计算机科学
COMPUTER SCIENCE

面向恶意流量识别的网络流量生成方法

张灿, 栗维勋, 汪明, 詹雄, 颉子光, 韩东岐, 王之梁, 杨家海

引用本文

张灿, 栗维勋, 汪明, 詹雄, 颉子光, 韩东岐, 王之梁, 杨家海. 面向恶意流量识别的网络流量生成方法[J]. 计算机科学, 2026, 53(4): 415-423.

ZHANG Can, LI Weixun, WANG Ming, ZHAN Xiong, XIE Ziguang, HAN Dongqi, WANG Zhiliang, YANG Jiahai. Network Traffic Generation Method for Malicious Traffic Identification[J]. Computer Science, 2026, 53(4): 415-423.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.250900139

面向恶意流量识别的网络流量生成方法

张 灿[1] 栗维勋[2] 汪 明[3] 詹 雄[3] 颉子光[4] 韩东岐[1] 王之梁[1] 杨家海[1]
1 清华大学网络科学与网络空间研究院 北京 100084
2 国网河北省电力有限公司 石家庄 050031
3 国家电网有限公司 北京 100031
4 国网陕西省电力有限公司 西安 710048
(zhangcan25@mails.tsinghua.edu.cn)

摘 要 恶意流量识别是网络安全防护中的关键任务,训练数据的质量直接决定识别模型的准确性。然而,受隐私保护、标注成本和类别不均衡等因素限制,真实数据获取十分困难。为解决上述挑战,提出了一种基于预训练-微调模型的细粒度网络流量生成方法。该方法首先设计了一种保留协议结构信息的静态分词方案,将原始流量转换为协议语义保持的可供自回归模型学习的序列表示。在此基础上,构建了预训练-微调的两阶段生成框架:先以大规模良性流量学习通用协议与时序模式,继而在标注的恶意流量上进行任务定向微调,生成具备明确攻击语义的高保真样本。为了验证流量生成方法的效果,设计了多个维度的实验评估,结果证明,所提方法在协议合规性(领域专家知识检查通过率高达99.95%)、分布相似性(生成/真实分布间推土机距离仅为0.0059)及生成多样性(真实邻域覆盖度超过50%)均优于主流基准模型;在使用生成流量训练的恶意流量识别任务中,相较于基准方法,所提方法唯一实现了多种分类器的检测效果提升。此外,设计了恶意功能验证实验,在两种攻击场景下验证了所提方法生成流量的攻击效果。实验结果表明,所提方法能够生成语法合规、统计相似且语义功能正确的细粒度恶意流量,为解决网络安全领域流量数据稀缺问题提供了有效的技术途径。

关键词:网络流量生成;恶意流量识别;生成式人工智能;自回归模型;预训练-微调
中图分类号 TP311

Network Traffic Generation Method for Malicious Traffic Identification

ZHANG Can[1],LI Weixun[2],WANG Ming[3],ZHAN Xiong[3],XIE Ziguang[4],HAN Dongqi[1],WANG Zhiliang[1] and YANG Jiahai[1]
1 Institute of Network Sciences and Network Space,Tsinghua University,Beijing 100084,China
2 State Grid Hebei Electric Power Company,Shijiazhuang 050031,China
3 State Grid Corporation of China,Beijing 100031,China
4 State Grid Shaanxi Electric Power Company,Xi'an 710048,China

Abstract Malicious traffic identification is a key task in cybersecurity,and the quality of training data directly determines the accuracy of detection models. However,obtaining real traffic data is challenging due to privacy concerns,high annotation costs,and class imbalance. To address these challenges,this paper proposes a fine-grained network traffic generation method based on a pre-training-fine-tuning paradigm. The method firstly introduces a static tokenization scheme that preserves protocol structure information,converting raw traffic into sequence representations that maintain protocol semantics and are suitable for autoregressive model learning. On this basis,a two-stage generation framework is constructed:pre-train on large-scale benign traffic to capture general protocol and temporal patterns,then fine-tune on task-specific labeled malicious traffic to generate high-fidelity samples with explicit attack semantics. To evaluate the effectiveness of the proposed method,multi-dimensional experiments are conducted. The results show that the method outperforms mainstream baselines in protocol compliance(achieving a 99.95% pass rate in expert knowledge checks),distribution similarity(with an Earth Mover's Distance of 0.0059 between generated and real distributions),and generation diversity(with real neighborhood coverage exceeding 50%). In malicious traffic identification tasks,the generated traffic uniquely improves the detection performance of multiple classifiers compared with baseline methods. In addition,malicious functionality verification experiments confirm that the generated traffic successfully reproduces attack effects in two attack scenarios. Overall,the results demonstrate that the proposed method can generate fine-grained malicious traffic that is syntactically compliant,statistically consistent,and semantically functional,providing an effective technical approach to alleviate the data scarcity problem in cybersecurity.

