基于突发特征词元自学习的未知加密恶意流量检测方法

沈 蒙[1],贾冀哲[1],赵卜凡[1],常力元[2],杨 明[3],任琛琛[1],宋 悦[2],祝烈煌[1*]
(1. 北京理工大学网络空间安全学院,北京 100081;2. 天翼安全科技有限公司基础能力部,北京 100020;
3. 国家超级计算济南中心,山东济南 250014)

摘 要: 当今,互联网流量已普遍加密,以保障其机密性与隐私性. 然而,攻击者常常滥用流量加密技术来隐藏其恶意网络行为. 由于加密恶意流量与加密良性流量具有相似特征,其能够轻易规避传统基于特征签名与深度包检测(Deep Packet Inspection,DPI)的检测方法. 现有加密恶意流量的检测研究主要集中于基于有监督学习的范式,尽管其在已知攻击类型上表现良好,但其有效性严重依赖于大量且持续更新的标记恶意流量样本. 面对恶意软件快速迭代、变种频繁以及加密隧道技术的广泛应用,有监督学习模型难以应对训练数据中未曾出现的未知攻击类型,存在显著的泛化能力不足问题. 此外,现有方法的特征表示多依赖于手工设计的统计特征,难以捕捉恶意行为在加密流量底层数据报文中的深层语义信息与复杂时序动态,导致特征区分度有限,无法有效适配新型攻击模式. 为此,本文提出了一种可靠的基于突发特征词元自学习的未知加密恶意流量检测方法 MalGuard. 通过分析网络传输底层机理及观察良性流量与恶意流量的关键特征差异性分布,创新地提出了一种基于流量突发特征的新型流量词元化表示方法,实现了对数据报文语义信息与时序动态的关联表征,为后续预训练模型提供了高信息密度的输入基础. 基于新型流量词元化表示方法,本文提出两项流量领域专用的自监督预训练任务——跨度掩码语言模型与跨度边界目标任务,通过掩码并重构流量数据报文的跨度内容,强化模型对跨度内数据报文上下文关联的整体感知,实现具备泛化能力的流量通用特征提取. 基于该特征,进一步构建适配流量特征分布的轻量级无监督学习算法,通过定位高维表征空间中的离群点,无需恶意标签数据即可实现对加密恶意流量的可靠检测. 为验证 MalGuard 的有效性,我们在三个公开数据集上进行了实验评估. 实验结果表明,MalGuard 在未知加密恶意流量上的检测表现超过了现有的最佳方法. 具体而言,将良性流量与恶意流量的样本数量定义为不平衡比例 β,在 β=4∶1 和 β=16∶1 时,MalGuard 的检测平均 F1 值分别为 91.76% 和 84.56%,相比现有最佳方法提高了 6.01 个百分点和 28.23 个百分点.

关键词: 网络安全;加密流量分析;恶意流量检测;恶意软件;自监督预训练;无监督学习
基金项目: 国家自然科学基金(No.62222201)
中图分类号: TP393.08 **文献标识码:** A **文章编号:** 0372-2112(2025)12-4231-19
电子学报 URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20250731

Unknown Encrypted Malicious Traffic Detection via Burst Feature Token Self-Learning

SHEN Meng[1], JIA Ji-zhe[1], ZHAO Bu-fan[1], CHANG Li-yuan[2], YANG Ming[3], REN Chen-chen[1], SONG Yue[2], ZHU Lie-huang[1*]
(1. *School of Cyberspace Science and Technology*, *Beijing Institute of Technology*, *Beijing* 100081, *China*;
2. *Department of Basic Ability*, *China Telecom Cybersecurity Technology Co., Ltd.*, *Beijing* 100020, *China*;
3. *National Supercomputing Center in Jinan*, *Jinan*, *Shandong* 250014, *China*)

Abstract: The widespread adoption of encrypted internet traffic ensures confidentiality and privacy, yet attackers increasingly leverage encryption techniques to conceal malicious network activities. As encrypted malicious traffic exhibits characteristics similar to benign encrypted traffic, it can easily evade traditional detection methods based on feature signatures and deep packet inspection (DPI). Current research on encrypted malicious traffic detection primarily focuses on super-

收稿日期:2025-08-21;录用日期:2025-12-17;责任编辑:覃怀银
*通讯作者:祝烈煌

vised learning paradigms. While effective against known attack types, their efficacy heavily relies on large, continuously updated labeled malicious traffic samples. Confronted with rapidly evolving malware variants and the widespread use of encryption tunneling techniques, supervised learning models struggle to generalize to unseen attack types, exhibiting significant limitations in adaptability. Furthermore, the feature representations in existing methods often depend on manually engineered statistical features, which fail to capture the deep semantic information and complex temporal dynamics of malicious behaviors within the underlying data packets of encrypted flows, resulting in limited feature discriminability and ineffectiveness against novel attack patterns. To address these challenges, we propose MalGuard, a reliable method for detecting unknown encrypted malicious traffic via self-supervised learning of burst-feature tokens. By analyzing the underlying mechanisms of network transmission and observing the key characteristic distributions between benign and malicious traffic, we innovatively propose a novel burst-aware traffic tokenization method, achieving a correlated representation of the semantic information and temporal dynamics of data packets and providing a high-information-density input foundation for subsequent model pre-training. Building on this token representation, we design two traffic-specific self-supervised pre-training tasks—Span-Masked Language Modeling and a Span Boundary Objective. These tasks mask and reconstruct spans of packet content to enhance the model's holistic perception of contextual dependencies within the data, enabling the extraction of generalized traffic features. Leveraging these features, we further construct a lightweight unsupervised learning algorithm adapted to the intrinsic distribution of traffic characteristics. By identifying outliers in the high-dimensional representation space, reliable detection of encrypted malicious traffic is achieved without requiring labeled malicious data. To validate the effectiveness of MalGuard, we conducted experimental evaluations on three public datasets. Experimental results demonstrate that MalGuard outperforms the SOTA methods in detecting unknown encrypted malicious traffic. Specifically, we define the imbalance ratio β as the ratio of benign to malicious samples, at β =4∶1 and β =16∶1, MalGuard achieves average F1 scores of 91.76% and 84.56%, surpassing the best existing baseline by 6.01 percentage points and 28.33 percentage points, respectively.

Key words: network security; encrypted traffic analysis; malicious traffic detection; malware; self-supervised pre-training; unsupervised learning

Foundation Item(s): National Natural Science Foundation of China (No.62222201)

1 引言

随着网络基础设施的高速发展与传输层安全性协议(Transport Layer Security,TLS)等加密通信协议的广泛部署,恶意软件的传输方式和通信行为也发生了显著变化.远程访问木马、勒索病毒等远程控制恶意软件利用加密技术进行命令控制与数据传输,隐藏其攻击载荷与通信模式.这使得传统基于深度包检测(Deep Packet Inspection,DPI)和手工规则的检测方法难以获取有效信息,检测恶意流量的能力大幅下降.截至2025年第一季度,71%的恶意软件已通过加密流量进行传播,该比例仍在逐年上升,对现有安全防御体系提出了巨大挑战,因此准确检测加密恶意流量迫在眉睫[1].

现有研究提出了多种加密恶意流量检测方法,主要分为基于有监督学习和基于无监督学习的检测方法.前者使用大规模标签数据训练检测模型,实现对特定类型加密恶意流量的检测.其中,一类方法基于数据包大小、时间间隔等统计特征构建行为模型[2~4];另一类则利用深度神经网络建模流量的序列特征[5~8],如卷积神经网络(Convolutional Neural Network,CNN)、Transformer等架构,但其对未知加密恶意流量的适应能力较弱.近年来,也有研究引入预训练-微调范式,在大规模无标签数据上学习流量通用表征,再用少量标签进行下游任务微调与分类,以缓解标签依赖问题[9~14].后者则通过异常检测、聚类分析等手段从良性流量中挖掘偏离行为,摆脱了对标签数据的依赖,在应对未知威胁和零日攻击方面展现出天然优势.其中,一类方法基于统计特征建模网络流量行为[15,16];另一类方法则关注序列结构建模,通过分析数据包序列中的时序模式检测异常流量[17~20].

尽管已有研究在加密恶意流量检测领域取得了显著进展,但现有检测方法仍面临两方面核心挑战:(1)恶意软件变种频繁、更新迅速,导致现有检测方法在特征表示上的表达能力受限,难以捕捉多样化的新型加密恶意流量;(2)恶意软件借助加密隧道传输通信内容[21],封装数据包会隐藏现有检测方法所依赖的协议字段与有效载荷,使得基于浅层特征提取和规则匹配的检测机制难以发挥作用.因此,如何在无恶意标签数据下,从加密流量中检测未知的加密恶意流量,成为当前网络安全领域亟待解决的重要问题.

为了解决上述挑战,本文提出了一种基于突发特征词元自学习的加密恶意流量检测方法MalGuard,能够在无恶意标签数据下实现加密恶意流量可靠检测.MalGuard的基本思想是将原始加密流量数据报文转换为预训练模型可理解的语义词元,使其能够提取针对恶意流量检测任务的通用特征.为此,我们对网络传输

的底层机理进行分析. 尽管加密技术隐藏了数据包的明文负载内容,但数据包的传输时序、方向和大小等元数据在传输过程中是无法加密且难以被完全掩盖的. 这些特征共同构成了流量的突发与底层网络活动的强关联性.

MalGuard包含3个关键模块. 首先,基于上述关键发现,提出了一种基于突发特征的流量词元化方法,将一条流划分成一系列突发(即连续同方向数据包序列),提取每个突发的数据报文和与前一突发时间间隔特征,形成突发特征对序列. 该序列有效涵盖了突发数据报文、时间间隔等具有区分度的元数据,实现对数据报文语义信息与时序动态的关联表征,提升预训练模型对流量词元的语义理解与上下文建模能力. 其次,为了从大规模无标签数据中深度学习通用的流量特征,提出了跨度掩码语言模型和跨度边界目标这2种新型的预训练任务,来学习流量领域特有的特征模式. 跨度掩码语言模型任务通过捕捉同一突发预设跨度内连续数据报文字节间的相关性,基于上下文对其进行重构表示,实现对数据报文词元的自学习;跨度边界目标任务通过利用被掩码片段左右边界的数据报文词元预测被掩码片段内容,强化对跨度内数据报文上下文关联的整体感知,从而实现流量通用特征提取. 最后,基于预训练模型提取的通用特征,采用无监督学习的隔离森林算法,构建良性流量特征空间,通过路径隔离定位和动态阈值判定,实现对未知加密恶意流量的可靠检测.

本文的主要贡献总结如下.

(1)提出了一种基于流量突发特征的新型流量词元化表示方法. 通过对网络传输底层机理的分析,并结合对良性流量与恶意流量在关键特征上的分布观察,发现了两者之间在突发数据报文长度、时间等特征上的显著统计差异性. 基于该差异性,设计了流量词元化表示方法,实现对数据报文语义信息与时序动态的关联表征,提升预训练模型对流量词元的语义理解与上下文建模能力.

(2)提出了2项流量领域专用的自监督预训练任务:跨度掩码语言模型与跨度边界目标任务. 通过跨度掩码并重构强化模型对跨度内数据报文上下文关联的整体感知,实现具有泛化性的流量通用特征提取. 基于提取的通用特征,构建适用于流量特征分布的轻量级无监督学习算法,对高维表征空间的离群点进行定位,使其能够在无恶意标签数据下实现加密恶意流量可靠检测.

(3)使用真实世界数据集对MalGuard进行评估. 实验结果证明,MalGuard相比现有方法在检测精度上取得了显著提升,在未知加密恶意流量上的检测表现超过了现有最佳方法. 具体而言,在CTU-Malware[22]数据集不平衡比例β=4∶1和β=16∶1时,MalGuard的检测平均F1值分别为91.76%和84.56%,比现有最佳检测方法提高了6.01%和28.33%.

2　相关工作

传统的加密恶意流量检测主要依赖于传统的基于签名的网络入侵检测系统(Network Intrusion Detection System,NIDS),这类系统通过提取加密恶意流量的特征模式并构建规则库或攻击签名[23],依赖手动编写的规则,利用模式匹配技术在网络流量中检测与已知攻击特征一致的行为,检测潜在的攻击行为[24~27]. 然而,随着网络服务和环境日益复杂,恶意软件快速变种、更新频繁,流量特征呈现高度多样化,传统方法依赖的浅层特征已经难以适应新型攻击场景[28]. 更重要的是,近年来加密隧道被广泛采用,攻击者将应用数据封装并加密传输,隐藏了协议字段和关键内容特征,使基于规则和深度包检测的方法难以获取有效信息,导致检测精度显著下降[29]. 本文对现有研究进行梳理,并将其分为基于有监督学习和基于无监督学习的检测方法,表1对这2类方法进行了总结.

