 信息对抗技术，2025，4(5)：22-41. [YIN Xinyu，SHI Fan，XU Chengxi，et al. IoT device identification method enhanced by packet-level traffic semantic features for large language models[J]. Information Countermeasure Technology，2025，4(5)：22-41.(in Chinese)]

基于分组级流量语义特征增强的 LLM IoT 设备识别方法

尹鑫宇，施　凡*，许成喜，章建成，葛明仪

（国防科技大学电子对抗学院，安徽合肥 230037）

摘　要　随着物联网(IoT)技术在各领域的快速普及，网络设备识别已经成为网络安全防护体系的关键环节，实时发现接入网络的 IoT 设备对于网络管理、安全防护和性能优化至关重要。准确掌握网络动态并识别这些 IoT 设备，是有效防御黑客攻击的必要前提。传统机器学习的识别方法不仅效率低下、特征选取复杂、环境迁移能力差，其准确率也难以满足实际防护需求。为此，提出了一种基于分组级流量语义特征增强的大语言模型(LLM)IoT 设备识别方法。首先，将复杂异构的 IoT 流量转化为通用的分组级流量语义特征；然后，使用分组级流量语义特征微调 LLM，使 LLM 能够自动学习潜在 IoT 设备流量特征并执行设备分类识别决策，从而实现端到端高效的 IoT 设备识别。在公开数据集 Aalto、UNSW 和混合 CIC IoT 数据集(2022、2023)上的实验结果表明，所提方法能够基于分组级流量语义特征有效识别 IoT 设备，并且该方法的平均识别准确率分别达到 99.99%、99.42%、98.83%。

关键词　物联网；设备识别；大语言模型；分组级流量语义特征

中图分类号　TP 393.8　　**文章编号**　2097-163X(2025)05-0022-20

文献标志码　A　　**DOI**　10.12399/j.issn.2097-163x.2025.05.002

IoT device identification method enhanced by packet-level traffic semantic features for large language models

YIN Xinyu，SHI Fan*，XU Chengxi，ZHANG Jiancheng，GE Mingyi

（College of Electronic Engineering，National University of Defense Technology，Hefei 230037，China）

Abstract　With the rapid popularization of Internet of Things (IoT) technology in various fields，network device identification has become a key link in the network security protection system. Real-time detection of IoT devices accessing the network is crucial for network management，security protection，and performance optimization. Accurately understanding network dynamics and identifying these IoT devices is a necessary prerequisite for effectively defending against hacker attacks. Traditional machine learning-based identification methods not only suffer from low efficiency，complex feature selection，and poor environmental transferability，but their accuracy also fails to meet the needs of practical protection. To address this issue，an IoT device identification method based on packet-level traffic semantic feature-enhanced large language models (LLM) was proposed. First，complex and heterogeneous

收稿日期：2025-07-03　　**修回日期**：2025-09-09

通信作者：施凡，E-mail：shifan17@nudt.edu.cn

基金项目：国家重点研发计划项目(2021YFB3100500)

IoT traffic was converted into universal packet-level traffic semantic features. Then, these packet-level traffic semantic features were used to fine-tune the LLM, enabling the LLM to automatically learn the potential traffic features of IoT devices and make device classification and identification decisions, thereby realizing end-to-end and efficient IoT device identification. Experimental results on the public datasets Aalto, UNSW, and hybrid CIC IoT datasets (2022, 2023) show that the proposed method can effectively identify IoT devices based on packet-level traffic semantic features, and its the average identification accuracy can reach 99.99%, 99.42%, and 98.83% respectively.

Keywords　IoT; device identification; LLM; packet-level traffic semantic features

0　引言

物联网(Internet of Things, IoT)技术的快速发展和广泛应用正在深刻改变人类的生产生活方式。当前，各类智能终端设备已全面应用在消费电子、家居安防、工业控制等关键领域，形成规模庞大的设备生态。据市场研究机构预测，到2026 年全球 IoT 设备接入规模将突破 800 亿台，相关市场规模预计攀升至 1.1 万亿美元[1-2]。这一迅猛增长态势在加速产业数字化转型的同时，也带来了显著的安全隐患。IoT 基础设施一般由计算资源和能源受限的物联设备组成，基于 IoT 的应用普遍采用轻量级通信协议。IoT 设备数量的激增直接导致了网络攻击面的急剧扩大。研究显示[3]，IoT 设备普遍存在的安全防护不足、网络服务漏洞以及更新机制缺陷等问题，使得 IoT 系统面临日益严峻的安全威胁。其中，超过 33% 的 IoT 设备存在未修复的已知漏洞，攻击者通过利用存在漏洞的 IoT 设备对有价值的节点进行入侵攻击。Mirai 僵尸网络攻击事件是 IoT 安全领域的典型威胁案例[4]。该恶意软件通过利用 IoT 设备的默认凭证漏洞，构建了大规模的僵尸网络，并针对关键网络基础设施发起分布式拒绝服务攻击，攻击导致包括 Twitter、Netflix 等在内的多个主流网站服务中断，造成了严重的经济损失。

从网络管理者角度出发，IoT 安全防护的首要环节在于设备识别。运营商需要掌握完善的设备信息，以便快速筛选并隔离存在安全隐患的 IoT 终端设备，同时对存在漏洞的设备实施安全加固。准确识别设备类型不仅为网络安全策略的实施提供依据，更为后续的设备配置与功能管理奠定了重要基础。由此可见，高精度的 IoT 设备类型识别技术具有重要的研究价值和应用意义。

当前主流的 IoT 设备识别方法主要是通过提取网络流量特征而后基于机器学习模型的分类识别来进行设备分类。这类方法虽然能够帮助网络管理员监控设备运行状态，但是仍然存在局限性：一是现有方法提取流量特征(数据分组级、数据流级)对特定环境过拟合，导致所提取的特征没有较强的环境迁移能力[5]。二是现有基于数据流特征方法虽然有较强的泛化能力但需要积累大量的数据分组以达到必要的流量规模阈值，这种分析模式存在较高的延迟[6]。三是现有的方法无法很好适配复杂多变的 IoT 环境，当网络中出现新的 IoT 设备时，无法快速适应环境变化，大幅增加了模型部署难度。

为了解决上述问题，本文提出了一种基于分组级流量语义特征增强的大语言模型(large language model, LLM) IoT 设备识别方法，该方法采用 LLM 作为基础架构，通过低秩自适应(low-rank adaptation, LoRA)微调(fine-tuning)技术实现高效的设备分类。具体而言，系统首先将原始网络流量转化为分组级流量语义特征；然后利用 LoRA 方法[7]对 LLM 进行参数高效的微调训练，使其能够准确理解分组级语义流量特征；在推理阶段，新采集流量数据经过相同特征转换后，可直接输入微调后的模型完成设备类型分类。本文的主要贡献如下：

1) 提出一种基于分组级流量语义特征增强的 LLM IoT 设备识别方法，将异构流量转化为通用语义特征，利用 LLM 自动学习关键特征，突破传统人工特征通用性差的局限，支持加密流量识

别，实现特征提取与分类的自动化。

2）该方法具备小样本高效学习能力，仅需约 4.4%的训练数据即可快速收敛，在测试集上的准确率达到了 97.82%，显著降低计算资源消耗。

3）在 Aalto、UNSW、混合 CIC IoT 等公开数据集上开展实验，结果表明模型具有优异的跨场景泛化能力，可适应多样化网络环境下的设备识别需求。

1 相关工作

随着 IoT 设备在智能家居、工业控制和医疗监护等领域的广泛应用，其安全性问题变得愈发突出。由于许多 IoT 设备受限于硬件资源且固件更新滞后，这些设备中存在大量已知及未知的安全漏洞，容易成为攻击者入侵内部网络的入口。因此，高效识别局域网中的 IoT 设备类型已成为网络安全防护体系中的关键环节之一[8]。准确识别设备类型有助于网络管理员迅速隔离异常或易受攻击的设备，防止攻击者利用这些设备进行横向渗透和长期潜伏。

基于机器学习的 IoT 设备识别方法广泛采用网络流量统计特征（如包大小、时延、协议类型）结合分类器（如随机森林、SVM 等）进行分类[9-11]。RAHMAN 等[12]构建了包含 105 种设备、超 2 亿条流量的大规模真实数据集，实验表明随机森林仅用 12 个特征即可实现高效分类。

为应对上述挑战，研究者开始采用深度学习实现自动特征学习，如用卷积神经网络（convolutional neural network，CNN）[13]建模原始流量，或用循环神经网络（recurrent neural network，RNN）捕捉时序依赖以提升精度[14]。然而，这类方法仍依赖大量标注数据，且模型结构固定，难以适应动态变化的设备行为。近年来，预训练语言模型（如 BERT、ALBERT）被广泛应用于各类任务。文献[15]首次提出基于预训练 Transformer 的 IoT 设备识别方法 IoTBERT，通过在无标签流量上预训练 ALBERT 学习特征表示，再结合标注数据微调并引入残差网络实现端到端识别。相比传统方法，IoTBERT 无需人工特征工程，但跨场景泛化能力仍有局限。

此外，文献[16]表明，LLM 使用链路级特征对 IoT 设备分类具备可行性。在 Wi-Fi、Zigbee 和蓝牙等多类设备上，该方法展现出良好的适应性：小样本下通过提示调优即可有效识别，大规模数据下微调后性能更具竞争力，验证了大模型在 IoT 设备识别方面的应用潜力。

2 研究方法

2.1 IoT 设备识别方法框架

本文提出的基于分组级流量语义特征增强的 LLM IoT 设备识别方法的使用场景如图 1 所示。该技术方案可使网络管理员实时掌握接入设备的类型及状态信息，有效识别未授权设备接入或异常行为，进而及时采取安全防护措施，保障智能环境系统的整体安全性。

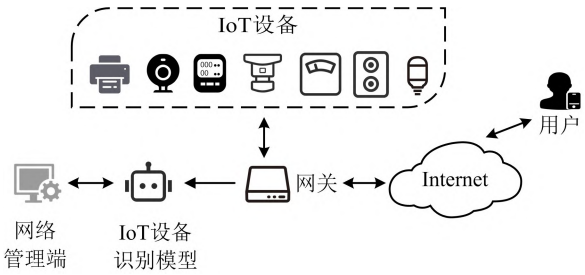


图 1 IoT 设备识别模型的使用场景

Fig. 1 The application scenarios of the IoT device identification model

该 IoT 设备识别方法采用多阶段处理流程（如图 2 所示）：首先对原始 IoT 流量进行解析，提取分组级语义信息（包括协议头部字段和载荷特征）；然后通过专用分词器将这些语义信息编码为数值化的词向量表示，形成 LLM 可理解的分组级流量语义向量特征；在模型训练阶段，基于 LoRA 方法对这些语义特征进行高效微调，优化 LLM 的参数；最终在推理阶段，将待识别设备的流量语义特征输入微调后的 LLM，通过模型自动学习的关键特征完成设备类型的分类识别。该流程在保证处理效率的同时，显著提升了模型对异构 IoT 设备的识别准确率。本文还拓展了模型的持续学习能力，通过引入增量学习策略，在仅需少量内存和训练时间的条件下，能够有效保留模型在原有任务中学习到的先验知识，同时显著提升对新加入设备的识别准确率，从而缓解模型在持续学习过程中面临的遗忘问题。