到稿日期:2025-09-23 返修日期:2025-12-18
基金项目:国家电网有限公司科技项目(5108-202413050A-1-1-ZN)
This work was supported by the Science and Technology Project of State Grid Corporation of China(5108-202413050A-1-1-ZN).
通信作者:韩东岐(handongqi@bupt.edu.cn)

Keywords Network traffic generation, Malicious traffic identification, Generative AI, Autoregressive model, Pre-training and fine-tuning

1 引言

随着互联网技术的飞速发展与网络规模的持续扩张，网络流量呈现指数级增长和日益多样化的特点。这些流量数据在蕴含丰富用户行为与网络状态信息的同时，也成为了各类网络攻击和恶意行为的载体，给网络安全防护带来了严峻挑战。因此，有效识别恶意流量已成为网络安全领域的关键课题[1-4]。然而，恶意流量识别系统的有效性往往依赖于高质量、具有代表性的流量数据集，而真实恶意流量数据因隐私限制、标注成本高昂及固有的类别不平衡问题而较难获取[5]。

为此，网络流量生成技术应运而生，旨在通过合成模拟流量数据为模型训练提供关键支撑。尽管流量生成领域取得了一定进展，但现有方法仍存在显著局限性。一方面，传统的流量生成方法严重依赖领域专家知识进行参数配置[6]，难以适应动态多变的网络环境，导致泛化性能不足。另一方面，尽管基于生成式人工智能的方法展现出巨大潜力[7-9]，但现有研究大多聚焦于粗粒度的流量特征模拟，普遍存在协议规范性不足、时序特征建模能力弱等问题，生成的流量与真实通信场景仍有较大差距，限制了其在复杂恶意流量识别任务中的应用价值。

针对上述问题，本文聚焦于面向恶意流量识别的网络流量生成方法研究，旨在构建一种能够生成高保真、多样化且具备精确时序与结构特征的网络流量数据的方法。本文以自回归预测机制为基础，设计了面向协议结构的静态表示与分词方案，并结合预训练-微调范式，逐步学习通用通信规律并适配特定攻击语义，从而实现语法合规、分布真实且多样的恶意流量生成。

本文的主要贡献如下：

1）提出一种细粒度的网络流量表示与分词方案，能够在保持语义完整性的同时有效建模时序与结构特征；

2）构建预训练-微调的两阶段流量生成框架，在大规模良性流量上学习通用模式，并通过恶意流量的任务定向微调，实现无需人工规则的高保真恶意流量生成；

3）建立覆盖协议合规性、统计保真度、生成多样性与下游任务性能的多维度评估体系，并首次通过恶意功能验证实验，证明流量生成方法在协议语义方面的有效性。

2 相关工作

2.1 传统网络流量生成方法

传统网络流量生成方法主要包括仿真驱动与模型驱动两类技术路线。

仿真驱动方法依托于网络仿真器（如 NS-3[10]，DYNAMO[11]等），通过构建网络拓扑与模拟协议行为，实现对节点间通信过程的高度可控复现。该方法虽能够精确建模网络协议栈，但其有效性高度依赖专家经验，需对网络结构与参数进行大量手动配置，因此难以高效生成多样化且规模庞大的真实流量数据。

模型驱动方法则基于预设的数学模型（如泊松分布）对流量行为进行抽象与模拟，代表性工具有 LitGen[12]，MGEN[13]等。该方法通过调节分布参数，能够在统计层面较好地拟合实际网络流量的负载特征，因此被广泛用于早期研究中，尤其适用于对宏观流量特性（如突发性与间歇性）进行建模。然而，由于其生成能力受限于模型假设，难以准确反映真实环境中复杂的协议交互、行为依赖及恶意模式，因此在应用层面存在明显局限性。

2.2 基于生成式人工智能模型的流量生成方法

近年来，随着深度学习技术的快速发展，基于生成式人工智能的网络流量生成方法逐渐成为重要研究方向。当前，基于生成对抗网络（Generative Adversarial Networks，GAN）、自回归模型（Autoregressive Model，AR）和扩散模型（Diffusion Model，DM）的方法已取得显著进展。图 1 从宏观上展示了这 3 类模型的通用流程，主要包括数据预处理、模型生成、生成数据及应用 4 个阶段。预处理阶段将原始流量（如 NetFlow 记录或 PCAP 文件）转换为模型可以处理的格式；将生成阶段利用上述 3 类模型合成数据；将生成结果按粒度划分为粗/细粒度数据[14]，最终可应用于流量分析与数据增强等下游任务。表 1 进一步对比了各类方法的代表性工作[15-25]。

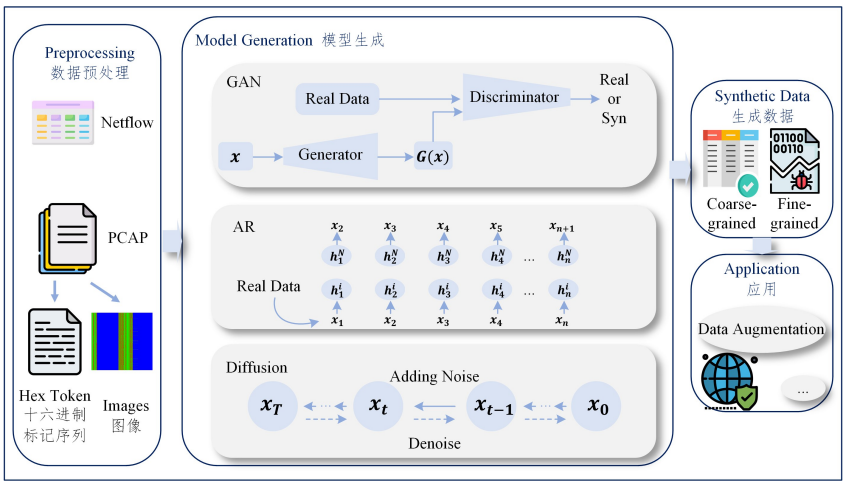

图 1 基于生成式人工智能的网络流量生成方法流程图

Fig. 1 Flowchart of network traffic generation method based on generative artificial intelligence

表 1　基于生成式人工智能网络流量生成方法的对比

Table 1　Comparison of network traffic generation methods based on generative artificial intelligence