表1　现有加密恶意流量检测方法总结

分类	方法	流量表示	特征提取器、分类器	能否应对新型恶意软件	能否应对加密隧道协议
基于有监督学习的检测方法	ST-Graph[30]	流量图	随机游走算法、随机森林算法	×	×
	CBSeq[31]	通道级别行为序列	Transformer	√	×
	YaTC[11]	流级字节矩阵	Transformer	×	√
	ET-BERT[10]	突发级词元	Transformer	×	√
	NetMamba[12]	固定步长数据包级词元	双向Mamba	×	√
	TrafficFormer[13]	数据包级词元	Transformer	×	√
基于无监督学习的检测方法	Kitsune[15]	统计特征	自编码器	√	×
	Anomal-E[32]	手工设计特征	图神经网络	√	×
	AMSL[33]	时间序列	卷积自编码器	√	×
	ContraMTD[34]	通道级统计特征	卷积神经网络、图注意力网络	√	√
	MalGuard	时序关联的突发级词元	Transformer、隔离森林算法	√	√

2.1 基于有监督学习的检测方法

大量研究使用有监督学习方法，通过在标签数据上提取统计特征或序列特征，训练分类模型检测加密恶意流量[35,36]，在已知恶意流量场景下取得了显著的检测性能. 传统方法主要提取统计特征，利用机器学习算法构建分类模型[37,38]. Holland 等人[2]提出自动化特征工程与分类模型训练框架，使用数据包的协议头部字段和生存时间、窗口大小等统计属性进行检测；Wang 等人[3]结合对比学习和联邦学习，利用协议类型、源字节数等统计特征实现更稳健的检测能力；Anderson 等人[4]提出基于上下文流的包间隔、包大小等特征检测加密恶意流量；Fu 等人[30]提出 ST-Graph，通过将多流构图并提取图特征与协议指纹特征来实现加密恶意流量检测. 然而静态的统计特征难以充分捕捉流量的动态变化，更多研究者开始关注流量的序列特征，利用深度学习方法进行加密恶意流量检测[39,40]. Lotfollahi 等人[41]提出使用端到端深度神经网络对原始加密流量的时间序列进行特征提取和检测；Cui 等人[31]提出基于多粒度顺序建模的深度序列分类模型，显式建模序列结构提升对未知加密恶意流量的检测能力.

近年来，预训练技术被广泛应用于有监督学习方法，先基于大规模无标签数据训练特征提取器，再利用少量标签数据微调. He 等人[9]提出的 PERT，将基于 Transformer 的双向编码模型[42]（Bidirectional Encoder Representations from Transformers，BERT）迁移到加密流量分类任务；Lin 等人[10]提出 ET-BERT，通过预训练-微调框架提升了流量分类精度；Zhao 等人[11]提出 YaTC，设计了多级流表示矩阵提取多层次流量信息，并结合 Transformer 通过大规模无标签数据预训练后微调实现高效分类；Wang 等人[12]提出的 NetMamba，使用基于结构化状态空间序列模型的 Mamba[43]结构，并采用掩码自编码器的预训练策略学习流量特征，在少量标签数据上微调实现流量分类；Zhou 等人[13]提出的 TrafficFormer，使用基于时空建模的图结构 Transformer，并在微调阶段应用数据增强策略；Qu 等人[14]提出的 TrafficGPT，使用自回归方式对无标签数据进行预训练，学习流量的长序列表示以实现加密流量分类；Shen 等人[44]提出的 SmartDetector，构建了能够捕获流量上下文信息的新型流量表示，并采用数据增强和对比学习提升检测模型的鲁棒性.

2.2 基于无监督学习的检测方法

针对未知加密恶意流量检测中标签数据匮乏、恶意流量模式多变的挑战，无监督学习方法因不依赖标签数据、能够挖掘偏离良性流量分布的异常模式，成为新的研究热点. Mirsky 等人[15]提出的 Kitsune，通过集成自编码器重构误差检测异常流量，从流量中提取隐式上下文特征，实现异常检测；Catillo 等人[16]提出的 CPS-GUARD，利用基于流量统计特征的深度自编码器，提升了加密恶意流量检测能力. 上述基于统计特征的方法往往难以捕获复杂多变的攻击行为，更多研究者开始对序列特征进行建模与分析. Zhang 等人[17]提出基于隔离森林的方法，通过固定长度的包序列矩阵建模，实现加密恶意流量检测；Wang 等人[18]提出基于端口负载字节分布的阈值模型识别异常流量，不依赖恶意标签，实现网络入侵检测；Fu 等人[19]提出的 Whisper，通过在频域上分析流量特征实现未知加密恶意流量检测；Fu 等人[20]提出的 HyperVision，通过识别异常流交互模式来检测未知加密恶意流量.

为了增强特征表达能力并克服手工设计特征的局限，研究者开始将预训练引入无监督检测领域. Caville 等人[32]提出的 Anomal-E，基于图神经网络进行自监督预训练，生成表征能力更强的流嵌入后用于聚类检测；Marino 等人[45]提出的 Net，通过自监督预训练 Transformer，不依赖分类标签，而是计算异常分数来检测加密恶意流量；Zhang 等人[33]提出的 AMSL，利用自监督模块和可变记忆融合模块，学习良性流量模式，并重构误差实现异常检测；Koukoulis 等人[46]在原始数据包序列上用对比学习进行预训练，在无监督场景中使用异常分数实现异常检测；Han 等人[34]提出的 ContraMTD，将对比学习应用于无监督的加密恶意流量检测，通过图双边注意力网络来捕获细粒度特征，实现恶意流量检测.

2.3 小结

尽管现有基于有监督学习与无监督学习的研究在加密恶意流量检测领域取得了显著进展，但仍存在两方面局限性. 首先，恶意软件家族更新迅速、变种频繁，通信行为呈现出高度的异质性，现有方法依赖的浅层统计特征或固定结构的建模方式难以全面表达复杂多变的流量特征，导致模型对未知恶意流量检测的泛化能力有限. 其次，随着加密隧道的广泛使用，通信数据在传输过程中被统一封装加密，隐藏了协议字段和载荷内容，使传统依赖规则匹配和浅层特征提取的检测方法难以获取有效信息，严重影响检测效果.

相比现有的流量词元化表示方法，本文流量词元化方法的创新主要体现在表征粒度和时序信息利用 2 个方面. NetMamba[12]和 TrafficFormer[13]以数据包为粒度对流量进行词元化；ET-BERT[10]虽然以突发为粒度进行词元化，但其没有考虑连续突发之间的时序特征，导致其词元化表征不全面的问题. 本文创新性地将突发数据报文及其时间间隔进行联合词元化，实现了对数据报文语义信息与时序动态的关联表征，从而克服现有方法因忽略时序特征而导致的词元化表征不全面问题.

3　问题定义和威胁模型

本节介绍了加密恶意流量检测的问题定义和威胁模型.

3.1　问题定义

本文的目标是设计一种端到端的加密恶意流量检测方法,核心目标是提升现有网络入侵检测系统对未知加密恶意流量的检测效果. 本文的方法设计遵循非拦截式旁路检测原则:当网络入侵检测系统部署于网关、路由器等边缘设备时,可通过流量镜像技术(如端口镜像)复制并转发双向网络流量至检测系统,确保其仅执行流量分析与检测功能,不干扰实际业务流量传输路径. 该特性使其能够无缝集成至现有网络防御体系,作为插件式增强模块运行. 本文的方法能够检测典型网络攻击类型,包括但不限于拒绝服务攻击、暴力破解攻击与端口扫描攻击,但无法检测不产生网络流量的攻击行为,如权限提升、本地代码注入等.

3.2　威胁模型

现有基于有监督学习的检测方法依赖标签数据,难以覆盖快速演化、频繁变种的恶意软件,面对"零日攻击"时性能显著下降[47,48]. 同时,大量恶意流量使用超文本传输安全协议(HyperText Transfer Protocol Secure,HTTPS)、TLS等加密协议封装传输,掩盖了协议字段和明文有效载荷,使传统检测方法失效. 因此,本文提出的检测方法能够在不依赖标签数据的前提下,有效检测潜藏在良性流量中的未知加密恶意流量. 威胁模型如图1所示.

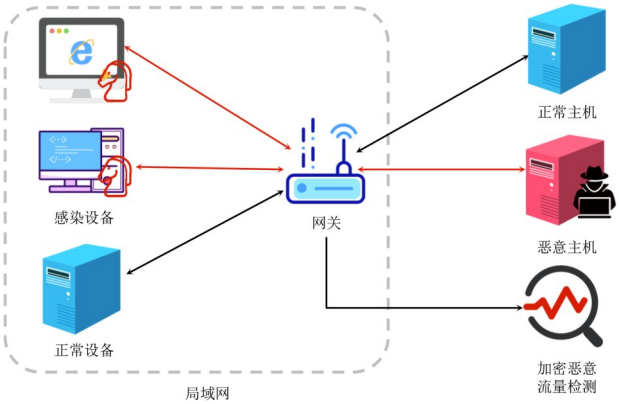


图1　威胁模型

假设攻击者可以不受约束地设计攻击流量模式,生成的加密恶意流量不在任何先前已知的恶意标签范围之内. 进一步假设攻击者缺乏操纵目标网络内端口的能力. 需要指出的是,像超文本传输协议(HyperText Transfer Protocol,HTTP)的80端口和HTTPS的443端口这类被广泛认可的标准端口,由于绑定它们需要系统级权限,因此攻击者难以直接将其开放或用于恶意目的. 假设检测系统具备通过网络网关监控加密流量的能力,但无法解密任何单个数据包. 检测系统可以持续收集流量数据以区分恶意流量和良性流量,但其无法预见攻击者可能发起的攻击类型或使用的恶意软件,也无法提前了解攻击者所使用的具体攻击方法.

4　流量表示方法

本节分析了网络传输的底层机理. 通过观察突发相关的关键特征分布,提出了一种新的基于突发特征的流量词元化表征方法,来捕捉这些高区分度特征.

4.1　网络传输底层机理分析与关键特征分布观察

检测加密恶意流量的核心在于寻找加密恶意流量与良性流量之间的显著差异性. 为此,本文深入分析网络传输的底层机制与负载特性,从加载元素的类型、规模及其时序信息等维度,探寻决定二者差异的内在因素. 在此基础上,将元素规模、加载时间、顺序及频率与产生的加密流量关联,提取能够有效表征加密流量的判别性特征.

在负载传输过程中,HTTP消息按时序传输,反映负载元素的请求与响应行为. 这些消息经由TLS协议加密保护并分片为TLS记录,作为传输控制协议(Transmission Control Protocol,TCP)段的有效载荷进行传输. 由于TCP是字节流协议,其在互联网协议(Internet Protocol,IP)数据包内传输记录,并对超过最大分段大小(Maximum Segment Size,MSS)的记录进行分段. 超过MSS的记录被分割至多个TCP有效载荷中,仅一个分段包含TLS头部,较小记录则常被TCP打包以填充至MSS上限. 这一封装过程在网络数据包与原始HTTP消息之间建立了固有关联.

基于负载传输过程,本文介绍2个核心概念的定义.(1)突发数据报文长度:连续同方向(服务器端到客户端或客户端到服务器端)的数据包序列包长度之和;(2)突发时间间隔:连续2个突发之间的时间间隔. 突发数据报文长度反映传输负载的规模特征,突发时间间隔则通过时序关系揭示负载传输的频率模式.

为深入理解加密恶意流量和良性流量的显著差异,利用CIC-IDS-2017[49]和CTU-Malware[22]这2个公开的恶意流量数据集(包含暴力破解攻击、拒绝服务攻击、Web应用攻击、恶意软件攻击等)进行分析. 通过可视化方法,对比分析二者在突发时间间隔、突发数据报文长度、突发持续时间上的分布特征. 具体来说,随机选取各类加密恶意流量样本各3 000个,并配置等量良性流量样本以平衡数据集. 采用包含相同五元组(源/目的IP地址、源/目的端口号、传输层协议)的流作为基本分析单元,并通过核密度估计(Kernel Density Estima-

tion,KDE)曲线对特征分析进行可视化,如图2所示.

突发时间间隔. 如图2所示,良性流量的突发时间间隔高度集中在接近0 ms的位置,这主要反映了底层网络协议栈的效率优化及实时交互需求,例如TCP连接的快速确认(Acknowledge,ACK)、域名系统(Domain Name System,DNS)短查询与HTTP响应,以及如同步(Synchronization,SYN)、中止(Finish,FIN)等单包控制信令等层面的即时性交互,这些高频的操作导致其突发间隔呈现极短且高度一致的特性. 相比之下,恶意流量的分布呈现多峰结构以及长尾特征,甚至延伸至300 ms,因多种攻击模式的规律性调度机制——包括命令与控制(Command and Control,C&C)通信的周期性心跳同步、分布式拒绝服务攻击(Distributed Denial of Service,DDoS)中僵尸网络的脉冲式发包策略[50]、端口扫描的探测包定时注入以及慢速攻击中刻意引入的人工延迟[51],此类攻击为规避检测或实现协同而采用的离散化、节律化发包策略,破坏了正常流量的连续性,从而在突发间隔分布上形成非正常的周期性特征.