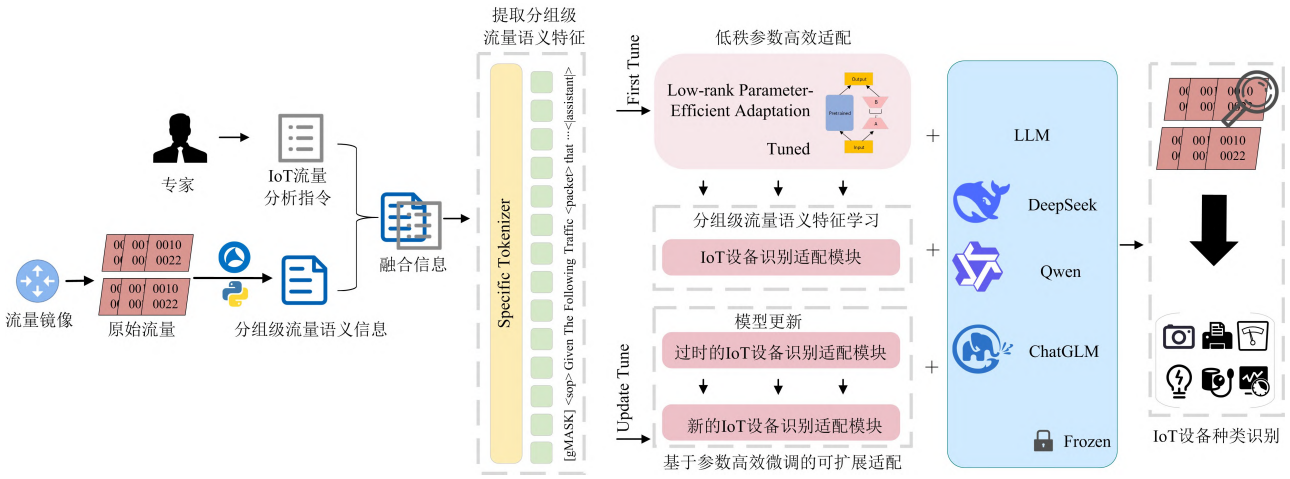


图 2　本文方法框架

Fig. 2　The framework of the method in this article

2.2　面向 LLM 输入的分组级流量语义信息提取方法

本文提出了一种面向 LLM 输入的分组级流量语义信息提取方法，其核心目标是从原始网络流量中抽取出具有通用语义价值的分组级特征，并与 IoT 分析指令进行有机融合，最终转化为适合 LLM 处理的标准化输入格式，见算法 1 所示。

算法 1　基于原始流量的分组级语义信息提取算法

输入：原始网络流量 pcap 文件

输出：标准格式的消息字典列表

1. MessageList ← ∅
2. 使用 Tshark 过滤 IoT 流量（TCP/UDP 协议），移除 MAC 和 IP 地址信息，得到数据包集合 $D_1, D_2, \cdots, D_n$
3. Fields ←["frame. encap_type","frame. time",...,"tcp. payload","udp. payload"]
4. for 每个数据包 D_i do
5. packetSemanticDict ← ∅
6. if D_i. has_layer("TCP") then
7. packetSemanticDict ← ExtractPacketFeatures (Fields,"TCP",D_i)
8. payloadField ← "tcp. payload"
9. else if D_i has_layer("UDP") then
10. packetSemanticDict ← ExtractPacketFeatures (Fields,"UDP",D_i)
11. payloadField ← "udp. payload"
12. end if
13. if packetSemanticDict[payloadField] ≠ ∅ then
14. 截取载荷前 1 000 字节：packetSemanticDict [payloadField] = packetSemanticDict[payloadField][0:1 000]
15. 转换为文本格式：textSample ← ConvertToText (packetSemanticDict)
16. /* 构建消息字典 */
17. messageDict ← ∅
18. messageDict["messages"] ←[{"role" : "system" , "content" : "[设备类型分类任务] +[设备类别列表]"}, {"role" : "user" , "content" : "<packet>:" + textSample}, {"role" : "assistant","content":"[待标注设备类别]"} ⇨训练时填充真实标签，推理预测时填充"—"。]
19. MessageList ← MessageList ∪ {messageDict}
20. end if
21. end for
22. return MessageList

具体实现流程分为 4 个关键步骤：第 1 步进行协议层过滤，首先基于传输层协议对数据进行筛选，仅保留包含 TCP 或 UDP 层的有效数据包，以排除不相关或低语义价值的数据干扰。第 2 步为有效载荷筛选，旨在进一步剔除不包含应用层有效载荷的数据包，如 TCP/UDP 空数据包（如纯 ACK 包等）。该步骤有助于聚焦承载实际通信内容的数据流，从而提升所提取语义信息的相关性与实用性。第 3 步通过预定义的分组级流量语义特征字段集合（如图 3 所示），将原始二进制流量数据转换为结构化的文本表示形式。在此过程中，移除了引入噪声的 MAC 地址和 IP 地址相关信息，并将应用层有效载荷转换为十六进制编码，以保留其原始语义信息的同时增强通用性。该特征提取方式对明文与加密流量均具适用性，尤其适用于无法解析高层协议语义信息的加密通信场景。

物理层字段
frame. encap_type, frame. time, frame. offset_shift, frame. time_epoch, frame. time_delta, frame. time_relative, frame. number, frame. len, frame. marked, frame. protocols
数据链路层字段
eth. dst_oui, eth. dst_oui_resolved, eth. dst. lg, eth. dst. ig, eth. src_oui, eth. src_oui_resolved, eth. src. lg, eth. src. ig, eth. type
网络层字段
ip. version, ip. hdr_len, ip. dsfield, ip. dsfield. dscp, ip. dsfield. ecn, ip. len, ip. id, ip. flags, ip. flags. rb, ip. flags. df, ip. flags. mf, ip. frag_offset, ip. ttl, ip. proto, ip. checksum, ip. checksum. status, ipv6. version, ipv6. tclass, ipv6. plen, ipv6. llim, ipv6. nxt, esp. spi, esp. seq
传输层(TCP) 字段
tcp. srcport, tcp. dstport, tcp. stream, tcp. completeness, tcp. len, tcp. seq, tcp. nxtseq, tcp. ack, tcp. hdr_len, tcp. flags, tcp. flags. res, tcp. flags. ns, tcp. flags. cwr, tcp. flags. ecn1, tcp. flags. urg, tcp. flags. ack, tcp. flags. push, tcp. flags. reset, tcp. flags. syn, tcp. flags. fin, tcp. flags. str, tcp. window_size, tcp. window_size. scalefactor, tcp. checksum, tcp. checksum. status, tcp. urgent_pointer, tcp. time_relative, tcp. time_delta, tcp. analysis. bytes_in_flight, tcp. analysis. push_bytes_sent, tcp. segment, tcp. segment. count, tcp. reassembled. length, tcp. payload
传输层(UDP) 字段
udp. srcport, udp. dstport, udp. stream, udp. length, udp. checksum, udp. checksum. status, udp. time_relative, udp. time_delta, udp. payload, udpcp. reassembled. length
传输层安全(TLS/DTLS) 字段
tls. record. content_type, tls. record. version, tls. record. length, dtls. record. content_type, dtls. record. version, dtls. record. length

图 3 分组级流量语义特征字段分类

Fig. 3 Group-level traffic semantic feature field classification

第 4 步,格式标准化,如图 4 所示,生成符合 LLM 输入要求的标准化格式。在模型训练、推理过程中,本文采用基于专家知识的 instruction 构建训练、训练数据模板,将 LLM 有效适配到流量语义空间。设计的 instruction 模板能够智能引导模型自动提取流量数据中的关键模式特征。具体实现上,本文采用 Tshark 工具[17]结合自定义 Python 脚本提取各协议层的结构化字段信息,通过数据清洗过滤干扰性 MAC 地址和 IP 地址字段,并将其规范化为“字段名:值”的键值对形式。同时,定义特殊标识符〈packet〉作为流量数据起始标记,最终构建包含分析指令与结构化流量数据的训练样本。该格式既保留了原始流量的关键语义特征,又实现了与自然语言指令之间的无缝衔接。该通用表示方法被进一步应用于 IoT 设备识别等下游任务以及模型微调过程中,展现出良好的适应性和泛化能力。

相较于传统方法,本方法采用端到端的学习策略,直接从原始流量中提取训练数据以学习通用流量语义表征,从而获得优异的跨场景泛化性能。

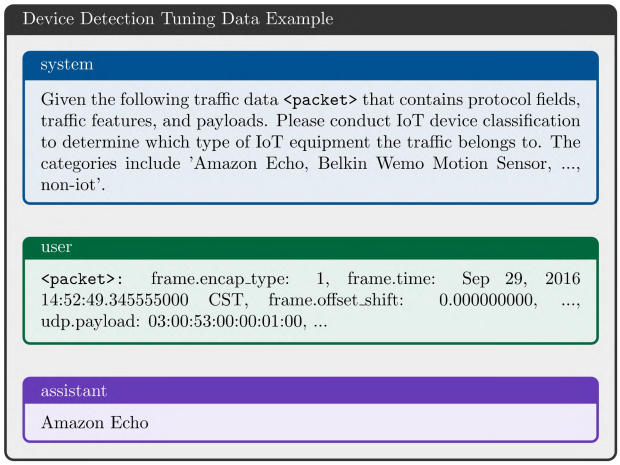


图 4 设备检测调优数据示例

Fig. 4 Device detection tuning data example

2.3 分组级流量语义信息分词器优化

本文提出了一种分组级流量语义信息分词器优化方法,该方法的核心创新点在于构建了专用的流量分词器,有效扩展了 LLM 原生分词器的语义处理能力,使 LLM 能够准确理解异构流量数据并构建有效的语义表征。

基于大规模训练数据的统计分析，我们提取了所有协议字段名（分组级流量语义特征字段，如图 3 所示）和高频特殊字符（如 \u00b7、〈packet〉、0x 等）作为特殊 tokens 扩展 LLM 的分词器。由于 LLM 原生分词器缺乏对异构 IoT 网络流量数据的先验知识，该扩展显著提升了模型对流量语义的理解能力。实验采用 ChatGLM4[18] 作为基础模型，见表 1 所列。

表 1　特定分词器与默认分词器分词对比

Tab. 1　Comparison of word segmentation between a specific word segitter and the default word segitter

默认分词结果 (Token length：763.25)	特定分词结果 (Token length：687.38)
'<','packet','>:','_frame','.enc','ap_type',':','_','1',',','_frame','.time',':','_0ct','_','_','9',',','_','2016','_','20',':','54',':','45','.','5','96','03','8','000','_CST',',',',','_frame','.offset_shift',':','_','0','.','000','000','000',',',',…[更多分词内容]	'<packet>',':','_',',frame.encap_type',':','_','1',',',',','_',',frame.time',':','_0ct','_',',','_',',9',',',',','_',',2016',',_',',20',':',',54',':',',45','.','5',',96',',03',',8',',000',',_CST',',',',',',_',',frame.offset_shift',':',',_',',0',',.',',000',',000',',',',…[更多分词内容]

改进后的分词器在流量数据处理上展现出显著优势：添加特殊 token 的分词器将流量数据的平均拆分长度从 763.25 降低至 687.38，完整保留了字段名称和特殊字符的结构信息，而默认分词器则会导致这些关键语义单元被不合理拆分。定量分析表明，改进后的分词器能够更精确地建模流量语义信息，不仅提升了模型的理解能力，还显著减少了训练时长。特别地，在小样本学习场景下，该表征方法使大模型展现出更优异的特征提取和理解能力。

2.4　基于参数高效微调的可拓展性适配

为实现 LLM 基于分组级流量语义特征的 IoT 设备分类，并增强其在动态 IoT 环境中的快速适应能力，本文提出了一种新的模型优化方法。该方法的核心在于充分利用 LLM 固有的模式挖掘和泛化能力，通过 Transformer 架构强大的记忆特性，从通用流量语义信息（如数据包长度、传输方向、协议标志位等）中自动学习具有高度区分性的关键流量特征。

本文以 ChatGLM4-9B 作为基础模型架构，采用基于 LoRA 的微调策略，使 LLM 能够有效获取 IoT 领域的专业知识。这种方法特别适用于复杂异构的流量环境下的 IoT 设备精准识别任务。在技术实现上，我们针对传统训练方法存在的局限性进行了重要改进。

假设预训练 LLM 的参数为 θ_{LLM}，传统全参数训练方法需要更新全部模型参数（即 $|\Delta\theta| = |\theta_{\mathrm{LLM}}|$）。这种方案存在 2 个显著缺陷：一是 LLM 庞大的参数量会导致新环境适应过程中的计算成本过高；二是全参数更新可能引发灾难性遗忘，导致模型丧失基础语言理解能力。