方法	代表性工作	生成数据粒度	特点/优势	局限性
生成对抗网络	WGAN[15],DoppelGANger[16],NetShare[17]	粗粒度	擅长捕获流级别统计特征	不适合细粒度数据生成
自回归模型	NetGPT[20] Lens[21]	粗粒度	在网络流量理解和基本生成任务上表现出色	目前工作局限于报头字段的粗粒度生成
扩散模型	NetDiffus[23],NetDiff[24] NetDiffusion[25]	粗粒度/细粒度	训练过程更稳定;NetDiffusion 模型能生成细粒度数据	生成效果对图像结构依赖性强且细粒度生成依赖复杂后处理,无法生成时间戳

其中,粗粒度数据通常指通过聚合或统计方式提取的流量特征。如图 1 中的“数据预处理”阶段所示,该类数据的获取主要有两条路径:1)可以直接来源于本身,即统计摘要的 NetFlow;2)可以通过对原始的 PCAP 文件进行处理,通过计算统计指标来聚合生成。此类表示能够有效捕捉流量数据的统计特征与行为模式,同时在一定程度上天然规避用户隐私问题,适用于大规模流量分析与行为建模。然而,该类方法难以保留数据包间的细粒度依赖关系和协议语义。

细粒度数据则严格保持原始数据包的字节级结构与协议栈层次,不仅完整保留各层协议头部的字段信息,还严格维持协议内部及数据包间的结构依赖与时间关系(如包间时延)。因此,该类数据通常需要更大的存储空间和更高的计算复杂度。

2.2.1　基于生成对抗网络的流量生成方法

生成对抗网络(GAN)由 Goodfellow 等[26]于 2014 年提出,包含生成器与判别器两个模块,通过对抗训练提升生成样本的质量。该类方法目前主要集中于粗粒度流量生成。例如,Ring 等[15]采用 WGAN 结合 IP2Vec 等编码技术,将分类特征(如 IP 地址)转换为连续表示,以合成流级统计特征。DoppelGANger[16]提出分层生成架构,将元数据与时间序列解耦,以捕捉流级别相关性。NetShare[17]将报文头生成问题建模为时间序列生成任务,采用混合编码策略处理 IP、端口与协议字段,在保真度与隐私性之间取得平衡,但仍未涵盖状态会话语义、应用层协议细节及精确包间时序等细粒度属性。总体而言,GAN 方法在细粒度生成方面存在固有局限:其更关注流级统计特征,且时间建模粒度较粗,难以捕获精确的数据包间时序关系。

2.2.2　基于自回归模型的流量生成方法

自回归模型将联合分布分解为条件概率序列,并通过已生成内容逐步预测下一元素。Transformer[18]及 GPT[19]等基于自注意力机制的大规模预训练语言模型,进一步推动了自回归在序列生成中的应用。Meng 等[20]提出的 NetGPT 基于 GPT-2 架构,通过字节级十六进制编码,将流量映射为标记序列,用于报头字段生成。类似地,基于 T5 的 Lens 模型[21]也侧重于源/目的 IP、端口及包长等粗粒度特征的生成。尽管这类方法在流量理解与生成任务中表现良好,但仍主要局限于粗粒度层面,尚未实现完全符合协议规范的细粒度流量合成。

2.2.3　基于扩散模型的流量生成方法

扩散模型通过前向加噪与反向去噪过程学习数据分布[22],具有训练稳定性和生成多样性的优势。在粗粒度生成方面,NetDiffus[23]将一维时间序列转换为 GASF 图像,以捕捉时间特征;NetDiff[24]则采用分层扩散模型对用户服务使用模式进行建模。在细粒度生成中,NetDiffusion[25]将原始报文转换为图像表示,并基于改进的 Stable Diffusion 模型生成数据,通过控制生成与后处理策略维持协议字段的合规性。然而,该方法对图像表示结构较为敏感,可变长数据包下的填充区域可能干扰有效字节建模,且仍需依赖复杂后处理确保协议一致性。

3　本文方法设计与实现

3.1　本文方法的基本思想

在图 1 所示的通用框架基础上,本文聚焦于基于自回归模型的技术路线,并提出了一种在预训练-微调范式下,专用于生成高保真度、细粒度恶意网络流量的具体实现方案。其详细架构如图 2 所示,由 3 个主要部分组成,即数据预处理、预训练和微调。每个部分都发挥着关键作用,以使模型能够学习并复现网络流量的复杂特征——从基础的协议语法到特定的恶意行为。

在数据预处理阶段,原始网络流量被系统性地转换为结构化的词元(Token)序列。该过程首先将字节流分割为统一的两字节十六进制字,然后使用一个静态且完备的词汇表进行映射。该方案确保协议字段的语义和结构完整,能够精确反映底层数据结构,并避免了由动态分词方法所带来的歧义。

在预训练阶段,模型在一个大规模、多样化的无标签通用网络流量语料库上进行训练,采用自监督学习目标,通过预测序列中的后续词元来学习捕捉网络通信中固有的基础语法、统计模式和时序依赖关系。这种大规模训练使模型能够建立对各种网络协议的泛化且鲁棒的理解,为生成语法合规的流量奠定了坚实的基础。

最后,在微调阶段,预训练好的模型将针对恶意流量生成这一特定下游任务进行适配。使用一个代表目标攻击场景的、规模更小的已标注数据集,来进一步优化模型的参数,以捕捉恶意流量所独有的、细微的行为特征和语义特性。这一策略性的适配阶段,确保了在预训练期间所学到的基础知识能够被精确地调整,以用于复现真实的攻击模式。