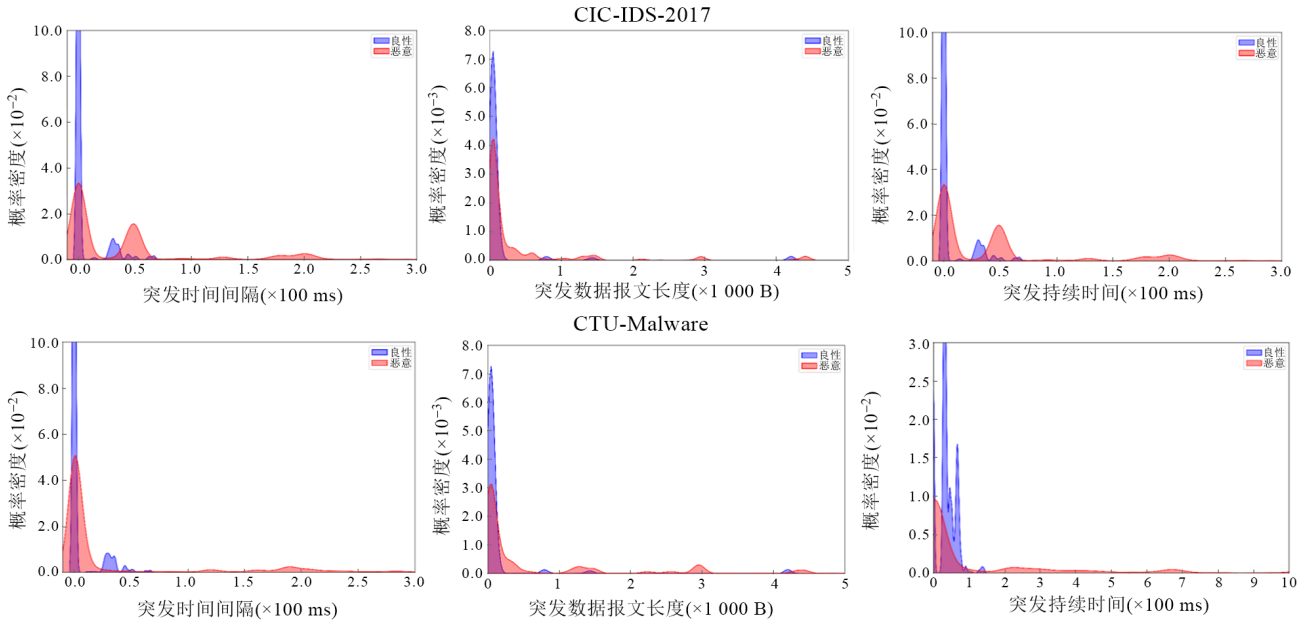


图2 CIC-IDS-2017[49]、CTU-Malware[22]中恶意流量与良性流量在3个特征上的核密度估计(KDE)

突发数据报文长度. 良性流量表现为在0字节处存在尖锐的主峰和若干次峰,这反映了其通信的多样性:既包含TCP ACK、控制信令等零负载操作,也包括HTTP分块传输、文件下载等具有可变负载的传输. 相比之下,恶意流量在0 B处呈现出高密度、快速衰减的单峰窄带分布,这体现出攻击行为的标准化特征. 例如,SYN泛洪攻击通常使用极小固定负载包[50],僵尸网络命令注入往往采用预制模板载荷[51],而端口扫描则发送统一的探针包. 这类预定义攻击操作使得突发数据报文长度被压缩在一个有限的取值区间内,本质上是因为网络攻击工具链的批量化生产模式消除了正常流量中负载的随机性.

突发持续时间. 良性流量的爆发性峰值近乎完全集中于0 ms附近,主要反映网络协议栈中瞬时单包操作的特征,包括TCP ACK确认、DNS单次查询响应以及HTTP短连接请求等交互,这些离散事件受底层操作系统和网卡驱动的优化几乎不产生可测量持续时间. 相比之下,恶意流量的平缓分布及其持续拖尾现象是因为攻击行为的连续性. 比如,端口扫描需连续发送探测包序列维持扫描状态,DDoS攻击为最大化攻击效能,会采用高密度发包策略延长突发时段[50],僵尸网络的命令控制通信需要多包指令传输以维持会话完整性,而暴力破解攻击则通过刻意延长突发传输间隔规避检测,这些操作迫使攻击行为无法以单包形式瞬时完成,从而形成具有统计显著性的突发持续时间分布特征.

尽管2个数据集的恶意流量种类、网络环境及采集时段均不相同,但良性流量与恶意流量在突发时间间隔、突发数据报文长度和突发持续时间等突发特征上始终呈现出共性分布差异,原因在于这些流量突发特征与底层网络活动具有强关联性. 这种跨攻击行为与跨场景的共性特征表明其具有普遍适用性,从而为加密恶意流量检测的特征选择提供了可靠依据.

4.2 基于突发特征的流量词元化

基于上述发现,我们提出了一种新型的基于突发特征的流量表示方法,将加密流量转化为能够适配自然语言处理范式的类词化词元,其处理流程主要分为突发特征提取和突发词元化. 主要符号说明如表2所示.

表2　主要符号说明表

符号	含义
$\boldsymbol{F}$	完整的会话流
$\boldsymbol{p}$	网络传输的最小单位,数据包
N	流中数据包$\boldsymbol{p}$的总数
$\boldsymbol{B}$	流的连续子序列,突发
K	流$\boldsymbol{F}$能够划分出的突发个数
Δ_k	相邻突发$\boldsymbol{B}_k$和$\boldsymbol{B}_{k-1}$之间的时间间隔
S	从流$\boldsymbol{F}$中提取的特征集
$\boldsymbol{p}_k$	第k个突发的特征对
$\boldsymbol{E}_{\text{token}}$	模型预训练阶段的输入嵌入类型,词元嵌入
$\boldsymbol{E}_{\text{pos}}$	模型预训练阶段的输入嵌入类型,位置嵌入
[CLS]	自然语言处理领域中预定义的特殊标记,分类符
[MASK]	自然语言处理领域中预定义的特殊标记,掩码符
[SEP]	自然语言处理领域中预定义的特殊标记,分隔符
[PAD]	自然语言处理领域中预定义的特殊标记,填充符
Burst	模型预训练阶段的突发词元输入序列
$\boldsymbol{x}_i$	突发词元输入序列的第i词元
L_{MLM}	跨度掩码语言模型的损失函数
L_{SBO}	跨度边界目标模型的损失函数
M	未知加密恶意流量检测模块每条会话流量截取的包数量
$\boldsymbol{\theta}$	基于双向编码器跨度表征的模型的可训练参数集合
N_estimators	基础估计器的数量,用于离群类别的估计与检测
max_samples	每棵树训练时的无放回抽样容量
Contamination_param	数据集中离群值的比例阈值

突发特征提取. 加密流量与自然语言处理及图像领域存在显著差异,因其本质是一串加密处理后的数据报文字节码,不具有人类可理解的内容及显性的语义单元. 在真实网络环境中,海量加密流量包含不同协议、应用、网站和服务等多种类别的异构流,导致特定流量的稳定判别性表征学习面临挑战. 基于网络传输底层机理分析与关键特征分布的观察,从流量中提取突发数据报文与突发时间间隔2个关键突发特征,形成突发特征对序列,用以捕捉正常流量与恶意流量之间的高区分性特征.

具体而言,首先依据流量五元组信息,即源/目的IP地址、源/目的端口号、传输层协议,对原始网络报文捕获文件进行分流,得到完整的会话流$\boldsymbol{F}=(\boldsymbol{p}_1,\boldsymbol{p}_2,\cdots,\boldsymbol{p}_n)$,其中$N$是流中数据包$\boldsymbol{p}$的总数,突发是流的连续子序列,其全部数据包具备方向一致性与时间相邻性,其定义如式(1)所示:

$$\boldsymbol{B}=(\boldsymbol{p}_i,\boldsymbol{p}_{i+1},\cdots,\boldsymbol{p}_j) \tag{1}$$

其中,$1\leqslant i\leqslant j\leqslant N$. 因此,整个流可以被划分为一系列交替的突发序列,如式(2)所示:

$$\boldsymbol{F}=[\boldsymbol{B}_1,\boldsymbol{B}_2,\cdots,\boldsymbol{B}_K] \tag{2}$$

其中,每个元素$\boldsymbol{B}$是一个突发,K是一条流$\boldsymbol{F}$能够划分出的突发个数,相邻突发方向不同. 例如,$\boldsymbol{B}_1$方向为客户端到服务器端,$\boldsymbol{B}_2$方向为服务器端到客户端. 相邻突发$\boldsymbol{B}_k$和$\boldsymbol{B}_{k-1}$之间的时间间隔Δ_k定义如式(3)所示:

$$\Delta_k=t_{\text{start}}(\boldsymbol{B}_k)-t_{\text{end}}(\boldsymbol{B}_{k-1}) \tag{3}$$

其中,$t_{\text{start}}(\boldsymbol{B})$和$t_{\text{end}}(\boldsymbol{B})$分别表示突发$\boldsymbol{B}$的第一个和最后一个数据包的时间戳.

最终,从流$\boldsymbol{F}$中提取出的特征集S由一系列特征对组成的序列. 每个特征对$\boldsymbol{p}_k$都是一个二元组,将一个突发时间间隔与其紧接其后出现的突发数据报文关联起来. 对于$k=1,2,\cdots,K$,我们能够获得特征对$\boldsymbol{p}_k=(\Delta_k,D_k)$. 其中,$D_k$是表示突发数据报文的字节序列. 因此,整个流$\boldsymbol{F}$通过突发特征提取得到这些特征对的序列,这个特征序列$S_{\boldsymbol{F}}$精确地刻画了流$\boldsymbol{F}$的完整行为,具体表示如式(4)所示:

$$S_{\boldsymbol{F}}=(\boldsymbol{p}_1,\boldsymbol{p}_2,\cdots,\boldsymbol{p}_k) \tag{4}$$

突发词元化. 将每个突发内的数据报文转化为词元序列,并在突发数据报文词元序列的前面补充突发时间间隔词元序列,形成突发词元序列,实现对数据报文语义信息与时序动态的关联表征. 该设计统一处理全部流量数据报文字段,能够同时支持非加密与加密流量表征,避免因协议异构性导致的特征偏差.

首先,通过双元语法模型将每个突发内的数据报文转化为词元. 每个突发数据报文被视为一个连续的十六进制字节序列. 双元语法模型首先将该序列切分为连续的、长度为2个字节的基本单元. 例如,序列“78e400”会被切分为“78”“8e”“e4”“40”“00”等双字节片段. 这种相邻字节配对的方式,能够有效捕捉加密流量数据报文中存在的局部相关性,即使在具有更高熵值的密文中,也能保留因加密实现、填充模式以及协议结构而产生的细微模式.

随后,这些双字节单元通过字节对编码(Byte-Pair Encoding,BPE)算法进行进一步融合与映射,来构建一个固定大小的词表(Vocabulary). 词表容量$|V|$设置为65 536,覆盖所有可能的双字节组合(0x0000至0xFFFF). BPE算法通过统计双字节单元在大量无标注流量数据中的共现频率,迭代合并高频出现的相邻单元,从而能够自动学习并生成更具代表性的复合词元. 这一步骤不仅压缩了序列长度,还使得模型能够捕获超出双字节的、更复杂的流量模式. 最终,每个单元都被分配一个唯一的整数ID,即转化为后续预训练模型能够处理的词元.

通过上述突发词元化过程,原本内容不可读的加密流量数据被转化为一种类似于句子的、由数字ID构成的词元序列. 这种词元化策略使得MalGuard能够直

接从字节层面学习加密流量的深层表征，无需依赖任何如IP、端口或证书之类的明文信息，因此在面对虚拟专用网络（Virtual Private Network，VPN）、洋葱路由器（The Onion Router，Tor）和新兴的TLS 1.3等强加密协议时，MalGuard依然展现出强大的特征提取和泛化能力，为后续的高精度恶意流量检测任务奠定了坚实基础.

在MalGuard的流量词元化框架中，突发时间间隔词元的设计旨在精确编码相邻突发之间的时间动态信息，从而增强模型对流量时序模式的捕获能力. 其核心实现基于时间戳的指数化编码，将连续的突发时间间隔转化为离散的词元序列，确保与模型的预训练机制兼容. 具体而言，对于每个突发，首先计算其与前一突发的时间戳差值Δt，若当前突发为流中的起始突发，则Δt设为0，以保持格式一致性. 随后，该时间间隔值被转换为指数形式，即表示为尾数（指浮点数的小数部分）和指数的组合，如式（5）所示：

$$\Delta t = m \times 10^{e} \tag{5}$$

其中，m为尾数，e为指数，这2个数值通过量化操作映射到整数域，以消除浮点数的不稳定性. 编码后，MalGuard将尾数和指数的二进制表示分解为8个字节，每个字节作为一个独立的词元，从而形成突发时间间隔词元的完整序列. 这种分字节处理不仅压缩了时间信息的数值范围，还保证了编码的可逆性.