为解决这些问题，本文采用 LoRA 微调技术，实现参数高效微调（parameter-efficient fine-tuning）具体包括：一是冻结 LLM 的主干参数 θ_{LLM}，保持其基础语言理解能力不变；二是引入低秩适配旁路结构，通过矩阵分解技术实现参数高效更新；三是专门构建面向 IoT 流量特征学习的适配参数模块 θ_{now}。这种微调策略不仅大幅降低了计算资源需求，更重要的是保留了模型原有的语言理解能力，同时实现了对 IoT 流量特征的精准捕捉。实验证明，该方法在保持模型基础性能的前提下，显著提升了 IoT 设备分类的准确率和环境适应效率。

给定安全专家的自然语言指令 $I_{\mathrm{instruction}}$ 以及流量数据 $x \in X = \{x_0, x_1, \cdots, x_n\}$（包含 n 个流量标记），流量检测任务要求将 $I_{\mathrm{instruction}}$ 和 x 作为 LLM 输入。随后，LLM 利用合成的 IoT 流量特征学习的参数 θ 识别得到设备分类结果 $y \in \{y_1, y_2, \cdots, y_m\}$（包含 m 个不同类型的 IoT 设备）。其中：

$$\theta = \theta_{\mathrm{LLM}} + \theta_{\mathrm{now}} \tag{1}$$

$$y = \mathrm{LLM}(I_{\mathrm{instruction}}, x, \theta) \tag{2}$$

式中，θ_{LLM} 为预训练大语言模型的原始参数，在微调过程中保持冻结，以保留其通用语义理解能力；θ_{now} 为通过低秩自适应引入的可训练参数，用于高效适配网络流量语义特征，避免全参数微调带来的计算开销。

LLM 微调采用如下损失函数。其中，X 为

分组级流量语义序列，即 $X=\{x_1,x_2,\cdots,x_n\}$，每个 x_i 表示输入序列中第 i 个 token。模型通过自回归方式学习序列的上下文依赖关系。具体表达式如下：

$$P(x_i \mid x_1,\cdots,x_{i-1})=\mathrm{softmax}(\boldsymbol{W}_l h_{i-1}) \quad (3)$$

$$J(\theta)=-\sum_i \ln P(x_i \mid x_1,\cdots,x_{i-1}) \quad (4)$$

式中，$\boldsymbol{W}_l$ 是用于流量包级语义信息表示学习的可训练参数矩阵；h_{i-1} 表示 LLM 在输入前 $i-1$ 个 token 后所编码的隐藏状态，反映了当前流量序列的历史上下文信息。通过逐个 token 预测的方式，使 LLM 能够有效捕捉流量数据中的时序特征与语义结构，从而实现对网络流量的通用语义表示学习。

针对 IoT 环境中设备频繁接入和通信协议多样化的情况，使用了一种基于 LoRA 轻量级适配 IoT 流量动态变化方案，使 LLM 能够快速适应不断变化的网络环境，更新适配参数模块为 θ_{new}。如图 5 所示，当面对新型设备或协议时，本文的方法仅需微调模型 0.029 6% 的参数，即可实现高效的领域适应。实验表明，该方案显著降低了模型适配成本，具体表现为：GPU 显存消耗减少 80%，训练时间缩短 50%。相比传统的全参数微调方法，本方案有效解决了 IoT 场景中模型重新训练带来的高计算成本问题，为实际部署中的动态流量识别任务提供了可行的轻量化解决方案。

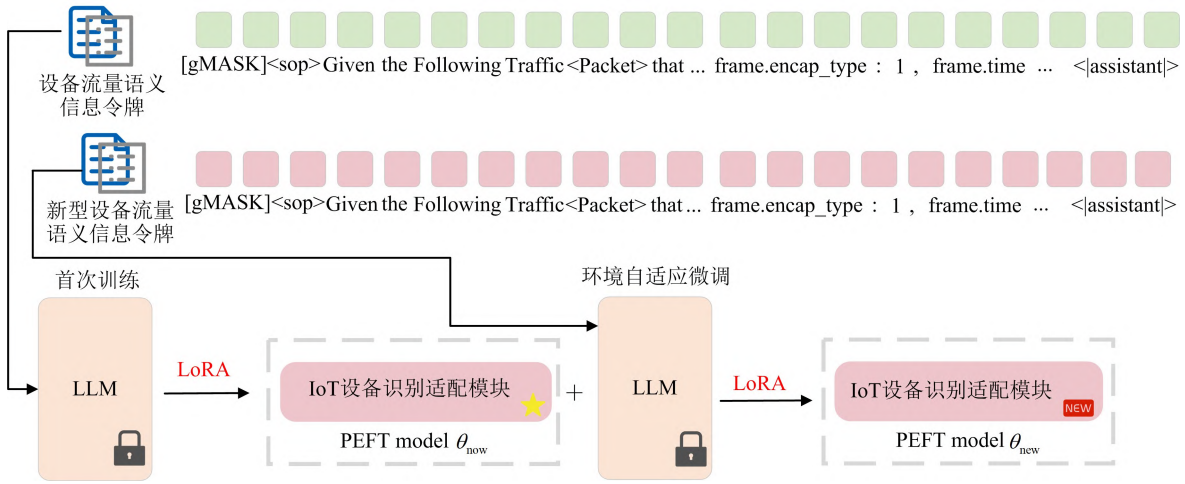


图 5 基于参数高效微调的可拓展性适配

Fig. 5 Scalable adaptation based on efficient fine-tuning of parameters

为实现模型在动态网络环境下的持续适应能力，本文引入一个可迭代的参数更新函数 Update(·)，用于在线优化轻量级适配模块。该机制旨在根据新观测到的流量模式和安全指令反馈，动态调整模型对物联网设备特征的表征能力。具体表达式如下：

$$\theta_{\text{new}}=\mathrm{Update}(\theta_{\text{LLM}},\theta_{\text{now}}) \quad (5)$$

$$\hat{\theta}=\theta_{\text{LLM}}+\theta_{\text{new}} \quad (6)$$

$$\hat{y}=\mathrm{LLM}(I_{\text{instruction}},x,\hat{\theta}) \quad (7)$$

式中，θ_{new} 表示更新后的适配参数模块；$\hat{\theta}$ 表示融合了最新适配能力的合成参数，用于学习和识别更新后的物联网流量特征；$\hat{y}$ 表示更新后模型识别出的设备类型集合，包含 k 个类别，即 $\hat{y}\in\{y_1,y_2,\cdots,y_k\}$。

3 论证实验

3.1 实验数据源

本文选用了公开的 Aalto 数据集[19]，涵盖了 31 种常见的智能设备类别，包括智能照明、摄像头、家用电器以及健康监测设备等。其中大多数测试设备通过 Wi-Fi 或以太网接入用户的网络；部分设备则通过 ZigBee 或 Z-Wave 等 IoT 协议，借助以太网或 Wi-Fi 集线器设备间接连接网络。Aalto 数据集中包含了一定比例的加密通信流量，尤其是在涉及用户隐私和敏感数据传输的应用场景下，如智能家居控制指令和健康监测数据。

此外，本文采用公开的 UNSW 数据集[20]，该数据集涵盖设备自主通信及用户交互产生的动

态流量，包含丰富的 IoT 协议与应用层内容，具有良好的真实性和代表性。数据集共包含 21 种 IoT 设备（如智能摄像头、传感器、健康监测仪等）和 9 种非 IoT 设备（如平板电脑、笔记本电脑和智能手机）。本文统一把非 IoT 设备归类为“non-IoT”这一单一类别，以构建更具区分度的分类任务。值得注意的是，该数据集中存在大量基于 TLS/SSL 等协议的加密流量，能够有效支持模型在加密环境下的设备识别能力评估。

本文还采用了加拿大网络安全研究所（CIC）发布的 CIC IoT 2022[21] 和 CIC IoT 2023 数据集[22]。数据涵盖了 59 种常见 IoT 设备（如智能音箱、智能插座等）在空闲与活动状态下的网络流量行为。本文将上述数据集中空闲与活动状态的正常流量进行整合，构建了一个大规模的良性流量数据集，用于模型的验证与泛化能力评估。值得注意的是，该数据集包含大量加密通信流量，尤其在涉及隐私敏感的设备（如智能音箱、摄像头）中，广泛采用 TLS/DTLS 协议进行传输层安全保护，进一步增强了模型在真实加密环境下的适用性验证。

3.2　数据预处理

本文实验中，微调所使用的数据是通过使用 Tshark 工具结合自定义的 Python 脚本，按照 MAC 地址对原始流量进行划分后获取的。为进一步避免数据分组中协议数据单元（PDU）头部字段（如 IP 地址或 MAC 地址）对模型训练造成干扰，在预处理阶段删除了 IP 地址和以太网地址字段。

随后，利用 Tshark 工具提取各协议层的结构化字段信息，并通过编写 Python 脚本对数据进行清洗，过滤掉无关或干扰性的 MAC 地址与 IP 地址，最终将有效字段规范化为“字段名：值”的键值对形式。对于载荷部分，将原始的二进制数据转换为十六进制表示，并在每 2 个十六进制数据之间插入冒号“：”作为分隔符[23]。最后在数据集中抽取数据样本，数据集数量分布如图 6～8 所示。在微调阶段，将数据集按照 6∶2∶2 的比例划分为训练集、验证集和测试集，以用于模型的参数调整和方法评估。

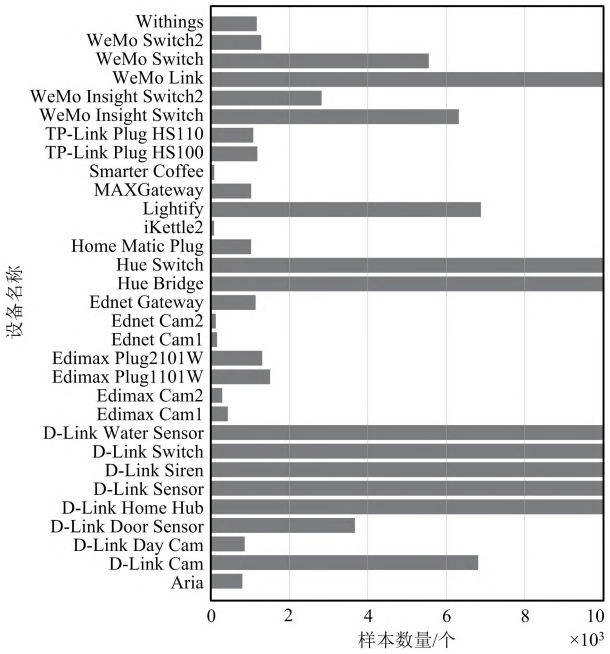


图 6　Aalto 数据集分布

Fig. 6　Aalto dataset distribution

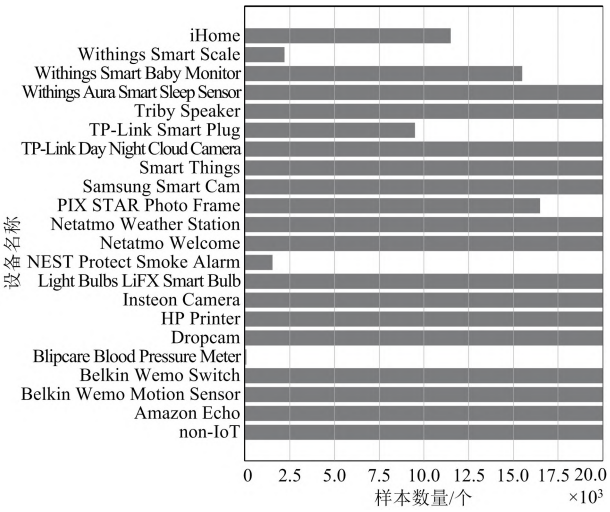


图 7　UNSW 数据集分布

Fig. 7　UNSW dataset distribution

3.3　实验环境及评估指标

3.3.1　实验环境及参数设置

本文实验在 Python 3.10.16 和 PyTorch 2.6.0（GPU 版本）的深度学习框架下进行，硬件平台采用 Intel(R) Xeon(R) Gold 6 330 CPU @ 2.00 GHz 处理器和 NVIDIA A800 80 GB PCIe Tensor Core GPU 加速器，操作系统为 Ubuntu 22.04 LTS。本实验是基于 GLM4-9B 预训练模型架构进行 LoRA 微调，在模型微调参数的选择上，设置训练轮数为 3，初始学习率为 5×10^{-5}，优化器选用 Adam，采用 BF16 混合精度训练策略，启用 16 线程并行数据预加载，配置单卡批处理大