本章的其余部分将详细阐述每个组件的设计与实现,并说明各个阶段如何共同作用。

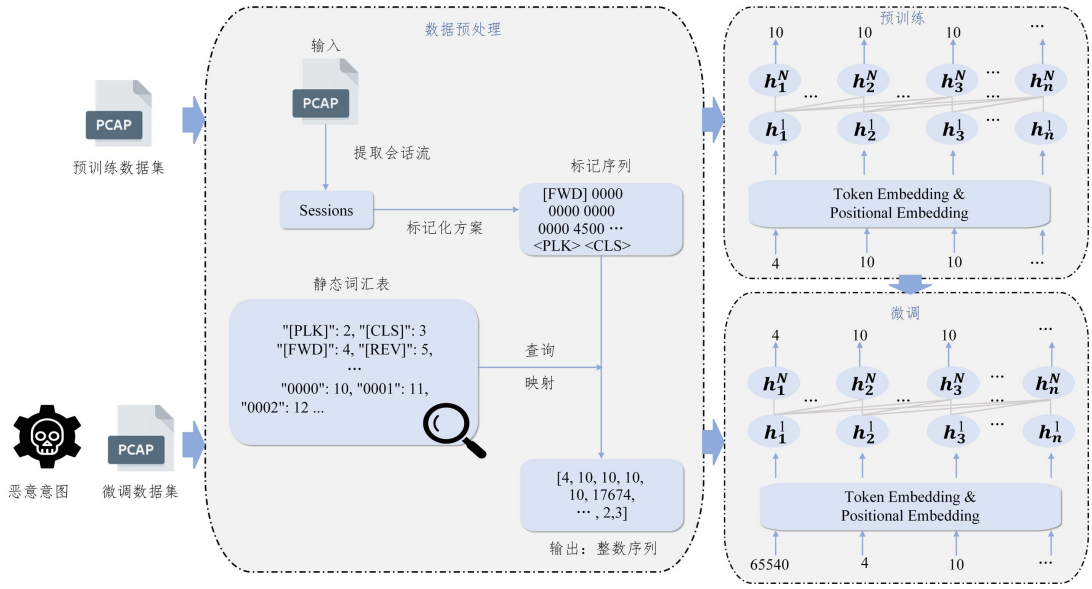

图 2 本文模型的架构

Fig. 2 Architecture of the proposed model

3.2 数据预处理

将原始网络流量应用于生成式模型的关键前提是，将其从字节流形态转换为模型可处理的结构化格式。自然语言处理（NLP）领域的标准分词方法，如字节对编码（BPE）或 WordPiece，由于是基于频率的合并策略，从根本上不适用于网络流量。这些方法无视网络协议中固有的、明确的字段边界，会导致字段切分与语义割裂等问题，从而破坏数据的结构完整性。

为克服这些局限性，本文提出了本文的核心贡献之一：一种细粒度的网络流量表示与分词方案。该方案旨在生成一种能够精确反映底层数据结构、避免歧义性的序列表示，专门用于应对网络协议数据的高度结构化和语义化特性。具体而言，该方案由会话流提取、标记化方案设计，以及词汇表构建与映射 3 个紧密衔接的步骤组成。

3.2.1 会话流提取

数据预处理的初始步骤是会话流提取。本文依据网络通信的“五元组”（源 IP 地址、目的 IP 地址、源端口、目的端口、传输层协议），对原始 PCAP 数据集中的全部数据包进行分组。通过此方式，数据被重构为一系列按时间顺序排列的、逻辑上独立的会话流。为确保各会话流的边界清晰，每条流的末尾均以一个特殊的会话结束标记[CLS]作为终止符。

3.2.2 标记化方案

在提取会话流之后，每个流都将通过一个全面的标记化方案进行处理，该方案旨在显式地编码其结构、时序和内容特征。该方案由以下 4 类标记组成。

1）数据包开始标记：该标记用于明确标识数据包的起始及其在会话流中的方向性。[FWD]表示正向（源到目的）数据包，[REV]表示反向（目的到源）数据包。为统一规范，每个流的首个数据包默认标记为[FWD]。

2）时间间隔标记：为捕捉流量的时间动态，此标记表示当前数据包与会话流起始时间的间隔。其计算方式为：首先获取当前数据包时间戳与流首个数据包时间戳的差值（精确至微秒），随后将该差值转换为十六进制字符串，并按每两个字节进行分组。

3）十六进制内容标记：为完整保留协议信息，数据包的关键协议层（主要为 IP，TCP/UDP，HTTP）报头内容被整体转换为十六进制字符串序列，并同样按两个字节分组。为降低模型在高度动态或确定性计算字段上的学习难度，本文对部分此类字段（如 Checksum，IP Identification 等）采用特殊的占位符标记（如[checksum]，[identification]）进行替代。

4）数据包结束标记：每个数据包的标记序列均以[PLK]标记结束，以明确划分数据包边界，保证序列结构的清晰性。

3.2.3 词汇表构建和映射

上述标记化方案产出的是一个由字符串构成的序列。预处理的最后一步，是利用一个静态且完备的词汇表，将此序列映射为模型可处理的整数序列。该词汇表由十六进制词汇表和特殊标记词汇表两部分构成。

1）十六进制词汇表：包含从 0000 到 ffff 的全部 65 536 个可能的四字符十六进制字符串。该完备集合确保了所有十六进制内容标记和时间间隔标记都能找到唯一的对应。