这一编码策略与MalGuard的整体词元结构紧密集成，如图3所示，突发时间间隔词元位于每个完整突发词元序列的特定位置，在突发数据报文词元序列之前，共同构成基于突发特征的流表示. 这种设计确保了时序信息与数据报文同步处理，实现对数据报文语义信息与时序动态的关联表征. 构造词元的可视化展示如图4所示.

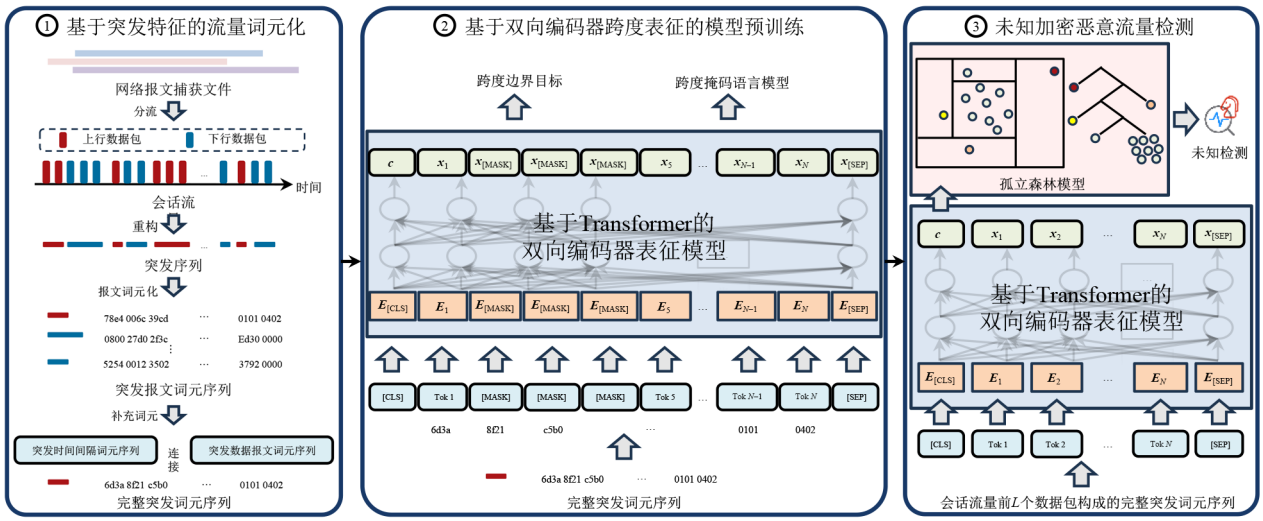


图3 MalGuard方法的系统概述

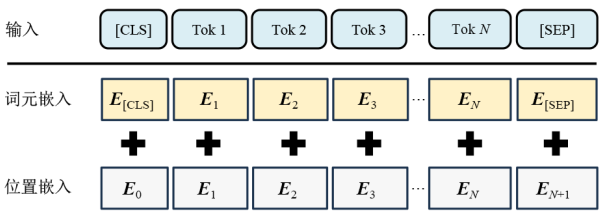


图4 模型输入的特征表示

为适配下一步模型预训练，我们增加了自然语言处理领域中分类符[CLS]、掩码符[MASK]、分隔符[SEP]和填充符[PAD]等预定义的特殊标记，用于规范模型输入格式、分割语义单元，并辅助模型训练. 其中，分类符[CLS]位于输入序列的开头，在后续未知加密恶意流量检测任务中作为整个序列的聚合表征. 掩码符[MASK]在预训练任务中随机替换输入突发词元中的部分字节，要求模型基于上下文预测被掩盖的原始字节，以学习字节间的上下文依赖关系. 分隔符[SEP]仅作为序列终止符，置于序列末尾. 填充符[PAD]将不满足最低长度要求的序列进行填充，以便模型批量处理，填充部分不参与实际计算，仅占位以保证张量形状一致. 对于一个给定的词元，它的输入形式由词元嵌入$\boldsymbol{E}_{\text{token}}$和位置嵌入$\boldsymbol{E}_{\text{pos}}$两部分构成.

5 本文方法

本节将介绍MalGuard方法以及基于双向编码器跨度表征的模型预训练模块和未知加密恶意流量检测模块的设计细节.

5.1 MalGuard方法概述

本文设计了一种适用于恶意流量检测领域的流量表征方法MalGuard，在训练阶段不使用任何恶意标签

数据的前提下实现未知加密恶意流量检测. MalGuard包含基于突发特征的流量词元化模块、基于双向编码器跨度表征的模型预训练模块及未知加密恶意流量检测模块3个主要模块,如图3所示.

基于突发特征的流量词元化模块. 为解决现有检测方法在恶意软件变种频繁、更新迅速下流量表征能力有限的问题,本文设计了一种基于流量突发特征的新型流量词元表示方法. 通过对网络传输底层机理的分析,并结合对良性流量与恶意流量在突发特征上的分布分析,发现两者之间在突发特征上的显著差异性. 基于显著差异,提出了基于流量突发特征的词元表示方法,使其能够实现对数据报文语义信息与时序动态的关联表征.

基于双向编码器跨度表征的模型预训练模块. 为解决加密隧道封装数据包隐藏协议字段使得基于浅层特征提取机制难以发挥作用的问题,本文设计了一种基于双向编码器跨度表征的自监督学习模块,通过对预设跨度内流量数据报文词元的连续掩码,设计专用于流量领域的跨度边界目标和跨度掩码语言模型两大预训练任务,实现对大规模无标签数据连续数据报文词元进行通用表征学习.

未知加密恶意流量检测模块. 本模块基于前面2个模块对流量的深度通用表征以及异常点在随机树结构中具有最短的平均路径长度的工作原理,构建适配于流量特征分布的轻量级无监督学习算法对离群点进行定位,使其能够在无恶意标签数据下实现对加密恶意流量的可靠检测.

5.2　基于双向编码器跨度表征的模型预训练

为了充分从无标签数据中学习流量通用表征,受BERT[42]和SpanBERT[52]启发,本文在2个关键维度突破其他流量分类领域的预训练模型框架. 首先,为了避免语义碎片化,采用基于几何分布的连续词元跨度掩码机制替代其他基于BERT的流量分类预训练模型原始的单点随机掩蔽,使其更适用于流量数据的结构化特征;其次,将原始的下一句预测模型(Next Sentence Prediction, NSP)替换成跨度边界目标(Span Boundary Objective, SBO)作为预训练子任务,强制模型仅通过跨度边界标记的表示重构被掩码跨度的完整语义.

在预训练阶段的设计中,突破了传统基于BERT的预训练模型采用随机位置的词元单点掩码策略的局限. 针对加密流量数据报文内容因加密而高度随机化、局部字节语义微弱的特性,创新性地引入了片段级别跨度掩码与跨度边界预测机制. 这种设计的核心在于将学习焦点从局部语义片段转向更为完整的语义片段. 片段级别跨度掩码和边界预测机制通过优化片段表示提升预训练模型对加密流量数据报文的理解和学习能力. 此外,对BERT下一句预测任务的移除避免了多报文拼接引入的噪声,并保留完整会话流量上下文,更贴合加密流量中行为关联分析的需求.

跨度掩码语言模型. 通过跨度掩码语言模型来捕获数据报文间的上下文依赖关系. 输入序列中随机选取一定跨度内连续的 k 个词元被替换为特殊标记[MASK],基于双向编码器跨度表征的模型通过深度双向上下文表征学习预测被掩码的原始词元.

具体而言,跨度通过几何分布采样动态生成长度可变的连续掩码片段,而非随机遮蔽离散词元. 这一机制模拟了加密流量中TLS证书链、HTTP头部字节块等协议字段的连续性特征,迫使模型学习报文内相邻字节的统计关联性,而非依赖自然语言中的显式语义.

对于给定的突发词元序列 $\mathbf{Burst}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n)$,采用迭代式文本跨度采样策略生成掩码子集 $Y\subseteq\mathbf{Burst}$ 直至消耗预设掩码比例,这里跟传统BERT模型保持一致,设置为15%. 每次迭代首先采样跨度长度(词元数)满足几何分布,随后均匀随机选择跨度起始点. 延续BERT的掩码规则,采取3种掩码策略:80%概率将进行掩码的词元替换为[MASK];10%概率将进行掩码的词元替换为随机词元;10%概率保持不变,但将独立词符掩码升级为跨度级联合掩码. 这一步骤我们最小化掩码位置的预测交叉熵损失,迫使模型从双向上下文中学习流量字节的规律. 跨度掩码语言模型的损失函数计算方式如式(6)所示:

$$L_{\mathrm{MLM}}=-\sum_{i\in Y}\log P(\boldsymbol{x}_i=\mathrm{token}_i|\boldsymbol{h}_i^{\mathrm{enc}};\boldsymbol{\theta}) \tag{6}$$

其中,Y代表掩码跨度集合,概率P是将被掩码的词元$\boldsymbol{x}_i$被预测为正确原始词元token_i的概率,$\boldsymbol{h}_i^{\mathrm{enc}}$代表位置$i$的编码器输出向量,$\boldsymbol{\theta}$代表基于双向编码器跨度表征的模型的可训练参数集合.

跨度边界目标模型. 在加密恶意流量检测领域,面对加密流量内容不可解析的挑战,跨度选择模型[53]通过重构数据报文内部的结构化语义表示,为解决加密恶意流量的隐蔽性提供了新的思路. 具体而言,该模型利用起始标记与终止标记定位TLS证书链、HTTP头部或DNS查询段等关键协议字段的边界,生成固定长度的跨度表示向量,从而将加密数据报文中的连续字节片段转化为可学习的语义单元.

为使预训练模型深度挖掘加密数据报文,本文引入跨度边界目标模型:在预训练阶段,随机掩码连续数据报文片段,模型仅依赖边界两侧的可观测词元预测被掩码片段的所有内部词元,并引入位置编码强化对片段相对位置的感知,从而迫使模型从加密扰动的噪声中学习结构化协议特征. 具体来说,将每个词元经过基于双向编码器跨度表征模型的输出表示为$\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n$. 给定掩码跨度内词元$(\boldsymbol{x}_s,\boldsymbol{x}_{s+1},\cdots,\boldsymbol{x}_e)\in Y$,

其中，s和e分别代表起止位置，跨度内任意词元$\boldsymbol{x}_i$的表征由外部边界词$\boldsymbol{x}_{s-1}$和$\boldsymbol{x}_{e+1}$的输出编码及目标词元相对位置嵌入$\boldsymbol{p}_{i-s+1}$共同运算所得，如式(7)所示：

$$\boldsymbol{y}_i=f(\boldsymbol{x}_{s-1},\boldsymbol{x}_{e+1},\boldsymbol{p}_{i-s+1}) \tag{7}$$

其中，位置嵌入标记了掩码词元相对于左边界词元$\boldsymbol{x}_{s-1}$的相对位置. 我们采用的表征函数$f(\cdot)$由2层基于高斯误差函数的激活函数(Gaussian Error Linear Units，GeLU)激活与归一化的前馈网络实现，如式(8)~(10)所示：

$$\boldsymbol{h}_0=\left[\boldsymbol{x}_{s-1};\boldsymbol{x}_{e+1};\boldsymbol{p}_{i-s+1}\right] \tag{8}$$

$$\boldsymbol{h}_1=\text{LayerNorm}\left(\text{GeLU}(\boldsymbol{W}_1\boldsymbol{h}_0)\right) \tag{9}$$

$$\boldsymbol{y}_i=\text{LayerNorm}\left(\text{GeLU}(\boldsymbol{W}_2\boldsymbol{h}_1)\right) \tag{10}$$

我们对掩码跨度$(\boldsymbol{x}_s,\boldsymbol{x}_{s+1},\cdots,\boldsymbol{x}_e)$内每个词元$\boldsymbol{x}_i$计算跨度边界损失，如式(11)所示：

$$L_{\text{SBO}}=-\sum_{(s,e)\in Y}\sum_{i=s}^{e}\log P(\text{token}_i=\boldsymbol{x}_i|\boldsymbol{y}_i;\boldsymbol{\theta}) \tag{11}$$

其中，s代表起始位置索引，e代表结束位置索引，Y代表所有掩码跨度集合，$\boldsymbol{y}_i$代表边界驱动表征，$\boldsymbol{\theta}$代表基于双向编码器跨度表征的模型的可训练参数集合.