小为 4，设置秩(rank)$r=8$、缩放因子 $\alpha=32$，对 Transformer 架构中的查询、键、值投影层进行适配，并引入 0.1 的 LoRA dropout 率以增强模型泛化能力；考虑到 IoT 设备流量的语义特征与序列长度分布，输入序列最大长度设为 2 048 个 token，输出序列最大长度限定为 32 个 token。训练完成后生成的模型具备良好的部署灵活性，可在配备 RTX 4 090 或 RTX 3 090 等用户级 GPU 的设备上高效运行，适用于边缘侧或本地化大模型应用场景。

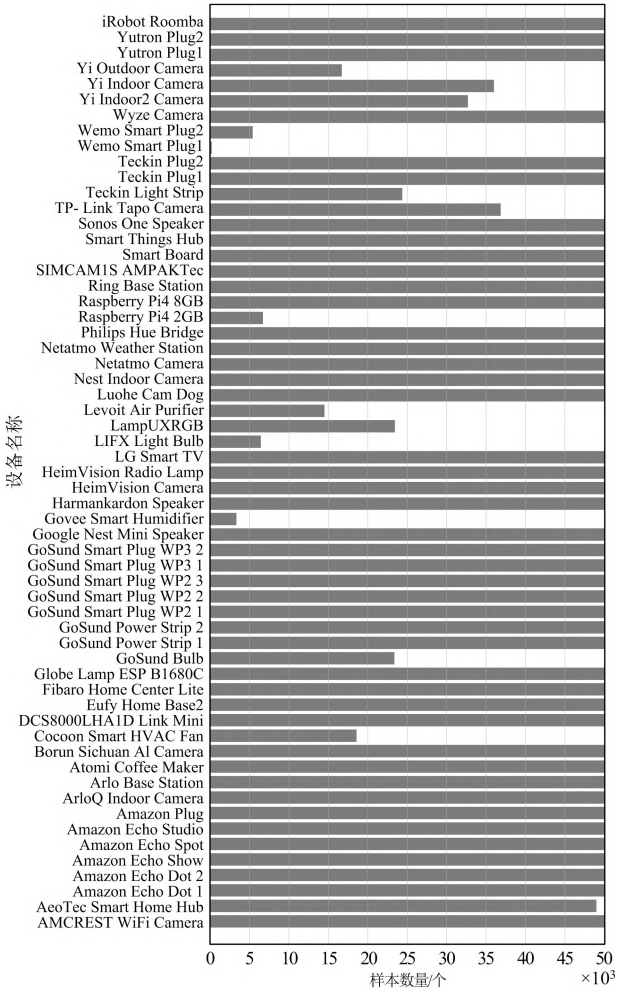


图 8 CIC IoT 数据集分布

Fig. 8 CIC IoT dataset distribution

3.3.2 评估指标

本文采用多维度评估指标体系对模型性能进行全面衡量，具体包括准确率(accuracy)、精确率(precision)、召回率(recall)和 F1 值(F1-score) 4 个核心指标。为消除类别不平衡带来的评估偏差，采用各类别指标的平均值作为最终评价标准。各指标定义如下：

1) 准确率。反映模型整体预测准确度，表示为：

$$A_{\text{accuracy}}=\frac{T_{\text{TP}}+T_{\text{TN}}}{T_{\text{TP}}+F_{\text{FP}}+T_{\text{TN}}+F_{\text{FN}}} \tag{8}$$

式中，T_{TP} 表示预测正确的阳性样本数，T_{TN} 表示预测正确的阴性样本数，F_{FP} 表示预测错误的阳性样本数，F_{FN} 表示预测错误的阴性样本数。

2) 精确率。衡量模型预测正例的可靠性，表示为：

$$P_{\text{precision}}=\frac{T_{\text{TP}}}{T_{\text{TP}}+F_{\text{FP}}} \tag{9}$$

3) 召回率。评估模型识别正例的全面性，表示为：

$$R_{\text{recall}}=\frac{T_{\text{TP}}}{T_{\text{TP}}+F_{\text{FN}}} \tag{10}$$

4) F1 值。综合精确率和召回率的调和平均值，表示为：

$$F_{\text{1-score}}=\frac{2T_{\text{TP}}}{2T_{\text{TP}}+F_{\text{FN}}+F_{\text{FP}}} \tag{11}$$

3.4 实验结果及分析

3.4.1 超参数设置对模型效率与特征表达的影响

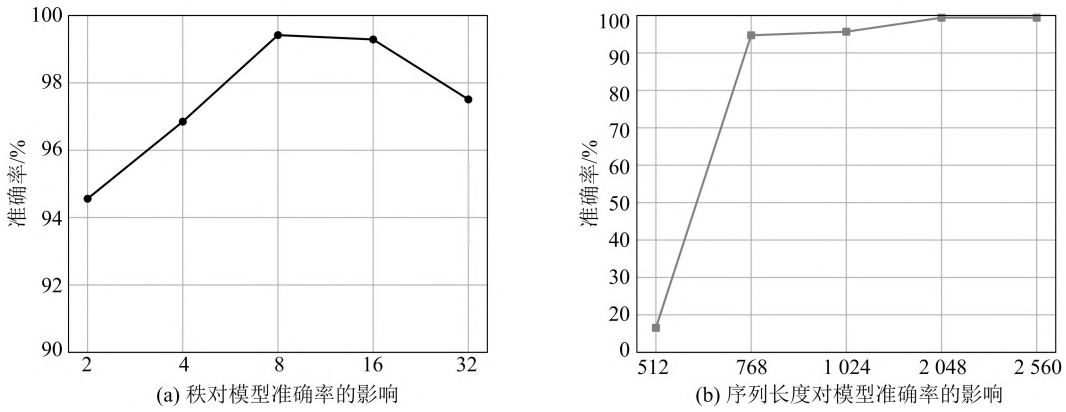
输入序列长度和秩是影响 LLM 性能的关键超参数。GLM4-9B 模型支持的最大输入序列长度为 128 000。当输入序列超过该长度时，需进行截断处理；若短于该长度，则使用〈|endoftext|〉标记进行填充以达到统一长度。序列长度的设置对模型性能具有显著影响：过长的序列会引入大量〈|endoftext|〉标记，增加不必要的计算开销并可能干扰模型注意力机制；过短的序列则可能导致 IoT 流量数据中的关键上下文信息丢失，影响特征表达的完整性。

对于秩而言，在 LoRA 等参数高效微调方法中，秩的大小直接关系到模型的拟合能力：秩过大时，模型参数增多，容易导致过拟合，尤其在训练数据有限的情况下泛化性能下降；秩过小时，模型的表达能力受限，难以捕捉复杂的特征模式，导致学习不足。因此，合理选择序列长度与秩的取值对于平衡模型效率与性能至关重要，需结合具体任务和数据特性进行优化，如图 9 所示。

3.4.2 在 Aalto、UNSW 与 CIC IoT 数据集上的识别性能验证

测试数据基于以上指标进行评估。Aalto 测试集上的平均准确率、召回率以及 F1 值指标分别达到 99.99%、99.99%、0.999 9；UNSW 测试集上的平均准确率、召回率以及 F1 值指标分别达到

99.42%、99.41%、0.994 1；CIC IoT 混合测试数据集上的平均准确率、召回率以及 F1 值指标分别达到 98.83%、98.83%、0.988 4。表 2～4 列出了模型在公共数据集上对每个 IoT 设备的识别结果。



(a) 秩对模型准确率的影响　　(b) 序列长度对模型准确率的影响

图 9　序列长度与秩值对模型准确率影响的对比分析

Fig. 9　A comparative analysis of the influence of sequence length and rank value on model accuracy

表 2　Aalto 数据分类识别结果

Tab. 2　Aalto data classification and recognition results

设备名称	精确率/%	召回率/%	F1 值
Aria	99.21	99.21	0.992 1
D-Link Sensor	99.96	100	0.999 8
Edimax Cam2	100	100	1.000 0
Ednet Gateway	100	100	1.000 0
Lightify	100	100	1.000 0
WeMo Insight Switch2	100	100	1.000 0
WeMo Insight Switch	100	100	1.000 0
Withings	100	100	1.000 0
D-Link Cam	100	100	1.000 0
D-Link Siren	100	100	1.000 0
Edimax Plug 1101W	100	100	1.000 0
Home Matic Plug	100	100	1.000 0
MAXGateway	100	100	1.000 0
Smarter Coffee	100	100	1.000 0
D-Link DayCam	100	100	1.000 0
D-Link Switch	100	100	1.000 0
Edimax Plug 2101W	100	100	1.000 0
Hue Bridge	100	99.97	0.999 8
TP-Link Plug HS100	100	100	1.000 0
WeMo Link	100	100	1.000 0
D-Link Door Sensor	100	100	1.000 0
D-Link Water Sensor	100	100	1.000 0
Ednet Cam1	100	100	1.000 0
Hue Switch	100	100	1.000 0
TP-Link Plug HS110	100	100	1.000 0
WeMo Switch2	100	100	1.000 0
D-Link Home Hub	100	100	1.000 0
Edimax Cam1	100	100	1.000 0
Ednet Cam2	100	100	1.000 0
iKettle2	100	100	1.000 0
WeMo Switch	100	100	1.000 0

表 3　UNSW 数据分类识别结果

Tab. 3　UNSW data classification and identification results

设备名称	精确率/%	召回率/%	F1 值
non-IoT	99.35	99.12	0.992 4
Amazon Echo	95.33	100	0.976 1
Belkin Wemo Motion Sensor	98.87	98.93	0.989 0
Blipcare Blood Pressure Meter	100	100	1.000 0
Dropcam	100	100	1.000 0
HP Printer	100	100	1.000 0
iHome	97.67	100	0.988 2
Insteon Camera	100	96.15	0.980 4
Light Bulbs LiFX Smart Bulb	99.97	100	0.999 8
Belkin Wemo Switch	98.93	98.83	0.988 8
NEST Protect Smoke Alarm	100	100	1.000 0
Netatmo Weather Station	99.97	99.97	0.999 7
Netatmo Welcome	100	99.87	0.999 3
PIX STAR Photo Frame	100	100	1.000 0
Samsung Smart Cam	99.40	99.97	0.997 7
Smart Things	100	99.97	0.999 8
TP-Link Day Night Cloud Camera	100	99.40	0.997 0
TP-Link Smart Plug	100	98.98	0.994 9
Triby Speaker	100	100	1.000 0
Withings Aura Smart Sleep Sensor	100	99.12	0.995 6
Withings Smart Baby Monitor	99.40	100	0.997 0
Withings Smart Scale	99.12	100	0.995 6