2）特殊标记词汇表：包含所有在前述方案中定义的结构化及占位符标记，如[CLS]，[FWD]，[REV]，[PLK]，以及各类占位符。

最终，每个会话流都被转换为一个整数序列，作为后续预训练与微调阶段的输入。为清晰起见，本文方案的整体流程、数据形态变换及关键参数总结如算法 1 所示。

算法 1 数据预处理

输入：原始网络流量 PCAP 文件 P；包含所有特殊标记和十六进制词汇的静态词汇表 V

输出：可供模型训练的整数序列集合 S

```
S←∅
G←Group(P) // 按五元组分组
for each g in G do
    T←[]
    t0←g.packets[0].timestamp
    for each p in g do
```

```
7.      d←Dir(p) // [FWD]或[REV]
8.      Δt←Intv(p.timestamp-t0) // 时间间隔
9.      h ← Hex(p.header) // 十六进制标记
10.     e ← '[PLK]'//包结束标记
11.     Pt←d ‖ Δt ‖ h ‖ e // 拼接包标记
12.     T.add(Pt)
13.   end for
14.   T.append([CLS])// 会话结束
15.   I←Map(T,V)// 映射为整数序列
16.   S.add(I)
17. end for
18. return S
```

3.3　预训练

预训练阶段的目标在于，通过对大规模网络流量数据进行自监督学习，构建一个能够捕捉流量基础模式与内在规律的泛化模型。该模型旨在为后续的下游任务提供一个强有力的特征表示基础。

本文采用 GPT-2 模型架构[19]作为基础模型，其核心机制为自回归预测，即基于给定的前序标记序列来预测下一个标记。该机制与网络流量的顺序性生成过程高度契合，能够有效建模协议字段之间以及数据包之间的序列依赖关系；且相较于参数量巨大的通用大预言模型，GPT-2 具有更适中的模型规模，在推理效率和训练开销方面更具优势，尤其适用于网络流量这类领域特定、语义结构相对规整的数据。在本研究中，选用 GPT-2 Small 模型，其参数量为 1.24 亿，由一个 Transformer 解码器堆栈构成，具体包含 12 个解码器层、768 维的隐藏层维度以及 12 个注意力头。该模型可以在保证模型泛化能力的同时平衡训练成本。此外，流量生成任务并不依赖通用自然语言理解与生成能力，因此也无须引入过于庞大的预训练模型。

在形式化表示方面，给定一个经过数据预处理后生成的整数序列，对应于图 2 中的“输出：整数序列”，例如将[4，10，10，…]表示为 $\boldsymbol{T}=\{t_1,t_2,\cdots,t_n\}$，如图 2 所示，模型在每个时间步接收前序词元$\boldsymbol{t}_{<i}$，并预测下一个词元$t_i$。模型的训练目标是最大化该序列的联合概率，基于链式法则，该联合概率可被分解为一系列条件概率的乘积。

$$P(\boldsymbol{T};\boldsymbol{\theta})=\prod_{i=1}^{n}P(t_i\mid \boldsymbol{t}_{<i};\boldsymbol{\theta}) \tag{1}$$

其中，$\boldsymbol{\theta}$ 代表在此阶段被训练的模型参数；$P(t_i\mid \boldsymbol{t}_{<i};\theta)$表示在给定前序序列$\boldsymbol{t}_{<i}=\{t_1,\cdots,t_{i-1}\}$的条件下，模型预测下一个词元为$t_i$的条件概率。这一概率的计算，正是通过图 2 所示的内部隐藏状态 h 来实现的。具体而言，GPT-2 模型首先将输入的离散词元序列$\boldsymbol{t}_{<i}$编码为一系列连续的高维隐藏状态向量$\boldsymbol{h}_{<i}$，如图 2 中从底层到顶层的$\boldsymbol{h}^1$至$\boldsymbol{h}^N$所示。这些隐藏状态编码了丰富的上下文信息，最终模型利用最新的隐藏状态h_{i-1}来计算一个覆盖整个词汇表的概率分布，并从中得到预测下一个词元为t_i的概率。

在实际训练过程中，模型通过最小化训练语料库的负对数似然来进行优化。该损失函数的定义如下：

$$\mathcal{L}_{\mathrm{NLL}}(\boldsymbol{\theta})=-\sum_{i=1}^{n}\log P(t_i\mid \boldsymbol{t}_{<i};\boldsymbol{\theta}) \tag{2}$$

通过在整个大规模语料库上最小化该损失函数，模型被驱动去学习网络流量数据中内含的复杂语法规则、语义关联以及时序依赖，最终形成一个对网络流量具有深度理解能力的基础模型。具体算法流程如算法 2 所示。

算法 2　预训练

输入：预训练数据集（经过数据预处理得到的整数序列集合）D_p；初始化的 GPT-2 模型 M；预训练超参数H_p

输出：表示预训练模型权重W_p

```
1. M.init()// 初始化模型参数
2. for e=1 to Hp.epochs do
3.     for each B in Dp do
4.         L←Loss(M,B)// 下一词预测损失
5.         Update(M,L)
6.     end for
7. end for
8. W_p←M.weights()
9. return Wp
```

3.4　微调

预训练阶段使模型获得了对网络流量语法和通用模式的泛化理解，但该模型本身并未针对任何特定任务进行优化，因此难以直接生成具备特定意图的恶意流量。为实现这一目标，必须通过微调，将模型的泛化能力向目标任务进行特化。