总体而言，最终的预训练目标是上述2个损失函数的求和，其数学定义如式(12)所示：

$$L_{\text{total}}=L_{\text{MLM}}+L_{\text{SBO}} \tag{12}$$

5.3 未知加密恶意流量检测

本文提出了一种基于孤立森林(Isolation Forest)的新型未知恶意流量检测框架. 该框架充分利用前文预训练模型提取的通用流量特征，通过高维特征空间中的路径隔离机制实现离群点检测，从而有效应对未知加密恶意流量的检测挑战. 其核心机理在于：加密恶意流量在高维表征空间中具有统计稀疏性，导致其在随机生成的隔离树结构中，呈现出相较于良性流量显著更短的平均路径长度. 这一特性使孤立森林算法在加密恶意流量检测场景中展现出独特优势，尤其适用于高维特征空间的数据分布、高度不平衡的分类目标(真实世界中恶意流量占比远低于良性流量)以及离散化特征表示的学习任务. 具体而言，当模型将流量表征投影至由多层隔离树构建的度量空间时，恶意流量因偏离良性流量分布模式而被快速隔离至近根节点，而良性流量则需经历更深的路径遍历，从而形成可量化的分离边界.

图3展示了未知加密恶意流量检测的算法流程框架. 首先，基于突发特征的流量词元化模块对原始会话流量执行结构化预处理，提取每条会话中由前M个数据包构成的突发词元. 随后，基于双向编码器跨度表征的预训练模型对该词元序列进行深层语义编码，通过Transformer编码器堆叠结构处理输入序列，利用多头注意力机制捕捉报文间的长程依赖关系. 最终提取首位的[CLS]标记所对应的隐藏状态向量$\boldsymbol{c}$，该向量通过全局池化聚合整个流量会话的上下文语义信息，形成具有强判别性的通用表征向量. 最后，将向量$\boldsymbol{c}$作为特征输入孤立森林检测器进行模型训练与测试. 这种分层处理架构实现了从原始报文到高阶行为表征的端到端转化，为后续加密恶意流量检测提供信息稠密的输入空间.

该模型的构建与实例化主要涉及以下2个关键方面：其一，设定数据中的异常比例估计值，该参数虽不参与模型训练过程，但直接决定了判定离群值的决策边界；其二，模型采用集成学习策略，通过构建多棵孤立树综合判断，从而有效缓解单一决策树容易过拟合的问题，提升检测模型的鲁棒性和稳定性. 模型实例化参数具体包括：N_estimators决定基础估计器的数量，用于离群类别的估计与检测；max_samples设定每棵树训练时的无放回抽样容量；Contamination_param设定数据集中离群值的比例阈值. 由于离群点与良性数据存在可分离性差异，其在树结构中的深度通常显著小于良性样本. 孤立森林算法的核心为计算样本的异常评分，如式(13)所示：

$$\text{score}(\boldsymbol{x},n)=2^{-\frac{E(h(\boldsymbol{x}))}{c(n)}} \tag{13}$$

其中，$c(n)=2H(n-1)-\dfrac{2(n-1)}{n}$为归一化因子，用于平衡样本数量对路径长度的影响；$H(i)$为调和数，通常估算为$ln(i)+\gamma$，其中$\gamma$为欧拉常数. 通过参数Contamination_param设定异常比例阈值α，并依据分位数$s_{\text{thres}}=\text{Quantile}(s,1-\alpha)$判定异常. 其算法时间复杂度$O(n\cdot t\cdot\log\psi)$满足实时性需求($t$：N_estimators，$\psi$：max_samples)，远低于基于距离或密度的方法，这种效率使其能够支持大规模流量数据处理. 此外，隔离森林算法的子采样机制使其无需遍历全量数据. 这不仅加速训练，还降低内存占用，而基于密度的聚类算法[54](Density-Based Spatial Clustering of Applications with Noise，DBSCAN)需全局计算距离矩阵，资源消耗更高.

6 实验结果与分析

本节基于3个真实世界数据集对MalGuard进行评估. 首先介绍基线通过方法、测试数据集、评价指标、验证过程说明和实现细节. 通过将MalGuard的检测准确性与基线方法进行对比，旨在验证MalGuard能够在良性流量与恶意流量数据不平衡比例下，实现可靠的未知加密恶意流量检测. 此外，我们还评估了MalGuard在不同传输层协议场景下的泛化能力和时间开销，并对MalGuard开展消融实验.

6.1 实验设置

基线方法. 为了衡量本方法的先进性,本文选择了4个基线方法进行对比.

(1)CBSeq[31]是一种基于有监督学习的未知加密恶意流量检测方法,通过多粒度的顺序建模构建深度序列分类模型,显式捕捉序列结构信息.

(2)Kitsune[15]是一种基于无监督学习的未知加密恶意流量检测方法,提取每个数据包的特征,使用自编码器重构误差检测异常流量.

(3)ContraMTD[34]是一种基于无监督学习的未知恶意软件检测方法,提出了基于图双边注意力网络并应用对比学习,捕获细粒度特征.

(4)YaTC[11]是一种结合预训练模型的有监督学习检测方法,使用多级流表示矩阵表示多层次流量信息,并使用Transformer通过大规模无标签数据预训练、下游任务微调,实现加密恶意流量检测.

其中,CBSeq[31]和ContraMTD[34]未对其源代码进行开源,我们通过联系原文作者获取了其源代码进行实现. 除了上述2种方法外,Kitsune[15]和YaTC[11]均使用作者公开的源代码构建. 为确保对比的公平性,我们对上述基线方法进行了参数调优,使其检测性能达到或超越其原文中报告的水平.

测试数据集. 本文使用以下3个开源的真实世界数据集进行实验评估,数据集包含的恶意流量类型与规模介绍如表3所示.

表3　包含恶意流量类型与规模的数据集介绍

数据集	数据集ID	流量类型
CTU-Malware[22]	D1	Dridex (26046)、Emotet (21580)、Geodo (12252)、Miuref (16008)、Zeus (16898)、TrickBot (24024)
CIC-IDS-2017[49]	D2	DoS (22919) 、FTP-Patator (7894)、Portscan (15773)、SSH-Patator (5861)、DDoS-LOIT (12708)
USTC-TFC 2016[55]	D3	Cridex (16385)、Geodo (40947)、HTBot (6341)、Miuref (13481)、 Neris (33890)、Nsis-ay (6165)、Shifu (9634)、Tinba (8505)、Virut (33147)、Zeus (10972)
	D4	良性流量 (309887)

(1)CTU-Malware[22]是一个常用于恶意软件流量分析的真实网络数据集,收录了多个恶意软件生成的通信行为,广泛应用于恶意流量检测与分析研究中. 在本研究中,我们选择了数据规模最大的6种恶意软件类型:Dridex、Emotet、Geodo、Miuref、Zeus和TrickBot,用于构建数据集D1.

(2)CIC-IDS-2017[49]是一个本领域广泛应用的真实世界网络入侵检测数据集,包含了从上百个主机采集的通信流量. 在本研究中,我们选择了数据规模最大的5种典型网络攻击类型:拒绝服务攻击(Denial of Service,DoS)、文件传输协议(File Transfer Protocol,FTP)暴力破解攻击(FTP-Patator)、端口扫描攻击(Portscan)、安全外壳协议(Secure Shell,SSH)暴力破解攻击(SSH-Patator)和LOIT分布式拒绝服务攻击(DDoS-LOIT),用于构建数据集D2.

(3)USTC-TFC 2016[55]是一个基于真实网络环境采集的流量数据集,涵盖了10种良性应用程序和10种不同恶意软件产生的通信流量数据,恶意软件类型包含Cridex、Geodo、HTBot、Miuref、Neris、Nsis-ay、Shifu、Tinba、Virut和Zeus. 本研究中,我们将其中的恶意流量用于构建数据集D3,良性流量则用于构建数据集D4.

我们以流(Flow)为统计粒度,使用Tshark工具对上述3个开源数据集中加密流量与非加密流量所占比例进行分析. 对3个数据集中的恶意流量数据包捕获(Packet Capture,PCAP)文件(即D1、D2、D3)分别进行分流操作,分流后的单流包含TCP流与用户数据报协议(User Datagram Protocol,UDP)流. 而后,分别遍历D1、D2、D3的每条流,来判断这条流是否被加密. 具体而言,通过检查每条TCP流是否包含TLS加密协议数据包,每条UDP流是否包含数据包传输层安全性协议(Datagram Transport Layer Security,DTLS)、快速UDP网络连接(Quick UDP Internet Connections,QUIC)等加密协议,来判断这条流是否被加密. 遍历全部流后,计算D1、D2和D3数据集中加密流量的占比,结果分别为42.1%、23.2%和20.9%.

在分析其流量特点时发现,在CTU-Malware数据集中,其恶意流量主要源自恶意软件与命令控制(C&C)服务器的通信活动. 鉴于C&C通信对隐蔽性的内在需求,采用TLS/HTTPS等加密协议已成为一种普遍做法,以规避检测,因此该数据集呈现出相对较高的加密流量比例. 相比之下,CIC-IDS-2017与USTC-TFC 2016的加密流量比例相对较低,这主要与其收录恶意行为的固有特性密切相关. USTC-TFC 2016数据集包含的恶意流量,多源于相对传统的恶意软件家族,这些恶意软件在进行C&C通信或数据回传时,较多地采用未加密的明文协议(如HTTP). CIC-IDS-2017数据集则模拟了包括暴力破解、拒绝服务、扫描攻击等多种攻击场景. 此类攻击的机制本身往往不依赖于加密的通道,例如,Web攻击能直接利用应用层明文协议(如HTTP)的漏洞发起,而部分网络扫描、暴力破解和拒绝服务攻击也一般通过非加密通道进行. 此外,这些数据集采集于2016—2017年,在一定程度上反映了当时恶意软件尚

未普遍采用强加密手段的背景，而现代恶意软件使用TLS等强加密的比例已显著上升[1].

需要特别指出的是，MalGuard在特征提取过程中并未对加密与非加密两类流量进行区分处理，而是采用统一的特征提取策略. 其本质是基于流量的元数据特征，不依赖任何明文信息. 具体来说，提取的是一条流中每个突发的数据报文和时间间隔特征构成的特征对序列. 因此，无论流量是否加密，MalGuard均以相同的方式处理，其检测效果依赖于流量行为模式的差异，而非负载内容是否可见.

评价指标. 加密恶意流量检测的核心目标在于提升检测的准确性. 为此，我们采用在学术界应用最为广泛[15,20,31]的F1值和AUC(Area Under the ROC Curve)值，即接受者操作特性曲线(Receiver Operating Characteristic curve，ROC)下面积作为评估指标. 此外，还引入了精确率(Precision)来作为辅助评估标准. 上述评价指标具体计算方式如式(14)所示：

$$\text{Precision}=\frac{\text{TP}}{\text{TP}+\text{FP}},\ \text{Recall}=\frac{\text{TP}}{\text{TP}+\text{FN}},$$
$$\text{F1}=\frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}},\ \text{FPR}=\frac{\text{FP}}{\text{FP}+\text{TN}} \tag{14}$$

其中，TP、TN、FP和FN分别代表真阳性、真阴性、假阳性和假阴性，FPR为假阳性率(False Positive Rate，FPR). AUC值越接近1，表明模型的整体检测性能越好.

验证过程说明. 需特别指出的是，本实验所用训练集与测试集完全独立. 为降低样本选择中随机性的影响，所有实验均重复5次，以确保结果更具可信度.

实现细节. 为避免数据包头(尤其是IP地址、端口号等有限集合中的强标识信息)引入的偏差干扰[41]，我们对以太网帧头、IP报头以及TCP报头中的协议端口字段采用一种流量匿名化的随机化策略[56]. 在基于双向编码器跨度表征的模型预训练阶段，将几何分布中的参数p设置为0.2，批大小设置为16，总训练部署设置为100 000，学习率设置为0.000 05，预热比例设置为0.1. 使用开源网络数据集[22,49,55]的约12 GB无标签数据(3个数据集分别包含VPN流量、恶意软件流量与网络攻击流量)用于模型预训练. 在未知加密恶意流量检测阶段，基于预训练模型在流量分析领域的研究实践[10~14]，截取每条会话流量的前5个数据包作为输入，将森林中树的数量设置为200，每棵树的训练样本量设置为256，预估数据集中异常点比例设置为0.05，随机种子设置为42.

对于无监督学习方法MalGuard、Kitsune[15]和ContraMTD[34]，本文仅使用D4中的部分良性流量数据进行模型训练，使用D1、D2、D3分别和D4中的非训练数据混合用于模型测试. 对于有监督学习方法CBSeq[31]和YaTC[11]，为了保证公平性，我们确保训练集和测试集中的恶意流量样本完全独立. 具体而言，良性数据集用法与无监督学习方法保持一致. 然而，对于恶意流量，分别用D1作训练数据训练模型，使用D2和D3数据作测试；用D3作训练数据训练模型，使用D1数据作测试.