表 4 CIC IoT 混合数据集分类识别结果

Tab. 4 Classification and recognition results of CIC IoT hybrid datasets

设备名称	精确率/%	召回率/%	F1 值	设备名称	精确率/%	召回率/%	F1 值
AMCREST WiFi Camera	95.66	100	0.977 8	LG Smart TV	100	99.80	0.999 0
AeoTec Smart Home Hub	94.50	95.00	0.947 5	LIFX Light Bulb	100	99.87	0.999 4
Amazon Echo Dot 1	100	99.73	0.998 6	LampUXRGB	100	99.92	0.999 6
Amazon Echo Dot 2	99.67	99.53	0.996 0	Levoit Air Purifier	100	100	1.000 0
Amazon Echo Show	100	99.27	0.996 3	Luohe Cam Dog	100	100	1.000 0
Amazon Echo Spot	100	99.40	0.997 0	Nest Indoor Camera	100	100	1.000 0
Amazon Echo Studio	100	99.87	0.999 4	Netatmo Camera	99.80	99.93	0.998 7
Amazon Plug	100	99.70	0.998 5	Netatmo Weather Station	100	99.73	0.998 6
Arlo Q Indoor Camera	99.87	100	0.999 4	Philips Hue Bridge	100	99.27	0.996 3
Arlo Base Station	100	99.20	0.996 0	Raspberry Pi4 2GB	97.59	99.59	0.985 8
Atomi Coffee Maker	100	99.60	0.998 0	Raspberry Pi4 8GB	100	99.53	0.997 6
Borun Sichuan AI Camera	100	99.93	0.999 7	Ring Base Station	100	100	1.000 0
Cocoon Smart HVAC Fan	100	96.84	0.983 9	SIMCAM1S AMPAKTec	99.06	98.07	0.985 6
DCS8000LHA1D Link Mini	100	100	1.000 0	Smart Board	100	100	1.000 0
Eufy Home Base2	100	100	1.000 0	Smart Things Hub	95.50	94.87	0.951 8
Fibaro Home Center Lite	100	99.93	0.999 7	Sonos One Speaker	100	99.73	0.998 6
Globe Lamp ESP B1680C	99.80	99.73	0.997 7	TP-Link Tapo Camera	100	99.40	0.997 0
GoSund Bulb	99.92	100	0.999 6	Teckin Light Strip	100	99.93	0.999 7
GoSund Power Strip 1	99.87	99.20	0.995 3	Teckin Plug1	85.04	90.93	0.878 9
GoSund Power Strip 2	99.60	99.53	0.995 7	Teckin Plug2	89.99	97.73	0.937 0
GoSund Smart Plug WP2 1	99.60	99.60	0.996 0	WeMo Smart Plug1	100	100	1.000 0
GoSund Smart Plug WP2 2	99.40	99.53	0.994 7	WeMo Smart Plug2	100	99.66	0.998 3
GoSund Smart Plug WP2 3	99.53	98.27	0.989 0	Wyze Camera	100	100	1.000 0
GoSund Smart Plug WP3 1	100	99.93	0.999 7	Yi Indoor2 Camera	100	98.23	0.991 1
GoSund Smart Plug WP3 2	100	100	1.000 0	Yi Indoor Camera	99.01	100	0.995 0
Google Nest Mini Speaker	100	100	1.000 0	Yi Outdoor Camera	100	94.51	0.971 8
Govee Smart Humidifier	100	100	1.000 0	Yutron Plug1	87.80	93.60	0.906 1
Harmankardon Speaker	100	96.33	0.981 3	Yutron Plug2	98.77	90.73	0.945 8
Heim Vision Camera	100	100	1.000 0	iRobot Roomba	100	100	1.000 0
Heim Vision Radio Lamp	99.93	99.40	0.996 6				

从实验结果可以看出，本文微调的 LLM 对大多数设备的识别精确率都很高，能达到 95%以上。在 Aalto 数据集上的测试结果表明，该模型展现出卓越的 IoT 设备分类能力。测试结果显示，31 种设备中有 28 种实现了 100%的完美识别率，其精确率、召回率达到 100%，F1 值达到 1.000 0。其余设备的识别性能同样出色。

在 UNSW 数据集上的测试结果表明，该模型展现出优异的设备分类能力，能够准确区分 IoT 设备与非 IoT 设备。模型对非 IoT 设备的识别精确率高达 99.35%，剩下 IoT 设备的精确率均为 95%以上，其中 6 种设备实现了精确率、召回率为 100%，F1 值为 1.000 0 的完美识别率。总体而言，该模型在 UNSW 数据集上表现出了强大的分类性能和高准确性。

在 CIC IoT 混合数据集上的实验结果表明，该模型在复杂网络环境（包括设备的空闲状态与交互状态）以及跨时间段场景下展现出良好的鲁棒性与适应性。模型在所有 IoT 设备类别上的平均准确率超过 98%，其中 13 种设备实现了精确率、召回率为 100%，F1 值为 1.000 0 的完美识别率，验证了其在多样化流量条件下的稳定分类性能。此外，实验结果进一步表明，该模型不仅能够准确区分不同类型的 IoT 设备，还具备对同类型设备不同硬件版本（如 Amazon Echo Dot 1 与 Echo Dot 2、Teckin Plug1 与 Plug2 等）进行细

粒度识别的能力。这一特性表明模型能够捕捉设备间细微的通信行为差异，具备较强的特征表达能力。综上所述，该模型在真实复杂网络环境中展现出高效且精确的设备识别能力，具有较强的实用价值与部署潜力。

图 10～12 展示了分类模型在测试集上的预测混淆矩阵分析结果（按行归一化，反映召回率）。实验结果表明，尽管原始数据集中存在一定程度的噪声干扰和冗余流量，导致极少数样本被误分类，但这些误差对模型整体的识别准确率影响较小。总体来看，模型仍展现出优异的分类性能，具有较高的识别精度和良好的泛化能力。