微调阶段的核心任务是将预训练模型调整为面向恶意流量生成的专用模型。此阶段延续了与预训练相同的自回归建模框架与损失函数，但训练数据则替换为经过精细标注的、涵盖多种攻击类型的高质量恶意流量数据集。

为实现对生成过程的可控性，将代表特定恶意意图的文本提示置于每个恶意流量样本序列的起始位置。因此，微调阶段的输入是一个以恶意意图提示开头的整数序列，且模型参数以预训练阶段的最终权重为初始化。在微调过程中，模型学习的是在给定恶意意图提示的条件下，生成相应流量序列的概率。通过在任务特定数据上的进一步优化，模型能够逐步建立起恶意语义提示与流量结构、时序特征之间的精确映射关系，其内部隐藏状态不仅能编码流量的上下文信息，更能将恶意意图提示的语义信息融入其中，从而使其具备根据指定意图生成相应类别恶意流量的能力。算法的具体流程如算法 3 所示。

算法 3　微调

输入：微调数据集（经过数据预处理得到的整数序列集合）D_f；恶意意图提示集合P_i；预训练模型权重W_p；微调超参数H_f

输出：微调模型权重W_f

```
1. M.load(Wp)// 初始化模型参数
2. Dt←Prepend(Df,Pi)
3. for e=1 to Hf.epochs do
4.     for each B in Dt do
5.         L←Loss(M,B)// 下一词预测损失
6.         Update(M,L)
7.     end for
8. end for
9. Wf←M.weights()
10. return Wf
```

本文所提出的流量生成方法由数据预处理、模型预训练和模型微调 3 个核心阶段构成，其完整流程总结如下。

首先，输入原始的 PCAP 网络流量数据。通过执行数据

预处理(算法 1),对数据进行系统性的转换,该过程包括会话流提取、设计并应用一套细粒度的标记化方案,以及最终的词汇表映射。此阶段的输出是两种可供模型训练的整数序列语料库:一个用于预训练的大规模通用流量语料库和一个用于微调的、带有特定标签的恶意流量语料库。

其次,加载大规模通用流量语料库。通过执行模型预训练(算法 2),在一个通用的、无标签的流量数据集上对 GPT-2 模型进行自监督学习。此阶段的目标是让模型学习网络流量通用的语法结构、时序规律和统计模式,其输出为一套能够理解通用流量的、泛化能力强的预训练模型权重。

最后,加载第二阶段产出的预训练模型权重,并准备带有恶意意图提示的恶意流量语料库。通过执行模型微调(算法 3),将模型的泛化知识适配到特定的恶意流量生成任务上。此阶段的输出是一个专用的生成模型,它能够接收用户的恶意意图指令作为输入,并生成符合该意图的高保真恶意网络流量序列。

4 实验与结果分析

4.1 实验设置

为进行实证评估,本文采用 CIC-IoT-2023[27] 数据集。该数据集由加拿大网络安全研究所发布,是一个面向物联网环境的、综合性的网络安全数据集。为全面评估所提方法的有效性,本文从该数据集中选取了良性流量以及 4 种具有代表性的恶意流量类别进行实验。具体的实验数据集构成如下:预训练数据集由约 30 万条良性网络会话流构成,用于支撑模型在预训练阶段学习网络通信的通用模式与内在规律;微调数据集则由约 8 万条恶意流量构成,涵盖分布式拒绝服务攻击(DDoS)、暴力破解攻击(Brute-force)、欺骗攻击(Spoofing)和侦察攻击(Reconnaissance)这 4 种攻击类型。

为提供全面的性能比较,本文选取了 4 种网络流量生成模型作为基准,分别是 DoppelGANger[16],NetShare[17],NetGPT[20],NetDiffusion[25]。选择这 4 种模型主要基于两个标准:1)它们的源代码均公开可用;2)它们的模型设计适用于恶意流量分类的场景要求。

在实验的评估阶段,本文训练完成的模型生成一个包含约 12 万条恶意流量的合成数据集,该数据集中的流量类别分布与微调数据集保持一致。此生成数据集,连同由各基准模型生成的对应数据集,将被用于后续章节中定义的各项指标的评估。

4.2 评估指标

为全面评估生成数据的质量,本文设计了一套多维度的评估体系,从领域专家知识检查、分布相似性、下游任务性能和生成多样性 4 个关键维度进行综合考量。除了这些评估指标之外,还包括专门设计的实验,以验证本文方法生成的流量的恶意功能。

4.2.1 领域专家知识检查

领域专家知识检查系统性地评估生成的网络流量是否符合协议规范和网络规则,具体的检查项如下:

1)验证数据包长度是否符合协议规范;

2)每一条生成流量只能有一种传输层网络协议;

3)验证服务端口与其对应协议之间的关系,例如确保 DNS 流量(端口 53)使用 UDP 协议;

4)验证与时间相关的字段的逻辑一致性,从而确保在适用的情况下,流持续时间是非负的,并且多包流具有非零的持续时间。

该维度的最终量化指标为检查通过率(Compliance Rate),定义为通过上述所有规则检查的流量数量占总生成流量的比例。

4.2.2 分布相似性

针对离散型数据(如协议类型),本文采用 Jensen-Shannon 散度(JSD)进行度量;针对连续型数据(如流持续时间),则采用推土机距离(Earth Mover's Distance, EMD)进行度量。

Jensen-Shannon 散度的计算式如下:

$$JSD(R \parallel G)=\frac{1}{2}D_{\mathrm{KL}}(R \parallel M)+\frac{1}{2}D_{\mathrm{KL}}(G \parallel M) \tag{3}$$