6.2 不平衡比例数据下的未知加密恶意流量检测

本节评估不同方法在未知加密恶意流量检测中处理不平衡比例数据的能力. 具体来说，为D1、D2、D3中的恶意流量与D4中的良性流量构建了一个二元分类器. 为了观察这些方法在良性数据和恶意数据不平衡比例变化时的性能，我们定义良性流量与恶意流量的样本数量的比例为β，并设置比例β=4∶1、β=16∶1和β=48∶1. 为了评估各检测方法的可靠性，将检测D1中6类恶意流量的精确率最低值加粗并标记了下划线. 评估结果如表4、表5和图5所示.

如表4所示，MalGuard在每个不平衡比例设置下均实现了最佳检测效果. 即使在D1数据集下不平衡比例β=16∶1时，MalGuard的平均F1值和平均AUC值分别为84.56%和97.64%，相比该不平衡比例下最佳基线方法YaTC分别提高了28.33个百分点和3.96个百分点，这表明MalGuard的基于突发特征的流量词元化和基于双向编码器跨度表征的模型预训练能够有效提取加密恶意流量与良性流量之间的显著差异性，并在高不平衡比例下保持可靠检测. 值得注意的是，当β=4∶1时，CBSeq的平均F1值仅为48.35%，这主要源于训练集与测试集恶意软件类型差异导致的流量特征分布偏移. 有监督学习模型需依赖全面且多样化的恶意流量训练集充分学习特征，才能保障在不同恶意流量数据集中的检测可靠性. 此外，尽管Kitsune和ContraMTD基于无监督学习框架，在未知加密恶意流量检测场景中被赋予理论预期优势，但其在不平衡比例下的检测性能却低于有监督方法YaTC. 其根本原因在于，YaTC通过强表征能力的流量表示方法与预训练模型，实现了对流量通用特征的深度提取，显著优于Kitsune和ContraMTD的泛化能力. 表5展示了不同方法在数据集D2、D3上的评估结果，进一步验证了MalGuard在多类型加密恶意流量检测中兼具泛化性.

如表4所示，在不同不平衡比例下，MalGuard对多种加密恶意流量的检测均表现出更稳定的精确率. 其平均精确率波动范围显著收窄，且最低精确率相比基线方法明显提升. 从β=4∶1增大到β=16∶1，最低精确率保持在73%以上，并相比其他方法，下降幅度最少. 在β=48∶1时，MalGuard的平均和最小精确率分别为67.00%和65.26%，相比该不平衡比例下最佳基线方法ContraMTD提高23.86个百分点和33.93个百分点. 这证明了即使没有恶意标签数据，MalGuard依然能够通过基于突发特征的流量词元化模块和基于双向编码器跨度

表4　在数据集D1中不同良性与恶意流量不平衡比例下不同检测方法的评估结果　　单位:%

β	恶意软件	Kitsune[15]			ContraMTD[34]			CBSeq[31]			YaTC[11]			MalGuard		
		Pre	F1	AUC	Pre	F1	AUC	Pre	F1	AUC	Pre	F1	AUC	Pre	F1	AUC
4:1	Dridex	**19.93**	33.06	42.86	68.16	69.68	72.57	47.70	48.86	67.75	70.85	82.88	90.90	87.33	93.19	98.41
	Emotet	21.44	35.12	53.05	**57.89**	48.65	71.08	47.66	49.60	68.10	89.89	89.92	94.03	**86.10**	87.91	93.74
	Geodo	20.09	33.40	49.10	68.28	69.82	71.86	47.16	47.68	67.80	82.29	90.28	96.20	87.33	93.19	98.43
	Miuref	20.97	34.57	52.71	59.53	52.21	66.25	47.15	47.75	67.75	70.75	82.87	93.50	87.13	92.34	98.13
	Zeus	20.06	33.35	48.37	62.03	60.65	72.84	**47.08**	47.82	67.49	**62.61**	77.01	92.13	87.09	92.13	98.05
	TrickBot	20.18	33.55	50.60	66.65	60.09	76.18	47.42	48.43	68.02	85.00	91.59	95.77	87.01	91.78	97.53
	平均值	20.44	33.84	49.44	63.76	60.18	71.79	47.31	48.35	67.81	76.89	85.75	93.76	**87.00**	**91.76**	**97.38**
16:1	Dridex	6.09	11.21	49.58	50.05	56.53	69.58	45.66	36.36	62.69	63.01	73.88	94.10	75.45	85.97	98.67
	Emotet	5.73	10.52	47.30	**46.89**	48.40	71.53	30.19	35.91	68.08	34.30	38.60	93.25	**73.43**	80.79	93.97
	Geodo	6.09	11.21	49.58	54.24	59.22	69.10	36.93	36.33	62.47	39.45	55.68	92.87	75.45	85.97	98.70
	Miuref	5.77	10.60	47.59	57.94	50.48	77.37	28.01	31.88	67.46	50.43	60.60	95.88	75.13	85.13	98.39
	Zeus	5.72	10.49	47.93	55.73	54.59	74.04	**21.21**	28.54	69.36	45.31	60.47	94.56	75.06	84.93	98.29
	TrickBot	**5.65**	10.40	46.75	59.46	55.16	66.80	33.62	35.43	70.75	**32.54**	48.16	91.44	74.92	84.58	97.79
	平均值	5.84	10.74	48.12	54.05	54.06	71.40	32.60	34.08	66.80	44.17	56.23	93.68	**74.91**	**84.56**	**97.64**
48:1	Dridex	1.05	1.07	23.53	**25.11**	33.43	59.09	8.69	15.57	78.55	17.62	29.75	94.12	67.64	80.66	98.60
	Emotet	**1.04**	1.07	23.34	40.12	44.52	51.91	5.30	9.02	58.26	**11.45**	20.53	90.87	**65.26**	75.59	93.91
	Geodo	1.10	1.19	20.82	48.93	49.46	52.55	5.40	9.19	59.69	18.16	30.51	94.30	67.64	80.66	98.62
	Miuref	1.09	1.17	20.34	46.89	48.48	69.21	5.33	9.23	59.34	17.88	28.92	86.58	67.26	79.84	98.33
	Zeus	1.06	1.10	22.59	48.77	46.74	56.48	6.31	11.06	62.81	19.06	30.80	87.98	67.17	79.64	98.25
	TrickBot	1.09	1.16	21.20	49.00	49.49	55.50	**3.92**	6.68	51.52	19.03	30.73	87.77	67.01	79.30	97.71
	平均值	1.07	1.13	21.97	43.14	45.35	57.46	5.83	10.12	61.69	17.20	28.54	90.27	**67.00**	**79.28**	**97.57**

注:加粗并带有下划线的数值表示各方法在检测6种不同恶意软件类别时的最低精确率,仅加粗数值表示MalGuard方法在对应数据集及不平衡比例下的平均Pre、F1值和AUC值.

表5　在数据集D2、D3中不同良性与恶意流量不平衡比例下不同检测方法的评估结果　　单位:%

β	数据集	Kitsune[15]			ContraMTD[34]			CBSeq[31]			YaTC[11]			MalGuard		
		Pre	F1	AUC	Pre	F1	AUC	Pre	F1	AUC	Pre	F1	AUC	Pre	F1	AUC
4:1	D2	47.38	46.44	66.36	87.91	73.65	70.78	64.38	71.07	77.19	88.79	84.28	99.09	**82.44**	**90.38**	**99.95**
	D3	64.09	65.15	76.91	81.33	57.55	83.20	84.83	78.52	83.99	92.30	90.82	98.26	**88.68**	**92.70**	**98.52**
16:1	D2	10.49	17.25	61.08	74.78	65.15	62.69	62.89	70.39	76.95	64.13	71.22	89.31	**72.73**	**82.27**	**98.89**
	D3	22.35	30.44	62.38	66.13	55.62	78.76	50.92	50.13	74.21	60.22	70.72	91.25	**76.04**	**85.29**	**97.78**
48:1	D2	2.18	4.26	52.58	63.73	60.75	60.37	55.02	60.15	62.46	62.16	60.45	73.21	**55.56**	**71.43**	**99.90**
	D3	1.22	2.39	55.02	50.59	48.88	76.71	25.03	33.36	69.07	54.75	51.52	88.62	**68.53**	**80.35**	**97.72**

注:加粗数值表示MalGuard方法在对应数据集及不平衡比例下的平均Pre、F1值和AUC值.

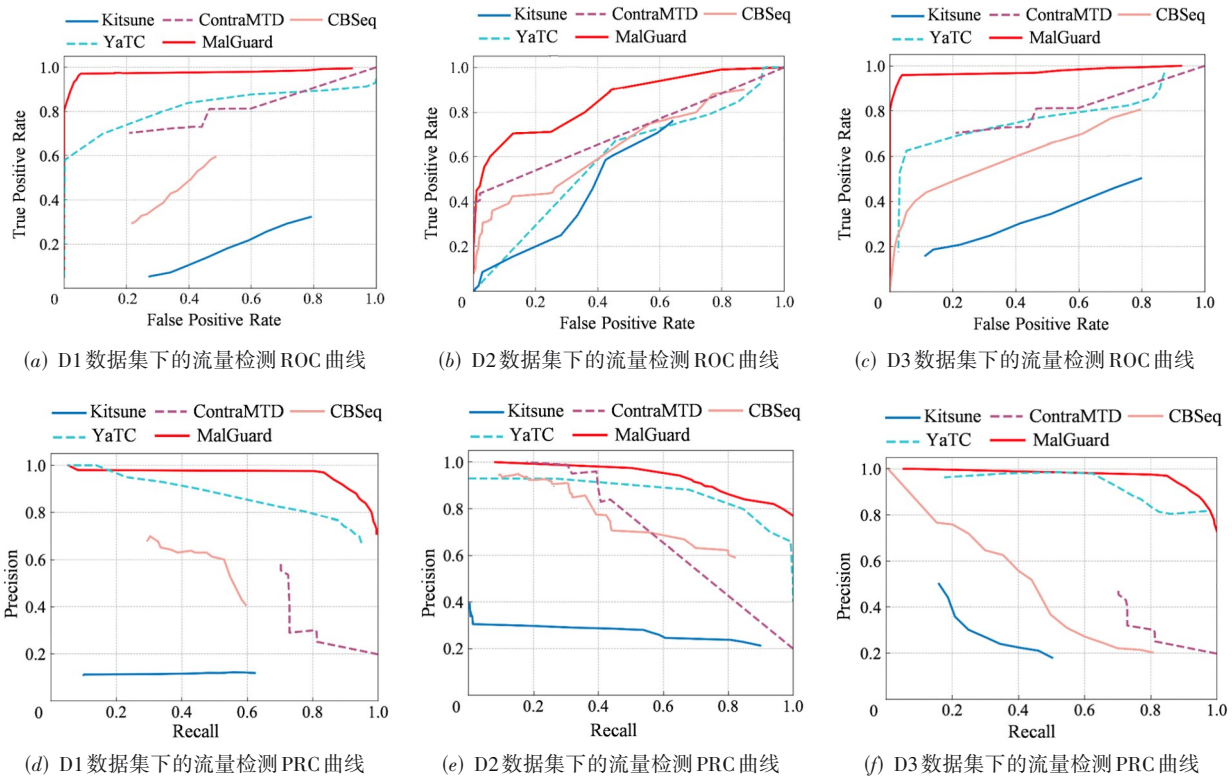
表征的模型预训练模块对加密流量的通用特征进行自学习,从而提升未知加密恶意流量检测效果.值得注意的是,在β=4:1时,即使该不平衡比例下最佳基线方法YaTC的平均精确率和平均F1值仅比MalGuard低10.11个百分点和6.01个百分点,但其最低精确率相比MalGuard低23.59个百分点,这表明YaTC无法对不同恶意流量类型实现整体的高精度可靠检测.

根据图5(a)、图5(b)、图5(c)的实验结果,MalGuard在与本场景最佳基线方法YaTC相比提升准确度的同时,显著降低了误报率.此外,图5(d)、图5(e)、图5(f)的精确率-召回率曲线进一步表明,MalGuard在相同召回率水平下的精确率显著优于所有基线方法,凸显其在高精度分类任务中的显著优势.

6.3　不同传输层协议场景下的泛化能力

本节评估MalGuard在未知加密恶意流量检测中对不同传输层协议的泛化能力,选择传输层最核心和使用最广泛的2种协议,即TCP协议和UDP协议.

在D1、D2、D3三个数据集上开展实验,并设置了良性流量与恶意流量不平衡比例β分别为4:1、16:1与48:1,以模拟不平衡比例数据场景,评估结果如表6所示.实验结果表明,MalGuard在TCP与UDP协议下均能保持有效的检测性能,说明其具备一定的跨协议泛化能力.