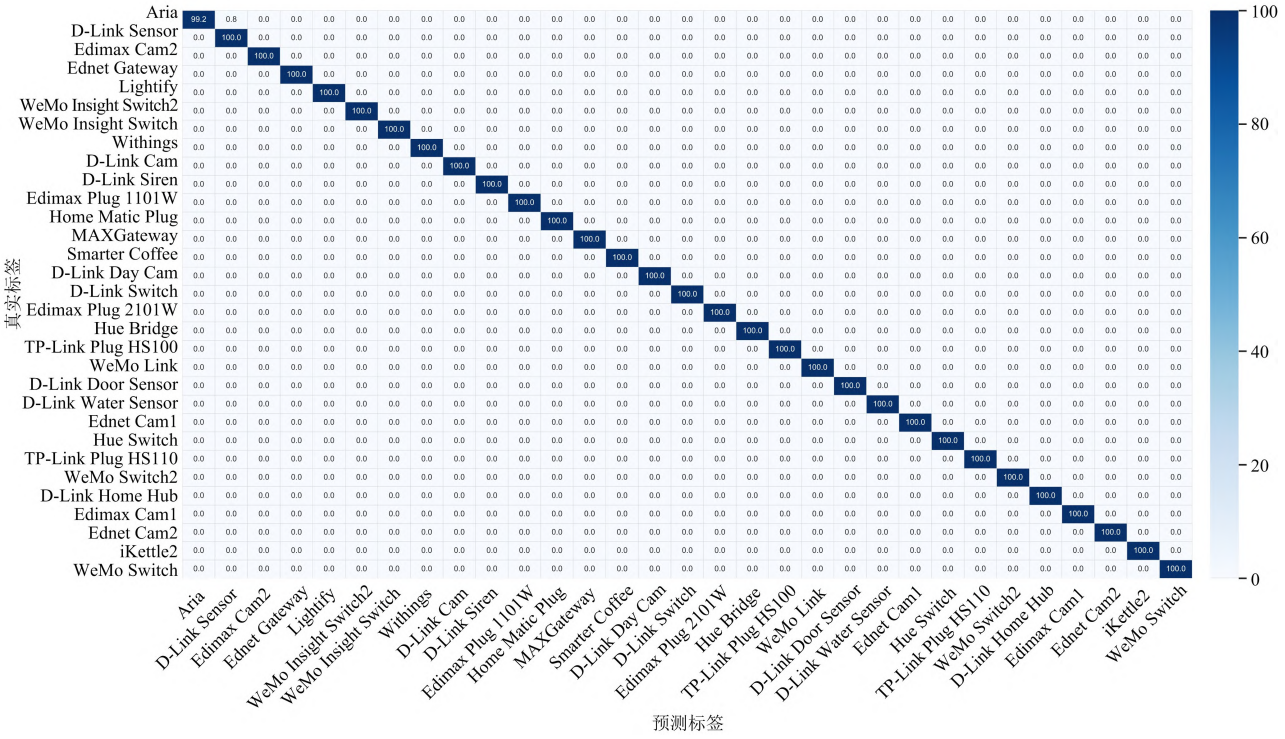


图 10　本文模型分类混淆矩阵（Aalto）

Fig. 10　The model classification confusion matrix（Aalto） in this article

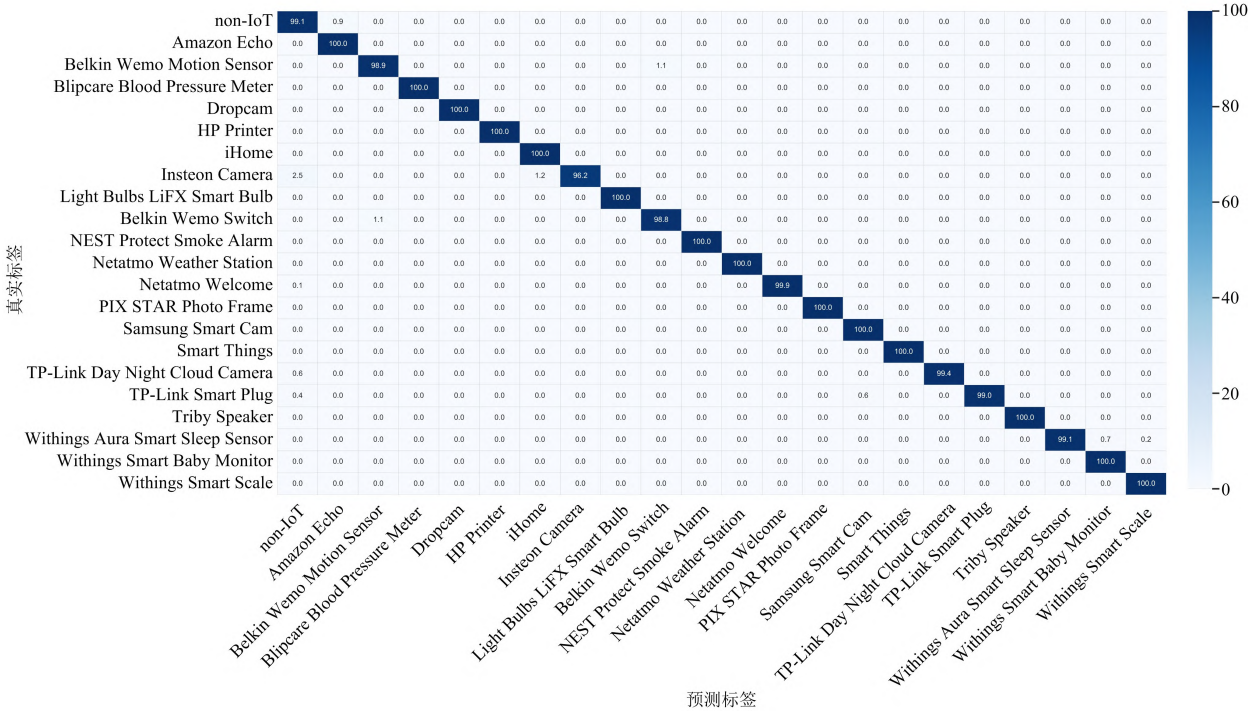


图 11　本文模型分类混淆矩阵（UNSW）

Fig. 11　The model classification confusion matrix（UNSW） in this paper

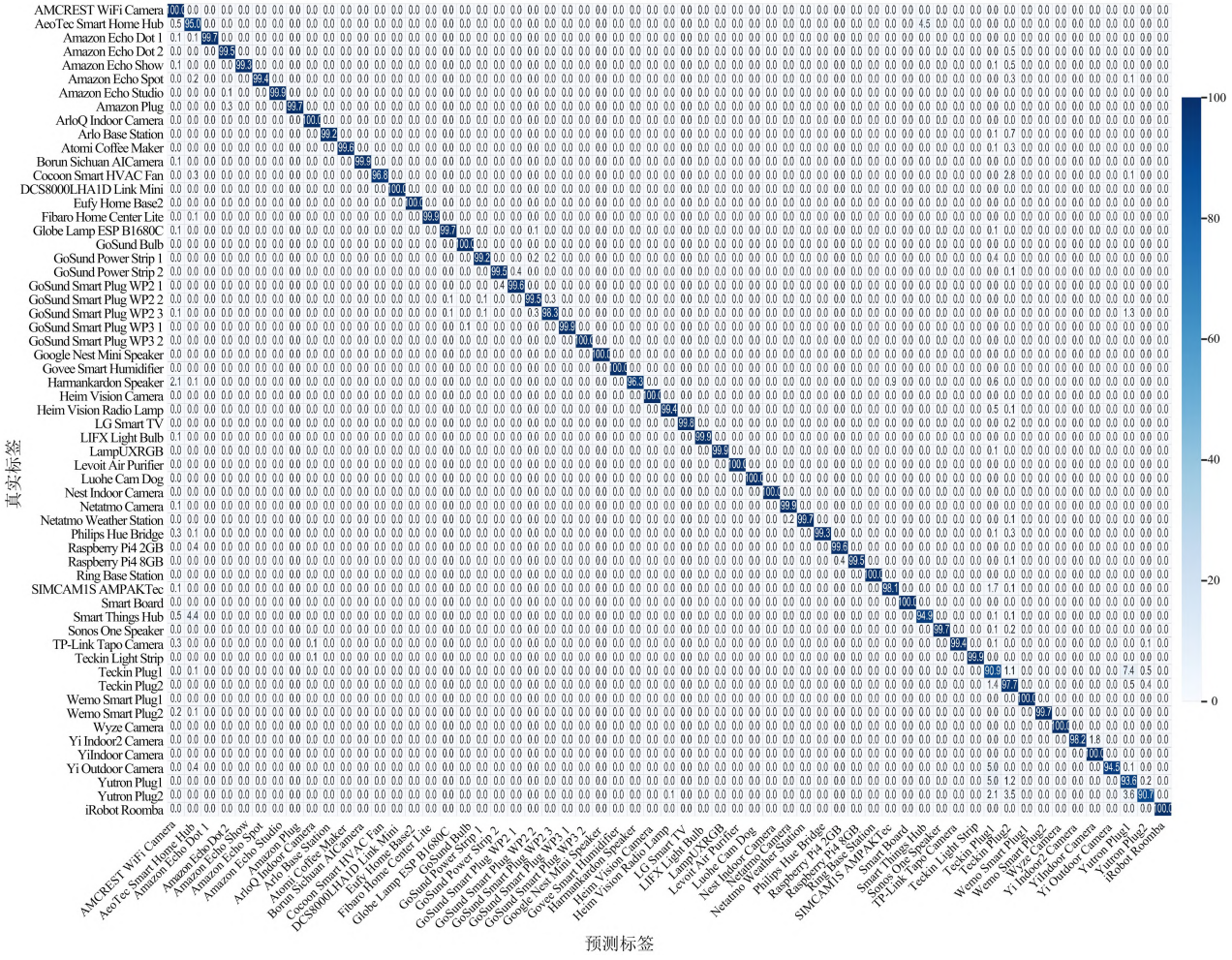


图 12 本文模型分类混淆矩阵(CIC IoT)

Fig. 12 The model classification confusion matrix(CIC IoT) in this paper

3.4.3 小样本场景下的识别性能验证

在流量过小的场景下无法提取相应的统计特征,或提取到的特征不足以完成识别任务,本文随机选取了 UNSW 数据集中的 8 000 条流量对模型进行微调。其平均准确率、召回率以及 F1 值指标分别达到了 97.82%、97.82%、0.978 0。小样本数据集(UNSW)中每个 IoT 设备的识别结果见表 5 所列,分类模型在测试集上的预测混淆矩阵(按行归一化,反映召回率)分析结果如图 13 所示。

表 5 小样本数据分类结果(UNSW)

Tab. 5 Small sample data classification results(UNSW)

设备名称	精确率/%	召回率/%	F1 值	设备名称	精确率/%	召回率/%	F1 值
non-IoT	96.19	99.34	0.977 4	Netatmo Weather Station	99.93	95.00	0.974 0
Amazon Echo	98.04	100	0.990 1	Netatmo Welcome	95.24	99.93	0.975 3
Belkin Wemo Motion Sensor	91.52	97.87	0.945 9	PIX STAR Photo Frame	100	100	1.000 0
Blipcare Blood Pressure Meter	91.67	100	0.956 5	Samsung Smart Cam	97.94	100	0.989 6
Dropcam	100	99.84	0.999 2	Smart Things	100	99.63	0.998 2
HP Printer	100	99.94	0.999 7	TP-Link Day Night Cloud Camera	99.33	98.43	0.988 8
iHome	99.87	93.27	0.964 6	TP-Link Smart Plug	98.99	99.84	0.994 1
Insteon Camera	100	97.54	0.987 6	Triby Speaker	100	84.13	0.913 8
Light Bulbs LiFX Smart Bulb	99.90	100	0.999 5	Withings Aura Smart Sleep Sensor	98.93	95.75	0.973 2
Belkin Wemo Switch	97.58	91.23	0.943 0	Withings Smart Baby Monitor	97.41	99.23	0.983 2
NEST Protect Smoke Alarm	100	100	1.000 0	Withings Smart Scale	93.87	97.34	0.955 7

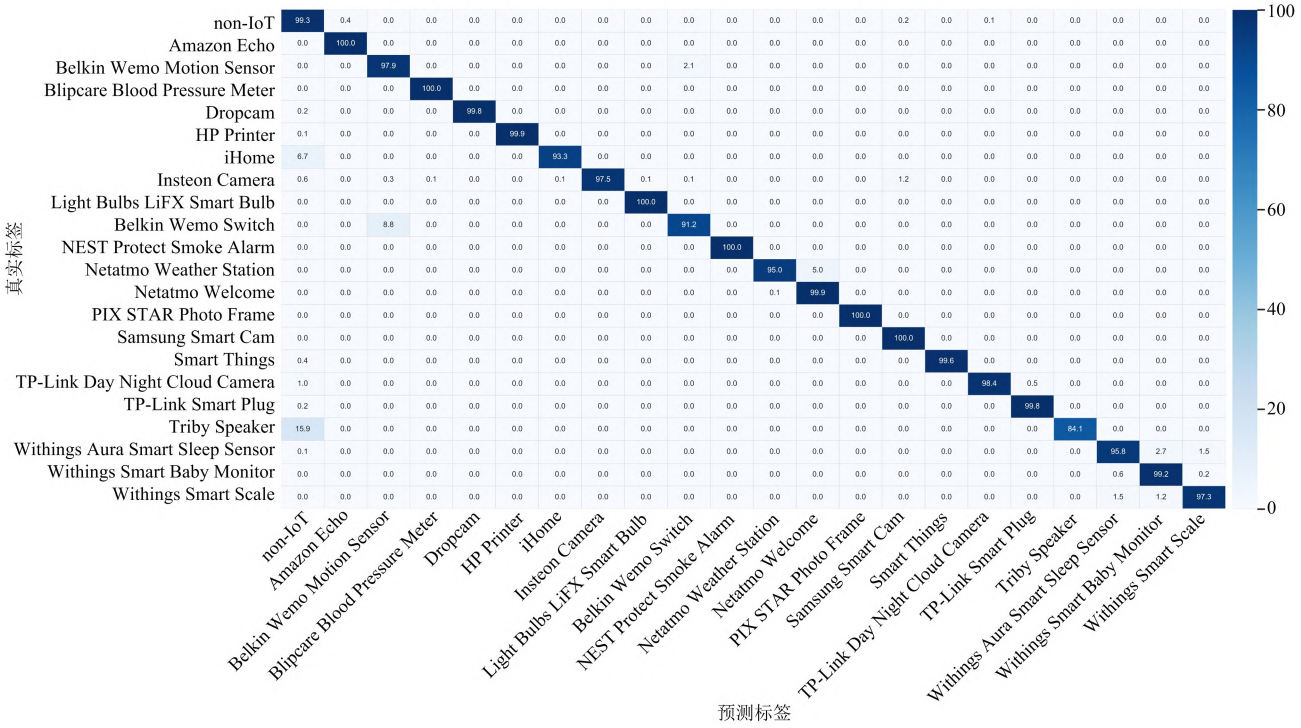


图 13　本文模型分类混淆矩阵(小样本)

Fig. 13　Confusion matrix of model classification in this paper(small sample)

在 UNSW 实验场景中，本文引入了一类实际部署中常见的 IoT 设备 D-Link Door Sensor，并采集其网络流量数据，基于原有数据集训练的模型，采用参数高效微调(PEFT)技术对新增设备进行快速适配，具体配置包括：使用 GLM4-9B 架构，在 BF16 混合精度下以学习率 3×10^{-5}、batch size 为 1、cosine 学习率调度和 10 步 warmup 进行 1 轮训练，输入序列最大长度为 2 048 个 token，输出序列最大长度为 32 个 token，并通过 LoRA($r=8$，$\alpha=32$，$d_{\text{dropout}}=0.1$)对查询-键-值投影层进行轻量化微调，从而实现对新环境的高效可扩展适应。

模型在包含新增设备的整体测试集上取得了 99.10%的总体识别准确率。环境变化数据的精确率、召回率及 F1 值分类实验结果见表 6 所列。对于原有环境中的各类设备，识别精确率均保持在 94%以上，展现出良好的泛化能力与稳定性。针对新增的 D-Link Door Sensor 设备，模型实现了对其 100%的识别精确率，召回率达到 98.18%，F1 值为 0.990 8。结果表明，该模型不仅能够高效识别原有设备，还能迅速适应新型设备的特征，在实际应用场景中，具备出色的可扩展性与实用性。

表 6　环境变化数据分类结果

Tab. 6　Classification results of environmental change data

设备名称	精确率/%	召回率/%	F1 值
non-IoT	98.15	94.79	0.964 4
Amazon Echo	100	100	1.000 0
Belkin Wemo Motion Sensor	98.74	98.93	0.988 3
Blipcare Blood Pressure Meter	100	100	1.000 0
Dropcam	99.93	100	0.999 7
HP Printer	100	93.09	0.964 2
iHome	100	99.82	0.999 1
Insteon Camera	99.80	97.02	0.983 9
Light Bulbs LiFX Smart Bulb	100	100	1.000 0
Belkin Wemo Switch	98.93	98.80	0.988 7
NEST Protect Smoke Alarm	100	99.78	0.998 9
Netatmo Weather Station	99.93	99.77	0.998 5
Netatmo Welcome	98.97	99.53	0.992 5
PIX STAR Photo Frame	100	100	1.000 0

续表

设备名称	精确率/%	召回率/%	F1 值
Samsung Smart Cam	94.79	99.97	0.973 1
Smart Things	100	100	1.000 0
TP-Link Day Night Cloud Camera	99.76	99.00	0.993 8
TP-Link Smart Plug	100	97.74	0.988 6
Triby Speaker	99.93	99.97	0.999 5
Withings Aura Smart Sleep Sensor	99.92	99.69	0.998 1
Withings Smart Baby Monitor	99.60	99.97	0.997 8
Withings Smart Scale	100	100	1.000 0
D-Link Door Sensor	100	98.18	0.990 8

3.4.4 跨数据集下的识别泛化能力评估

跨数据集场景下的 IoT 设备识别泛化性能实验，为评估模型的跨数据集泛化能力，本文在 CIC IoT 混合数据集上将 Amazon Echo 系列的多个子型号（如 Echo Dot、Echo Show、Echo Studio 等）聚合为统一类别“Amazon Echo”。同时，选取在 2 个数据集中表示相同或高度相似 IoT 设备类型的设备，包括 Borun Sichuan AI Camera、Fibaro Home Center Lite、LIFX Light Bulb、Philips Hue Bridge 和 TP-Link Tapo Camera，用于测试在 UNSW 数据集上训练的模型对这些设备的识别性能。

实验结果（见表 7 所列）表明，模型在 CIC-IoT 数据集中的“Amazon Echo”类别上取得了优异表现，精确率、召回率和 F1 值分别达到 100%、99.89%、0.999 4。此外，对其他设备的识别效果同样出色，Borun Sichuan AI Camera 的精确率、召回率和 F1 值分别为 99.93%、100%、0.999 7；Fibaro Home Center Lite 的分别为 100%、99.87%、0.999 3；LIFX Light Bulb 的分别达到 100%、100%、1.000 0；Philips Hue Bridge 的分别达到 99.80%、99.87%、0.998 3；TP-Link Tapo Camera 的分别达到 100%、81.87%、0.900 3。上述结果充分验证了该模型在不同数据分布下的强鲁棒性及在实际场景中的应用潜力。