其中,R 和 G 分别是真实数据和生成数据的概率质量函数,$M=\frac{1}{2}(R+G)$,D_{KL} 是 Kullback-Leibler 散度。

推土机距离的计算式为:

$$EMD(R,G)=\inf_{\gamma\in\Gamma(R,G)}\int_{\mathbb{R}\times\mathbb{R}}|x-y|\,\mathrm{d}\gamma(x,y) \tag{4}$$

其中,$\Gamma(R,G)$ 表示 R 和 G 的所有可能联合分布的集合。

JSD 的取值范围为 $[0,1]$,而 EMD 的取值范围为 $[0,+\infty)$,两者均是值越小,表示两个分布越相似。

4.2.3 下游任务性能

该维度通过在下游的恶意流量识别任务中的表现,来检验生成数据的实用价值。评估主要采用以下两种方法。

1)在生成数据上训练,在真实数据上测试(Train on Synthetic, Test on Real, TSTR):在生成的合成数据上训练一个恶意流量分类模型,然后在真实的测试集上评估其性能。

2)数据增强有效性评估:将生成数据加入到原始训练集中,评估在此增强数据集上训练的分类模型,相较于仅使用原始数据训练的模型,其性能的提升幅度。

为量化分类性能,本文采用准确率(Accuracy)和 F1 得分(F1-Score)作为主要评估指标,两者的取值均为 $[0,1]$,值越高,代表性能越好。

4.2.4 生成多样性

该维度用于评估生成流量在真实数据特征空间中的覆盖程度,以反映生成方法生成多样化样本的能力。

本文采用覆盖率(Coverage)指标进行度量。其计算方法为:首先,为每个真实数据样本构建一个邻域(本文采用 k 近邻算法,$k=5$);然后,统计包含至少一个生成样本的真实数据邻域所占的比例。

高覆盖率表明生成数据能够有效捕获真实数据的整体分布,低覆盖率则表明真实数据分布中的某些区域缺乏相应的生成样本。

4.3 实验结果

4.3.1 领域专家知识检查

本项指标旨在评估生成流量在协议层面的语法正确性。其通过一个预定义的规则检查框架,量化符合基础网络协议规范的生成样本所占的比例。实验结果如表 2 所列。

表 2　不同方法在领域专家知识检查、分布相似性以及生成多样性方面的评估结果

Table 2　Evaluation results of different methods in terms of domain expert knowledge verification, distribution similarity, semantic consistency, and generation diversity

Data Granularity	Model	Compliance Rate	JSD	EMD	Coverage
Coarse-grained	DoppelGANger	0.8392	0.2842	0.0539	0.0456
	NetShare	0.9613	0.2281	0.1622	0.1486
	NetGPT	0.9869	**0.1857**	0.1885	0.4167
Fine-grained	NetDiffusion	**1.0000**[1)]	0.5905	0.9856	0.0512
	Ours	**0.9995**	0.2465	**0.0059**	**0.5015**

NetDiffusion 达到了 1.0000 的理论最优通过率，这主要归因于其在生成流程中包含了确定性的后处理校正环节。然而，该模型在不使用后处理的情况下，原始生成结果的检查通过率下降超过 0.6，这说明其本身并未充分学到语义，仅靠人工规则对生成内容进行修正，这也导致其在其他指标上表现较差。

相比之下，本文方法未依靠任何后处理机制，同样实现了 0.9995 的极高通过率，说明其本身内化了协议结构知识，实现合规生成。另一基于自回归范式（AR）的 NetGPT 表现亦十分稳健，取得了 0.9869 的通过率。相较之下，专注于粗粒度特征生成的生成对抗网络方法在该项指标上表现欠佳，其中 NetShare 的通过率为0.9613，而 DoppelGANger 的得分最低，仅为 0.8392。

4.3.2　分布相似性

表 2 中的分布相似性评估结果表明，本文方法在两项核心指标上均展现出优异性能。具体而言，本文方法在 EMD 指标上达到 0.0059，优于其他所有模型，体现出在连续型数据分布还原方面的绝对优势。在 JSD 指标方面，本文方法取得 0.2465，略差于 NetGPT，NetShare。但从整体评估角度来看，EMD 指标对连续变量（如包长、流持续时间）的分布敏感，更能反映生成流量在关键行为特征上的统计真实性。因此，本文方法在 EMD 上的优势表明，其生成的流量在核心统计特性上更贴近真实分布，虽在 JSD 指标上未取得最优，但综合性能仍具有明显优势。

4.3.3　下游任务性能

本文通过一项恶意流量多分类任务，对生成数据的下游任务实用性进行了评估。由于 NetGPT 仅生成粗粒度包级别数据且 NetDiffusion 无法生成时间戳，因此该评估仅针对 DoppelGANger，NetShare 及本文方法。

在生成数据训练、真实数据测试（TSTR）的评估中，首先建立了一个性能基准，即在真实数据上训练分类器（随机森林 RF、梯度提升 GB、XGBoost），该基准在所有分类器上均取得了稳健的性能。如图 3 所示，由本文方法生成的数据表现出最优的 TSTR 性能，其训练出的分类器性能最大程度上接近了基准水平。相较之下，NetShare 与 DoppelGANger 的生成数据在该直接替换场景下，所训练模型的性能偏低。