(*a*) D1数据集下的流量检测ROC曲线 (*b*) D2数据集下的流量检测ROC曲线 (*c*) D3数据集下的流量检测ROC曲线

(*d*) D1数据集下的流量检测PRC曲线 (*e*) D2数据集下的流量检测PRC曲线 (*f*) D3数据集下的流量检测PRC曲线

图5 MalGuard和所有基线方法的ROC和PRC曲线

表6 MalGuard在数据集D1、D2、D3中不同不平衡比例下针对不同传输层协议(TCP、UDP)的评估结果 单位: %

β	传输层协议	D1			D2			D3		
		Pre	F1	AUC	Pre	F1	AUC	Pre	F1	AUC
4:1	TCP	88.89	94.12	99.91	88.11	93.68	97.64	91.00	93.43	99.46
	UDP	83.98	90.02	98.76	80.32	89.09	98.13	83.91	89.77	97.81
16:1	TCP	77.52	87.34	99.95	73.53	84.75	98.75	78.74	88.11	94.94
	UDP	71.43	83.33	95.75	64.10	78.13	99.45	71.43	83.33	97.11
48:1	TCP	68.53	80.66	98.04	60.01	74.52	99.95	68.86	81.39	98.58
	UDP	61.35	76.05	99.39	55.14	70.18	99.12	61.35	76.05	95.19

具体而言,在各类不平衡比例与数据集设置下,TCP协议上的检测性能整体优于UDP协议. 例如,在D1数据集β=4:1时,TCP协议在精确率、F1值与AUC上分别达到88.89%、94.12%与99.91%,而UDP协议下相应结果为83.98%、90.02%与98.76%. 这一差异源于TCP协议本身所具有的连接可靠性、按序传输等特性,使得其流量突发模式更为规整,更易于被检测模型捕捉. 而UDP协议无连接、不可靠的特性导致其流量模式波动较大,增加了特征学习难度.

6.4 效率评估

本节从训练时间和测试时间2个角度评估不同方法的时间成本. 通常,训练时间包含特征提取时间和分类器训练时间. 对于基于元学习的方法,还包括预训练时间. 测试时间是指已完成训练的分类器对待检测流量进行分类所用的时间. 结果如表7和表8所示.

如表7所示,因为MalGuard检测算法的低时间复杂度,其分类器训练时间最短. MalGuard特征提取时间和模型预训练时间较长,主要源于其双向编码器跨度表征的预训练模块计算开销,但该模块显著提升了检测精确率. 鉴于真实场景中预训练无需频繁执行,此开销是可接受的性能权衡.

如表8所示,MalGuard平均每条流量的测试时间仅为4.36 ms,仅次于YaTC. 由于MalGuard检测算法的时间复杂度$O(n\cdot t\cdot\log\psi)$满足实时性需求(t: N_estimators,ψ: max_samples),远低于基于距离或密度的方法,如局部离群因子算法(Local Outlier Factor, LOF)的时间复杂度为$O(n^2)$,这种效率使其能够支持大规模流量数据分析与处理. 由于YaTC采用轻量化的流量Trans-

表7　不同检测方法的训练时间　　单位: s

方法	特征提取时间	模型预训练时间	分类器训练时间	总时间
Kitsune[15]	22.05	-	26.71	48.76
ContraMTD[34]	176.72	-	323.01	499.73
CBSeq[31]	602.55	-	121.64	724.19
YaTC[11]	14.09	1923.50	76.55	2014.14
MalGuard	**364.52**	**2072.45**	**5.39**	**2442.36**

注:加粗数值表示MalGuard方法在对应数据集及不平衡比例下的平均Pre、F1值和AUC值.

表8　不同检测方法的测试时间　　单位: ms

方法	特征提取时间	检测时间	总时间
Kitsune[15]	3.29	3.55	6.84
ContraMTD[34]	5.53	0.02	5.55
CBSeq[31]	32.74	2.98	35.72
YaTC[11]	1.41	0.06	1.47
MalGuard	**4.34**	**0.02**	**4.36**

注:加粗数值表示MalGuard方法在对应数据集及不平衡比例下的平均Pre、F1值和AUC值.

former结构,且流量表示方法较为简单,因此其测试耗时最短.

6.5　消融实验

本节评估MalGuard中基于突发特征的流量词元化、基于双向编码器跨度表征的模型预训练、未知加密恶意流量检测3大主要模块的贡献. 在良性数据与恶意数据的不平衡比例β=4∶1的前提下,在数据集D3和D4上开展消融实验,实验结果如表9所示.

表9　MalGuard关键模块的消融实验　　单位: %

模块	变化	Pre	F1	AUC
流量词元化(模块1)	仅划分数据包,不划分突发	65.47	78.56	98.23
	划分突发,不提取突发时间间隔信息[10]	78.94	87.52	98.27
模型预训练(模块2)	替换为ET-BERT预训练模型[10]	84.37	91.46	98.68
恶意流量检测(模块3)	替换为单类支持向量机[57]	77.20	86.44	98.30
	替换为自编码器[58]	75.77	85.54	98.30
	替换为局部离群因子[59]	67.91	80.29	98.27
	替换为基于密度的聚类算法[54]	82.24	89.52	98.33
完整MalGuard	—	**88.68**	**92.70**	**98.52**

注:加粗数值表示MalGuard方法在对应数据集及不平衡比例下的平均Pre、F1值和AUC值.

基于突发特征的流量词元化模块. 为了探索MalGuard中基于突发特征的流量词元化模块的有效性,使用2种现有的流量词元化方法来替代本文提出的基于突发特征的流量词元化方法,分别为仅按照数据包粒度来划分词元与仅划分突发而不提取时间间隔信息[10]. 与使用基于突发特征的流量词元化方法相比,上述流量词元化方法的精确率分别下降23.21个百分点和9.74个百分点,F1值分别下降14.14个百分点和5.18个百分点. 这证明基于突发特征的流量词元化方法能够在时间维度上提取更多具有区分度的信息,从而提升基于双向编码器跨度表征模型对流量通用特征的学习能力.

基于双向编码器跨度表征的模型预训练模块. 为了探索MalGuard中基于双向编码器跨度表征的模型预训练模块的有效性,本文使用同样基于BERT模型的方法ET-BERT[10]的预训练模型来替代我们的预训练模型. 值得注意的是,两者的核心区别是跨度掩码方式和跨度边界目标的预训练任务,ET-BERT采用随机位置掩码的方式以及同源突发预测的预训练任务. 与基于双向编码器跨度表征的预训练模型相比,替换成ET-BERT预训练模型后,精确率和F1值分别下降了4.31个百分点和1.24个百分点. 这证明了我们的跨度掩码方式和跨度边界目标预训练任务更加适用于加密恶意流量检测领域,能够更好地解决加密流量数据报文内容因加密而高度随机化、局部字节语义微弱等问题,从而实现更可靠的未知加密恶意流量检测.

未知加密恶意流量检测模块. 在设计MalGuard时,选用孤立森林作为MalGuard的检测算法,主要基于该算法在异常检测任务中已被广泛验证的有效性. 然而,单类支持向量机[57](One-Class SVM)、自编码器[58](Autoencoder)、局部离群因子[59](LOF)与基于密度的聚类算法[54](DBSCAN)在异常检测领域也被证明表现优异. 为了评估这些无监督学习方法能否提升MalGuard的检测表现,我们用这些方法替代隔离森林算法. 与隔离森林算法相比,这些方法的精确率分别下降11.48、12.91、20.77和6.44个百分点,F1值分别下降6.26、7.16、12.41和3.18个百分点. 这是因为隔离森林算法基于异常点易被快速隔离的直观思想,而非依赖密度阈值或聚类中心距离,使它在稀疏区域和局部密度变化大的场景中表现出更强的鲁棒性.

7　讨论

本节讨论了MalGuard的潜在局限性以及未来研究方向.

高质量数据集稀缺. 尽管本文采用的恶意流量数据集的加密流量占比普遍不高,但它们仍然是当前加密恶意流量检测研究领域广泛采用的主流基准数据

集[10,11,20,31,44]. 这一现象本身也反映了该研究领域面临的一个普遍性挑战:目前学术界严重缺乏纯净的、标注完善且加密流量占比很高的公开网络恶意流量数据集. 同时,在真实网络环境中大规模采集加密恶意流量样本,也因涉及法律、隐私和安全性等严格限制而难以实现. MalGuard并未利用非加密流量中可能存在的任何明文信息,而是利用了有效涵盖了突发数据报文长度、时间等具有区分度元数据的突发数据报文和突发时间间隔特征. 因此,我们在这些加密与非加密混合型数据集上进行实验来评估MalGuard有效性的做法具有可行性. 未来,我们考虑在受控的实验环境中,通过由多台受控制的主机构建的小型仿真网络,主动生成并采集多种应用加密协议封装的恶意流量,从而构建更加纯净的高质量加密恶意流量数据集.

预训练通用性分析与模型更新策略. 首先对模型预训练的通用性进行分析,MalGuard的检测框架核心是自监督预训练与无监督异常检测的结合. 预训练阶段的目标是从无标签数据中学习一种通用的、与特定恶意标签无关的网络流量表征. 正如自监督学习在流量分类相关领域的研究[10~14]表明,通过这种方式学习到的特征表示往往具有良好的泛化能力和鲁棒性. 一旦通过预训练获得了这种通用的特征编码器,在面对场景更新时,无需进行完整的重新预训练. 预训练模型所学习到的通用特征较为稳定,足以支撑MalGuard有效运作. 其次,对于更显著的变更,例如,网络中新部署了采用全新协议的重要应用,导致流量特征分布发生较大变化,我们将设计一种基于部分数据重用的高效模型更新机制. 具体而言,仅采集新时期的部分无标签流量数据,与原有预训练数据混合,而后在已预训练模型的基础上进行增量预训练或针对性微调. 通过引入增量学习或模型遗忘与再学习策略,能够在不完全重新训练的情况下,使模型有效适应新数据特征,同时保留原有知识. 这种策略预计将比重新开始预训练显著减少计算开销和时间成本.

8 结论

本文提出了MalGuard,一种可靠的基于突发特征词元自学习的未知加密恶意流量检测方法. 具体而言,MalGuard利用良性流量和恶意流量的交互模式分析与关键特征分布观察,设计了一种基于流量突发特征的新型流量词元化表示方法,实现对数据报文语义信息与时序动态的关联表征. 基于新型流量词元化表征,MalGuard构建了基于双向编码器跨度表征的预训练模型,以学习大规模无标签数据具有泛化能力的通用特征,并构建基于隔离森林的轻量级无监督学习分类器,实现在无恶意标签数据下对加密恶意流量的可靠检测. 我们开展了充分的实验对MalGuard进行评估,结果证明MalGuard相比其他基线方法取得了更优的检测效果. 未来,我们将模拟受控实验环境采集高质量加密恶意流量数据集,并将设计更高效的模型预训练与微调架构,使其能够进一步提升检测效能.

参考文献

[1] WATCHGUARD. WatchGuard's Threat Lab analyzes the latest malware and internet attacks[EB/OL]. [2025-10-10]. https://www. watchguard. com/wgrd-resource-center/security-report-q1-2025.

[2] HOLLAND J, SCHMITT P, FEAMSTER N, et al. New directions in automated traffic analysis[C]//Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2021: 3366-3383.

[3] WANG N, SHI S H, CHEN Y M, et al. FeCo: Boosting intrusion detection capability in IoT networks via contrastive learning[J]. IEEE Transactions on Dependable and Secure Computing, 2025, 22(4): 4215-4230.

[4] ANDERSON B, MCGREW D. Identifying encrypted malware traffic with contextual flow data[C]//Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security. New York: ACM, 2016: 35-46.

[5] SHEN M, ZHANG J P, ZHU L H, 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.

[6] SHEN M, WU J H, AI J Y, et al. Swallow: A transfer-robust website fingerprinting attack *via* consistent feature learning[C]//Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2025: 1574-1588.

[7] SHEN M, JI K X, GAO Z B, et al. Subverting website fingerprinting defenses with robust traffic representation[C]// USENIX Security Symposium. California: USENIX Association, 2023

[8] SHEN M, JI K X, WU J H, et al. Real-time website fingerprinting defense via traffic cluster anonymization[C]//2024 IEEE Symposium on Security and Privacy. Piscataway: IEEE, 2024: 3238-3256.

[9] HE H Y, YANG Z G, CHEN X N. PERT: Payload encoding representation from transformer for encrypted traffic classification[C]//2020 ITU Kaleidoscope: Industry-Driven Digital Transformation. Piscataway: IEEE, 2020: 9303204.

[10] LIN X J, XIONG G, GOU G P, et al. ET-BERT: A contextualized datagram representation with pre-training transformers for encrypted traffic classification[C]//Proceedings of the ACM Web Conference 2022. New York: ACM, 2022: 633-642.

[11] ZHAO R J, ZHAN M W, DENG X W, et al. A novel self-supervised framework based on masked autoencoder for traffic classification[J]. IEEE/ACM Transactions on Networking, 2024, 32(3): 2012-2025.