3.4.5 加密环境下的识别性能验证

在加密环境下对 IoT 设备识别模型的性能评估实验中，提取了 UNSW 数据集中的所有加密流量数据，用于测试模型在加密场景下的识别能力。实验结果（见表 8 所列）表明，模型取得了 99.94%的准确率、99.94%的召回率以及 0.999 1 的 F1 值。这表明，即使在完全加密的网络环境中，模型仍能保持极高的识别性能。

表 7 IoT 设备类型在跨数据集环境下的分类性能结果

Tab. 7 The classification performance results of IoT device types in a cross-dataset environment

设备名称	精确率/%	召回率/%	F1 值
Amazon Echo	100	99.89	0.999 4
Borun Sichuan AI Camera	99.93	100	0.999 7
Fibaro Home Center Lite	100	99.87	0.999 3
LIFX Light Bulb	100	100	1.000 0
Philips Hue Bridge	99.80	99.87	0.998 3
TP-Link Tapo Camera	100	81.87	0.900 3

表 8 IoT 设备类型在加密环境下的分类性能结果

Tab. 8 The classification performance results of IoT device types in an encrypted environment

设备名称	精确率/%	召回率/%	F1 值
non-IoT	99.95	99.98	0.999 7
Amazon Echo	100	100	1.000 0
Belkin Wemo Motion Sensor	96.79	100	0.983 7
Blipcare Blood Pressure Meter	100	100	1.000 0
Dropcam	100	100	1.000 0
HP Printer	100	100	1.000 0
iHome	100	100	1.000 0
Insteon Camera	99.41	100	0.997 0
Light Bulbs LiFX Smart Bulb	100	100	1.000 0
Netatmo Welcome	99.67	100	0.998 4
PIX STAR Photo Frame	100	100	1.000 0
Smart Things	100	100	1.000 0
TP Link Day Night Cloud Camera	100	100	1.000 0
TP Link Smart Plug	100	100	1.000 0
Triby Speaker	100	100	1.000 0

3.4.6　面向设备伪装攻击的识别性能验证

在真实网络环境中，攻击者可能通过伪造 IP 地址、MAC 地址甚至 OUI（组织唯一标识符）来伪装设备身份。为评估模型在此类对抗场景下的鲁棒性，在 UNSW 数据集上对 OUI 字段进行随机化处理，模拟设备硬件标识的恶意伪造行为，并在此基础上进行训练与测试。实验结果（见表 9 所列）表明，模型在该伪装场景下仍取得了 98.46% 的准确率、98.46% 的召回率和 0.984 5 的 F1 值，展现出对伪造标识信息的良好抵抗能力与较强的现实适应性。

表 9　IoT 设备类型在硬件标识伪造环境下的分类性能评估

Tab. 9　The classification performance evaluation of IoT device types in a hardware identifier forgery environment

设备名称	精确率/%	召回率/%	F1 值
non-IoT	98.06	98.45	0.982 5
Amazon Echo	95.23	99.20	0.971 8
Belkin Wemo Motion Sensor	97.19	94.57	0.958 6
Blipcare Blood PressureMeter	100	100	1.000 0
Dropcam	99.87	98.44	0.991 5
HP Printer	97.78	99.72	0.987 4
iHome	100	96.63	0.982 9
Insteon Camera	100	95.11	0.975 0
Light Bulbs LiFX Smart Bulb	99.97	99.13	0.995 5
Belkin Wemo Switch	95.10	97.07	0.960 7
NEST Protect Smoke Alarm	94.04	98.44	0.961 9
Netatmo Weather Station	99.37	100	0.996 8
Netatmo Welcome	99.87	98.87	0.993 6
PIX STAR Photo Frame	98.45	99.45	0.989 5
Samsung Smart Cam	98.98	99.80	0.993 9
Smart Things	98.85	99.97	0.994 0
TP Link Day Night Cloud Camera	99.36	98.87	0.991 1
TP Link Smart Plug	97.64	98.06	0.978 5
Triby Speaker	99.04	99.70	0.993 7
Withings Aura Smart Sleep Sensor	99.42	98.51	0.989 6
Withings Smart Baby Monitor	99.20	99.67	0.994 3
Withings Smart Scale	99.85	97.04	0.984 2

3.5　对比实验

对比实验的主要内容包括：原始大模型与微调模型的性能对比、原始与改进分词器的性能对比、不同载荷长度截取的对比。

3.5.1　原始大模型与微调模型的性能对比实验

为评估 LLM 在 IoT 设备识别中的领域适应能力，本文对比了原始模型与微调模型的性能。实验在 UNSW 测试集上进行，采用准确率、召回率和 F1 值作为评价指标。结果（如图 14 所示）表明，原始模型的准确率与召回率均为 16.93%，F1 值为 0.148 9，识别效果较差。经 IoT 流量数据微调后，模型性能显著提升，其准确率、召回率分别达到 99.42%、99.41%，F1 值为 0.994 1。微调有效增强了模型对领域特征的学习能力，大幅提升了分类精度。

3.5.2　原始与改进分词器性能对比实验

原始分词器在处理流量数据时，往往会对语义信息进行碎片化切分，导致关键信息被分割，不利于模型理解。而改进后的专用分词器能够更好地保留并完整切分出流量中的关键语义信息，有助于大模型更准确地理解和学习数据特征。

如图 15 所示，该图从平均令牌长度、模型训练时长以及小样本训练准确率 3 个方面对默认分词器与专用分词器进行了对比。可以看出，在平均令牌长度方面，专用分词器显著减少了分割后的 token 数量，由默认分词器的 763.2 个降低至 687.4 个，说明其在语义压缩和表达效率方面更具优势；在模型训练时长方面，使用专用分词器后训练时间由 31.1 h 缩短至 27.3 h，训练效率明显提升；而在小样本训练准确率上，专用分词器达到了 97.8%，相较默认分词器的 94.4% 提升了约 3.4%，展现出更强的小样本学习能力和泛化性能。

3.5.3　不同载荷长度截取的对比实验

本文分别在小样本环境与充足样本环境下（Aalto 数据集）评估了不同数据载荷长度对模型准确率的影响。实验结果显示，随着数据载荷长度的增加，模型的识别准确率呈现出显著上升趋势。尤其值得注意的是，在截取的数据载荷长度达到 1 024 字节时，模型达到了最优的准确度表现，如图 16 所示。

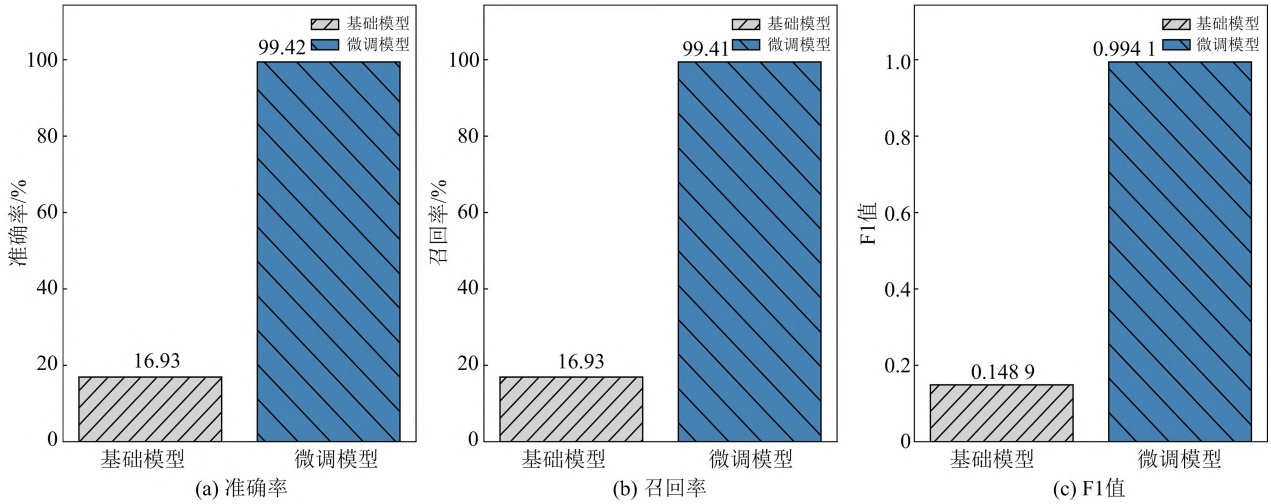


图 14 大模型领域适配前后的分类性能对比

Fig. 14 Comparison of classification performance before and after domain adaptation of large models

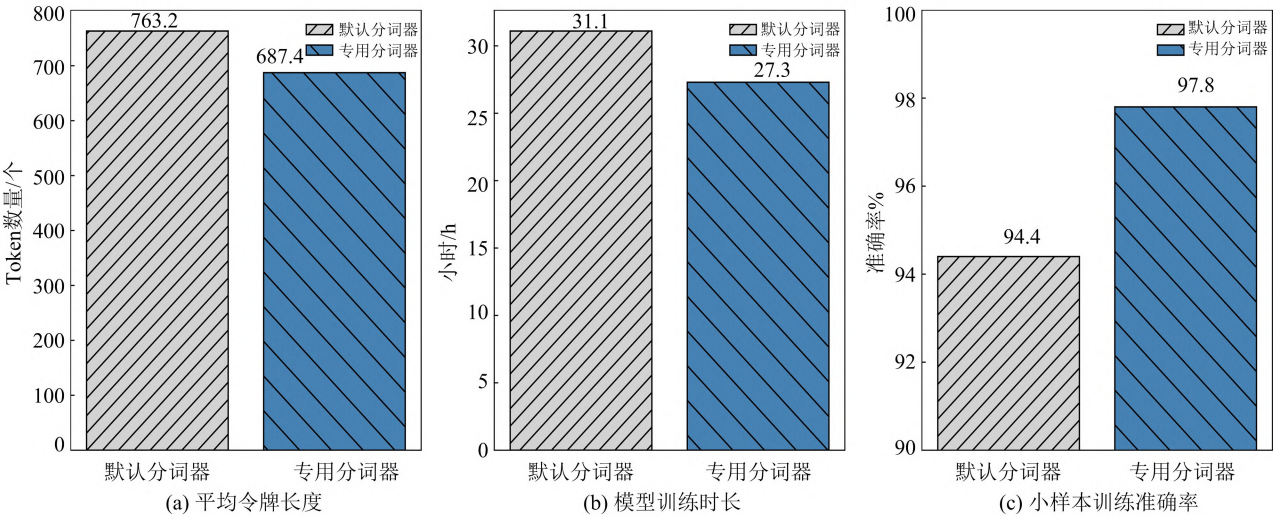


图 15 原始与改进分词器性能对比图

Fig. 15 A comparison chart of the performance of the original and improved tokenizers

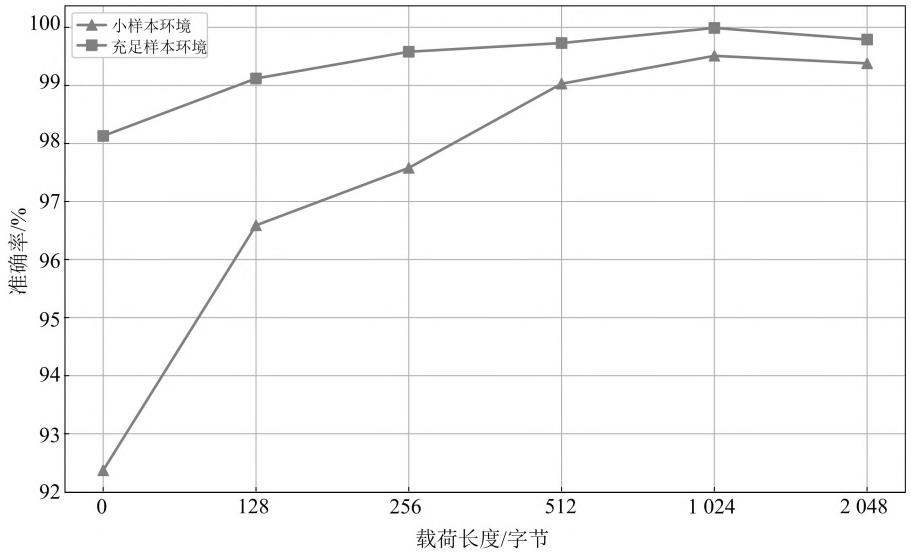


图 16 不同数据载荷长度下的截取效果对比图

Fig. 16 A schematic diagram comparing the interception effects under different data payload lengths