其次，在数据增强实验中，模拟真实场景中恶意流量样本稀缺、类别不平衡的问题。数据增强前后类别的分布情况如表 3 所列。实验结果表明，使用本文方法生成的样本对训练集进行增强是有效的，在所有分类器上，其 F1 得分均能持平或略微超过基准性能。与此形成对比的是，尽管 NetShare 和 DoppelGANger 的生成数据同样被用于平衡数据集，但如表 4 所列，它们的加入反而可能导致 F1 得分相较于基准出现轻微的性能下降。

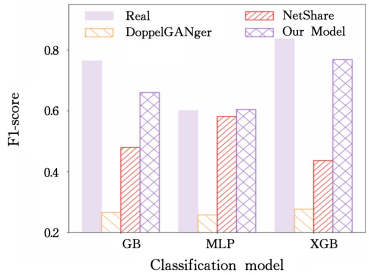

图 3　TSTR 任务下不同机器学习分类器的 F1 分数对比

Fig. 3　Comparison of F1-scores for the TSTR task across different ML classifiers

表 3　数据增强前后类别的分布情况

Table 3　Distribution of categories before and after data augmentation

流量类型	增强前		增强后	
	数量	占比/%	数量	占比/%
DDoS	10000	29.76	10000	25.00
Bruteforce	3600	10.72	10000	25.00
Spoofing	10000	29.76	10000	25.00
Reconnaissance	10000	29.76	10000	25.00

表 4　采用不同生成方法平衡数据集前后的 F1 分数对比

Table 4　Comparison of F1-scores before and after balancing the dataset with different generation methods

Generation Model	Classification Model	F1-Score Pre/Post Balancing	Δ
DoppelGANger	GB	0.764→0.759	−0.5%
	MLP	0.599→0.598	−0.1%
	XGB	0.835→0.826	−0.9%
NetShare	GB	0.764→0.753	−1.1%
	MLP	0.599→0.599	0.0%
	XGB	0.835→0.826	−0.9%
Ours	GB	0.764→0.767	**+0.3%**
	MLP	0.599→0.606	**+0.7%**
	XGB	0.835→0.835	0.0%

4.3.4　生成多样性

表 2 中的覆盖率（Coverage）指标用于评估生成样本的多样性。本文方法在此项评估中具有优势，取得了 0.5015 的最高覆盖率得分，NetGPT 得分为 0.4167。与此形成对比的是，粗粒度的生成对抗网络方法与 NetDiffusion 在多样性方面表现较差，其覆盖率得分普遍较低：DoppelGANger，NetShare 及

[1)] 该检查通过率为 NetDiffusion 经过后处理校正后的结果。

NetDiffusion 的得分分别仅为 0.0456,0.1486 和 0.0512。

综合上述 4 个维度的实验结果,本文方法在生成高保真度、细粒度的恶意网络流量方面,展现出了相较于基准模型的综合性与系统性优势。

首先,在基础质量层面,本文方法兼顾了协议语法的合规性与统计分布的相似性。在领域专家知识检查中,本文方法取得了高达 0.9995 的通过率,仅次于依赖确定性后处理的 NetDiffusion,充分证明了本文方法通过自回归学习本身,能高度掌握网络协议的内在语法。在分布相似性上,本文方法在模拟连续型特征(EMD 指标最优)方面表现突出,并在其他统计指标上亦表现出色,表明其能够忠实地复现真实流量的统计特性。

其次,在下游任务性能方面,本文方法具有明显优势。实验结果表明,基于本文方法生成数据训练的分类模型在"生成数据训练、真实数据测试"(TSTR)场景下,其性能可最大程度地接近真实数据训练的性能基准。此外,本文方法也是唯一能够在数据增强应用中为下游模型性能带来正面提升的方法。与之形成对比的是,其余基准方法生成的流量在此项评估中表现不佳,甚至对模型性能产生负面影响。这一结果有力证明了,本文方法不仅能生成"看起来像"的数据,更能生成蕴含真实攻击核心语义的数据。

4.4 恶意流量验证

为进一步验证本文方法所生成流量的真实性与有效性,本节设计并实施了一项恶意流量功能验证实验。该实验旨在确认生成的恶意流量在实际网络环境中,是否能够成功复现其所对应的攻击效果。

在本实验中,选取了慢速 DDoS 攻击与 HTTP 泛洪攻击作为代表性场景。此选择主要基于实验设计的严谨性与效能评估的客观性两大原则。首先,这两类攻击具备明确的攻击模式和成熟的效能评估指标。其攻击效果可以直接通过对目标服务的关键性能指标进行监控来客观度量,例如服务器响应延迟、请求成功率、CPU 负载等,从而确保了验证过程的客观性和结果的明确性。相较而言,其他攻击类型(如侦察或欺骗攻击)的验证,通常需要复现复杂的网络环境拓扑,且其成功与否往往依赖于对日志、报警等多源信息的间接证据链进行综合判定。

实验在一个隔离的内网环境中进行,该环境由一台配置为攻击机和一台部署了 Apache2 Web 服务的靶机(目标服务器)组成,两者均运行 Ubuntu 18.04 操作系统。

攻击重放采用了一种三阶段方法,以确保高度保真地还原流量特性。首先,利用从原始真实流量中提取的 HTTP 头部字段值,对本文方法生成的流量进行补充,构造出协议完备的 PCAP 文件。其次,在攻击机上通过 Python socket 编程接口,并行地创建并维持与目标服务器的多个 TCP 连接。最后,依据 PCAP 文件中精确的包间时间间隔,依次发送数据包,从而精确复现生成流量的时序与内容特征。

1)慢速 DDoS 攻击验证

首先对生成的慢速 DDoS 攻击流量进行功能性验证。该攻击(由 Slowloris 工具产生)的核心机制在于:