[12] WANG T Z, XIE X H, WANG W D, et al. Netmamba: Efficient network traffic classification via pre-training unidirectional mamba[C]//2024 IEEE 32nd International Conference on Network Protocols. Piscataway: IEEE, 2025: 10858569.

[13] ZHOU G M, GUO X W, LIU Z T, et al. TrafficFormer: An efficient pre-trained model for traffic data[C]//2025 IEEE Symposium on Security and Privacy. Piscataway: IEEE, 2025: 1844-1860.

[14] QU J, MA X B, LI J F. TrafficGPT: Breaking the token barrier for efficient long traffic analysis and generation[EB/OL]. (2024-03-18) [2025-10-10]. https://arXiv. org/abs/2403.05822.

[15] MIRSKY Y, DOITSHMAN T, ELOVICI Y, et al. Kitsune: An ensemble of autoencoders for online network intrusion detection[EB/OL]. (2018-05-27) [2025-10-10]. https://arXiv.org/abs/1802.09089.

[16] CATILLO M, PECCHIA A, VILLANO U. CPS-GUARD: Intrusion detection for cyber-physical systems and IoT devices using outlier-aware deep autoencoders[J]. Computers & Security, 2023, 129: 103210.

[17] ZHANG P, HE F Z, ZHANG H, et al. Real-time malicious traffic detection with online isolation forest over SD-WAN[J]. IEEE Transactions on Information Forensics and Security, 2023, 18: 2076-2090.

[18] WANG K, STOLFO S J. Anomalous payload-based network intrusion detection[C]//Recent Advances in Intrusion Detection. Berlin: Springer, 2004: 203-222.

[19] FU C P, LI Q, SHEN M, et al. Realtime robust malicious traffic detection via frequency domain analysis[C]//Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2021: 3431-3446.

[20] FU C P, LI Q, XU K. Detecting unknown encrypted malicious traffic in real time via flow interaction graph analysis[EB/OL]. (2023-01-31)[2025-10-10]. https://arXiv.org/abs/2301.13686.

[21] RAMESH R, EVDOKIMOV L, XUE D W, et al. VPNalyzer: Systematic investigation of the VPN ecosystem[C]// Proceedings 2022 Network and Distributed System Security Symposium. Internet Society, 2022: 24285.

[22] STRATOSPHERE. Stratosphere laboratory datasets[EB/OL]. (2020-03-13) [2025-10-10] . https://www. stratosphereips.org/datasets-overview.

[23] NGUYEN T T T, ARMITAGE G. A survey of techniques for Internet traffic classification using machine learning[J]. IEEE Communications Surveys & Tutorials, 2008, 10(4): 56-76.

[24] GUPTA A, SHARMA L S. A categorical survey of state-of-the-art intrusion detection system-Snort[J]. International Journal of Information and Computer Security, 2020, 13(3/4): 337-356.

[25] CHIBA Z, ABGHOUR N, MOUSSAID K, et al. Newest collaborative and hybrid network intrusion detection framework based on suricata and isolation forest algorithm[C]//Proceedings of the 4th International Conference on Smart City Applications. New York: ACM, 2019: 1-11.

[26] DONG C, LU Z G, CUI Z L, et al. MBTree: Detecting encryption RATs communication using malicious behavior tree[J]. IEEE Transactions on Information Forensics and Security, 2021, 16: 3589-3603.

[27] LI H D, HU H X, GU G F, et al. vNIDS: Towards elastic security with safe and efficient virtualization of network intrusion detection systems[C]//Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2018: 17-34.

[28] TAYLOR V F, SPOLAOR R, CONTI M, et al. AppScanner: Automatic fingerprinting of smartphone apps from encrypted network traffic[C]//2016 IEEE European Symposium on Security and Privacy. Piscataway: IEEE, 2016: 439-454.

[29] PAPADOGIANNAKI E, IOANNIDIS S. A survey on encrypted network traffic analysis applications, techniques, and countermeasures[J]. ACM Computing Surveys, 2022, 54(6): 1-35.

[30] FU Z Q, LIU M X, QIN Y, et al. Encrypted malware traffic detection via graph-based network analysis[C]//Proceedings of the 25th International Symposium on Research in Attacks, Intrusions and Defenses. New York: ACM, 2022: 495-509.

[31] CUI S S, DONG C, SHEN M, et al. CBSeq: A channel-level behavior sequence for encrypted malware traffic detection[J]. IEEE Transactions on Information Forensics

and Security, 2023, 18: 5011-5025.

[32] CAVILLE E, LO W W, LAYEGHY S, et al. Anomal-E: A self-supervised network intrusion detection system based on graph neural networks[J]. Knowledge-Based Systems, 2022, 258: 110030.

[33] ZHANG Y X, WANG J D, CHEN Y Q, et al. Adaptive memory networks with self-supervised learning for unsupervised anomaly detection[J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(12): 12068-12080.

[34] HAN X Y, CUI S S, QIN J, et al. ContraMTD: An unsupervised malicious network traffic detection method based on contrastive learning[C]//Proceedings of the ACM Web Conference 2024. New York: ACM, 2024: 1680-1689.

[35] 轩勃娜, 李进. 基于改进CNN的恶意软件分类方法[J]. 电子学报, 2023, 51(5): 1187-1197.
XUAN B N, LI J. Malware classification method based on improved CNN[J]. Acta Electronica Sinica, 2023, 51(5): 1187-1197. (in Chinese)

[36] 谢丽霞, 魏晨阳, 杨宏宇, 等. 基于多维度动态加权alpha图像融合与特征增强的恶意软件检测方法[J]. 电子学报, 2025, 53(3): 849-863.
XIE L X, WEI C Y, YANG H Y, et al. Malware detection method based on multi-dimensional dynamic weighted alpha image fusion and feature enhancement[J]. Acta Electronica Sinica, 2025, 53(3): 849-863. (in Chinese)

[37] 刘新. 基于机器学习的恶意软件分析方法与智能检测技术研究[D]. 湘潭: 湘潭大学, 2014.
LIU X. Research on Analysis of Malware Based on Machine Learning and Intelligent Detection Technology[D]. Xiangtan: Xiangtan University, 2014. (in Chinese)

[38] 景鸿理, 黄娜, 李建国. 基于机器学习的恶意软件检测研究进展及挑战[J]. 信息技术与网络安全, 2020, 39(11): 38-44, 68.
JING H L, HUANG N, LI J G. Research progress and challenges of malware detection method based on machine learning[J]. Information Techology and Network Security, 2020, 39(11): 38-44, 68. (in Chinese)

[39] 李敏. 基于深度学习的恶意软件检测方法研究[D]. 北京: 华北电力大学, 2023.
LI M. Research on Malware Detection Method Based on Deep Learning[D]. Beijing: North China Electric Power University, 2023. (in Chinese)

[40] 郑锐, 汪秋云, 傅建明, 等. 一种基于深度学习的恶意软件家族分类模型[J]. 信息安全学报, 2020, 5(1): 1-9.
ZHENG R, WANG Q Y, FU J M, et al. A novel malware classification model based on deep learning[J]. Journal of Cyber Security, 2020, 5(1): 1-9. (in Chinese)

[41] LOTFOLLAHI M, JAFARI SIAVOSHANI M, SHIRALI HOSSEIN ZADE R, et al. Deep packet: A novel approach for encrypted traffic classification using deep learning[J]. Soft Computing, 2020, 24(3): 1999-2012.

[42] DEVLIN J, CHANG M W, LEE K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding[C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Kerrville: Association for Computational Linguistics, 2019: 4171-4186.

[43] GU A, DAO T. Mamba: Linear-time sequence modeling with selective state spaces[EB/OL]. (2024-05-31) [2025-10-10]. https://arXiv.org/abs/2312.00752.

[44] SHEN M, WU J H, YE K, et al. Robust detection of malicious encrypted traffic via contrastive learning[J]. IEEE Transactions on Information Forensics and Security, 2025, 20: 4228-4242.

[45] MARINO D L, WICKRAMASINGHE C S, RIEGER C, et al. Self-supervised and interpretable anomaly detection using network transformers[J]. IEEE Transactions on Industrial Informatics, 2025, 21(5): 4252-4261.

[46] KOUKOULIS I, SYRIGOS I, KORAKIS T. Self-supervised transformer-based contrastive learning for intrusion detection systems[EB/OL]. (2025-05-12) [2025-10-10]. https://arXiv.org/abs/2505.08816.

[47] CHANDOLA V, BANERJEE A, KUMAR V. Anomaly detection: A survey[J]. ACM Computing Surveys, 2009, 41(3): 1-58.

[48] TEGELER F, FU X M, VIGNA G, et al. BotFinder: Finding bots in network traffic without deep packet inspection[C]//Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies. New York: ACM, 2012: 349-360.

[49] SHARAFALDIN I, HABIBI LASHKARI A, GHORBANI A A. Toward generating a new intrusion detection dataset and intrusion traffic characterization[C]//Proceedings of the 4th International Conference on Information Systems Security and Privacy. SCITEPRESS - Science and Technology Publications, 2018: 108-116.

[50] LOPEZ A D, MOHAN A P, NAIR S. Network traffic behavioral analytics for detection of DDoS attacks[J]. SMU Data Science Review, 2019, 2(1): 14.

[51] KHRAISAT A, GONDAL I, VAMPLEW P, et al. Survey of intrusion detection systems: Techniques, datasets and

challenges[J]. Cybersecurity, 2019, 2(1): 20.
[52] JOSHI M, CHEN D Q, LIU Y H, et al. SpanBERT: Improving pre-training by representing and predicting spans[J]. Transactions of the Association for Computational Linguistics, 2020, 8: 64-77.
[53] HE L H, LEE K, LEVY O, et al. Jointly predicting predicates and arguments in neural semantic role labeling[EB/OL]. (2018-08-13)[2025-10-10]. https://arXiv.org/abs/1805.04787.
[54] GAN J H, TAO Y F. DBSCAN revisited: Mis-claim, unfixability, and approximation[C]//Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2015: 519-530.
[55] WANG W, ZHU M, ZENG X W, et al. Malware traffic classification using convolutional neural network for representation learning[C]//2017 International Conference on Information Networking. Piscataway: IEEE, 2017: 712-717.
[56] KOUKIS D, ANTONATOS S, ANTONIADES D, et al. A generic anonymization framework for network traffic[C]//2006 IEEE International Conference on Communications. Piscataway: IEEE, 2006: 2302-2309.
[57] SCHÖLKOPF B, PLATT J C, SHAWE-TAYLOR J, et al. Estimating the support of a high-dimensional distribution[J]. Neural Computation, 2001, 13(7): 1443-1471.
[58] VINCENT P, LAROCHELLE H, LAJOIE I, et al. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion[J]. Journal of Machine Learning Research, 2010, 11: 3371-3408.
[59] BREUNIG M M, KRIEGEL H P, NG R T, et al. LOF: Identifying density-based local outliers[C]//Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2000: 93-104.

作者简介

沈　蒙　男,1988年1月出生于山东省德州市. 现为北京理工大学网络空间安全学院教授,博士生导师. 主要研究方向为网络加密流量分析、数据隐私安全、区块链应用. 中国电子学会会员编号:E190019899M.
E-mail: shenmeng@bit.edu.cn

杨　明　男,1981年3月出生于山东省东营市. 现为齐鲁工业大学(山东省科学院)、山东省计算中心(国家超级计算济南中心)研究员,博士生导师. 主要研究方向为数据安全、人工智能安全.
E-mail: yangm@sdas.org

贾冀哲　男,1997年2月出生于辽宁省沈阳市. 现为北京理工大学网络空间安全学院博士研究生. 主要研究方向为网络加密流量分析、恶意流量检测.
E-mail: jiajizhe@bit.edu.cn

任琛琛　女,2003年4月出生于河北省保定市. 现为北京理工大学网络空间安全学院硕士研究生. 主要研究方向为网络加密流量分析、恶意流量检测、网站指纹攻击.
E-mail: chenchenren@bit.edu.cn

赵卜凡　男,2002年11月出生于山东省德州市. 现为北京理工大学网络空间安全学院博士研究生. 主要研究方向为网络加密流量分析、恶意流量检测.
E-mail: zhaobufan@bit.edu.cn

宋　悦　男,1987年7月出生于北京市. 现为天翼安全科技有限公司基础能力部基础研究组组长. 主要研究方向为网络安全技术.
E-mail: songy7@chinatelecom.cn

常力元　男,1984年2月出生于吉林省吉林市. 现为中国电信集团首席专家、天翼安全科技有限公司副总工程师. 主要研究方向为网络安全技术. 中国电子学会会员编号:E190087084M.
E-mail: changly@chinatelecom.cn

祝烈煌　男,1976年9月出生于浙江省衢州市. 现为北京理工大学网络空间安全学院教授,博士生导师. 主要研究方向为密码学、网络和信息安全. 中国电子学会会员编号:E190010255M.
E-mail: liehuangz@bit.edu.cn