3.5.4　对比讨论

表 10 对比了现有 IoT 设备识别方法的关键技术指标，包括数据处理方式、特征提取策略和识别性能等。通过统一采用 Aalto、UNSW、CIC IoT 混合数据集进行评测，可以客观比较不同方法的分类效果。

表 10　IoT 设备识别技术对比

Tab. 10　Comparison of IoT device Identification technologies

识别方法	特征类型	数据集	评估指标	结果
IoTSentinel	数据分组头部字段	Aalto	准确率	81.50%
文献[24]	流统计特征	Aalto	F1 值	0.903 0
IoTDevID	数据分组头部字段和有效载荷	Aalto	准确率	83.30%
		UNSW	准确率	94.30%
文献[16]	数据分组链路层语义特征	UNSW	准确率	63.73%
			F1 值	0.645 9
文献[12]	流统计特征	CIC IoT 混合	准确率	86.30%
			F1 值	0.876 0
文献[20]	流统计特征和协议字段	UNSW	准确率	98.40%
IoTBERT	数据分组十六进制编码	Aalto	准确率	97.20%
		UNSW	F1 值	0.915 0
			准确率	92.10%
本文方法	分组级流量语义特征通用表示	Aalto	准确率	99.99%
			F1 值	0.999 9
		UNSW	准确率	99.42%
			F1 值	0.994 1
		CIC IoT 混合	准确率	98.83%
			F1 值	0.988 4

在 IoT 设备识别领域，多种方法已被提出并应用于实际场景中，然而这些方法在特征提取、算法选择以及识别性能上仍存在不同程度的局限性。IoTSentinel 模型[19]主要依赖于协议字段信息、IP 地址和数据包大小等数据包属性进行特征提取，其平均识别准确率约为 81.50%；文献[24]通过对数据分组的各个字段进行特征筛选，最终实现了约 0.903 0 的 F1 值。IoTDevID 方法[25]针对 Aalto 和 UNSW 数据集中的数据包内容行为特征进行建模，在这 2 个数据集上的识别准确率分别为 83.30% 和 94.30%。该方法依赖于繁琐的特征筛选过程，且所提取的特征通用性较差。基于 LLM 的设备分类方法利用链路层协议特征并通过微调实现设备识别[16]，最终在 UNSW 数据集上达到 63.73% 的准确率，但由于特征表达能力不足，整体性能仍有提升空间。文献[12]通过使用传统的机器学习方法，如决策树、随机森林、K 近邻和支持向量机，基于流统计特征来进行训练，识别准确率达到 86.30%，但其依赖长时间流量统计，导致识别延迟较高。文献[20]方法结合网络流量分析，提取协议字段的端口号、DNS 地址及密码信息，并与流统计特征相结合，最终识别准确率约为 98.40%。IoTBERT 方法[15]通过使用数据分组编码作为输入，在设备识别准确率上有所提升，在 Aalto 数据集中的准确率为 97.20%，在 UNSW 数据集中为 92.10%，但在不同环境下的稳定性仍显不足。

相比之下，本文提出的 IoT 设备识别方法在多个方面展现出显著优势。在识别精度方面，本方法在 Aalto 数据集上取得了 99.99% 的准确率，在 UNSW 数据集上达到 99.42%，在 CIC IoT 混合数据集上亦达到 98.83%，均优于现有主流方法，体现出卓越的分类性能。

4　结束语

本文提出了一种基于分组级流量语义特征增强的 LLM IoT 设备识别方法。该方法通过将复杂异构的 IoT 流量转化为标准化的分组级流量语义特征，利用 LLM 强大的特征学习能力，实现了端到端的高效设备识别。具体而言，本方法具有以下突出优势：一是通过分组级流量语义特征的通用表示，有效解决了传统方法中关键特征筛选复杂的问题；二是借助 LLM 的强大表征学习能力，可以自动捕获 IoT 设备流量的深层次特征；三是端到端的识别框架大大简化了识别流程，提高了识别效率。在公开数据集 Aalto、UNSW、CIC IoT 混合数据集上的实验结果表明，本文方法的平均识别准确率均显著优于其他的现有方法。

然而，在实际网络环境中仍存在一些需要进一步解决的问题：一是网络流量中可能存在复杂的异常数据包（攻击或恶意流量），这些异常数据可能会影响模型的初始特征学习；二是虽然通用分组级语义特征表示避免了复杂的关键特征筛选工程，但在面对大规模、高频率的冗余流量（如周期性心跳包、重复广播帧等）时，影响模型的推理效率与资源利用率。为此，建议在实际部署时增加数据预处理模块，并配备高性能计算设备以确保模型的最佳性能。

在未来的研究工作中，我们将持续扩充 IoT 设备流量语义特征的规模与多样性，以构建更具通用性和表征能力的设备流量特征表示。为进一步提升模型性能，下一步将持续扩展数据集覆盖范围，纳入更多基于不同通信协议和应用场景的 IoT 设备类型，同时增加非 IoT 设备样本，从而全面提升模型的分类准确率和跨场景泛化能力。特别地，未来将重点探索零样本与小样本学习技术在 IoT 设备识别领域的创新应用，开发自适应学习框架以应对新型设备快速接入带来的识别挑战，显著提升模型对未知设备的泛化性能。

参 考 文 献

[1] ROSEN M. Driving the digital agenda requires strategic architecture[EB/OL]. (2015-04-22)[2025-07-01]. https://idc-cema.com/dwn/SF_177701.

[2] FORTUNE BUSINESS INSIGHTS. IoT market size, growth: IoT industry report 2026[EB/OL]. (2019-01-01)[2025-07-01]. https://www.fortunebusinessinsights.com/industry-reports/internet-of-things-iot-market-100307.

[3] ALLIFANN M, ZUALKERNAN I A. Ranking security of IoT-based smart home consumer devices [J]. IEEE Access, 2022, 10: 18352-18369.

[4] ZHANG X L, UPTON O, BEEBE N L, et al. IoT botnet forensics: a comprehensive digital forensic case study on mirai botnet servers[J]. Forensic Science International: Digital Investigation, 2020, 32: 300926.

[5] SHENG C, ZHOU W, HAN Q-L, et al. Network traffic fingerprinting for IIoT device identification: a survey[J]. IEEE Transactions on Industrial Informatics, 2025, 21(5): 3541-3554.

[6] WU H, WU Q Y, GUANG C, et al. SFIM: identify user behavior based on stable features[J]. Peer-to-Peer Networking and Applications, 2021, 14(6): 3674-3687.

[7] HU E, SHEN Y L, WALLIS P, et al. LoRA: low-rank adaptation of large Language models[EB/OL]. [2025-07-01]. https://arxiv.org/pdf/2106.09685v1/1000.

[8] 樊琳娜，李城龙，吴毅超，等. 物联网设备识别及异常检测研究综述[J]. 软件学报，2024，35(1)：288-308.
FAN Linna, LI Chenglong, WU Yichao, et al. Survey on IoT device identification and anomaly detection[J]. Journal of Software, 2024, 35(1): 288-308. (in Chinese)

[9] STEINWART I, CHRISTMANN A. Support vector machines[M]. New York: Springer, 2008.

[10] CHEN T Q, GUESTRIN C. XGBoost: a scalable tree boosting system[C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco: ACM, 2016: 785-794.

[11] TAUNK K, DE S, VERMA S, et al. A brief review of nearest neighbor algorithm for learning and classification[C]//Proceedings of 2019 International Conference on Intelligent Computing and Control Systems. Madurai: IEEE, 2019: 1255-1259.

[12] RAHMAN M M, BOUHAFS F, HOSEINI S A, et al. UNSW HomeNet: a network traffic flow dataset for AI-based smart home device classification[J]. Computers & Industrial Engineering, 2025, 204: 111041.

[13] KUMAR R, SWARNKAR M, SINGAL G, et al. IoT network traffic classification using machine learning algorithms: an experimental analysis [J]. IEEE Internet of Things Journal, 2022, 9(2): 989-1008.

[14] TAKASAKI C, KORIKAWA T, HATTORI K, et al. Traffic behavior-based device type classification[C]// Proceedings of 2023 International Conference on Computing, Networking and Communications. Honolulu: IEEE, 2023: 353-357.

[15] 邢长友，王梓澎，张国敏，等. 基于预训练 Transformers 的物联网设备识别方法[J]. 信息网络安全，2024，24(8)：1277-1290.
XING Changyou, WANG Zipeng, ZHANG Guomin, et al. IoT device identification method based on pre-trained Transformers[J]. Netinfo Security, 2024, 24(8): 1277-1290. (in Chinese)

[16] MORALES G, ROMIT F T, BIENEK-PARRISH A, et al. IoT device classification using link-level features for traditional machine learning and large language models[C]//Proceedings of the 10th International Conference on Information Systems Security and Privacy. [S. l.: s. n.], 2024: 297-308.

[17] COMBS G. Wireshark and Tshark network protocol analyzer [EB/OL]. [2025-07-01]. https://www.wireshark.org.

[18] ZENG A H, XU B, WANG B W, et al. ChatGLM: a

family of large language models from GLM-130B to GLM-4 all tools [EB/OL]. [2025-07-01]. https://arxiv.org/abs/2406.12793.

[19] MIETTINEN M, MARCHAL S, HAFEEZ N, et al. IoT sentinel demo: automated device-type identification for security enforcement in IoT [C]//Proceedings of the 37th International Conference on Distributed Computing Systems. Atlanta: IEEE, 2017: 2177-2184.

[20] SIVANATHAN A, GHARAKHEILI H H, LOI F, et al. Classifying IoT devices in smart environments using network traffic characteristics [J]. IEEE Transactions on Mobile Computing, 2019, 18(8): 1745-1759.

[21] DADKHAH S, MAHDIKHANI H, DANSO P K, et al. Towards the development of a realistic multidimensional IoT profiling dataset [C]//Proceedings of the 19th Annual International Conference on Privacy, Security & Trust. Fredericton: IEEE, 2022: 1-11.

[22] 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.

[23] The Wireshark Project. Wireshark user's guide [EB/OL]. [2025-07-01]. https://www.wireshark.org/docs/.

[24] HAMAD S A, ZHANG W E, SHENG Q Z, et al. IoT device identification via network-flow based fingerprinting and learning [C]//Proceedings of the 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications. Rotorua: IEEE, 2019: 103-111.

[25] KOSTAS K, JUST M, LONES M A, et al. IoTDevID: a behavior-based device identification method for the IoT [J]. IEEE Internet of Things Journal, 2022, 9(23): 23741-23749.

作 者 简 介

尹鑫宇

男，2002 年生，硕士研究生，研究方向为网络空间测绘

E-mail：yinxinyu@nudt.edu.cn

施　凡

男，1983 年生，博士，教授，研究方向为网络空间测绘和网络空间测量

E-mail：shifan17@nudt.edu.cn

许成喜

男，1989 年生，博士，副教授，研究方向为互联网基础设施安全和网络空间测绘

E-mail：xuchengxi@nudt.edu.cn

章建成

男，2004 年生，硕士研究生，研究方向为网络空间测绘

E-mail：dawnlight@foxmail.com

葛明仪

女，2003 年生，硕士研究生，研究方向为网络空间测绘

E-mail：18980437826@163.com

责任编辑　殷文卓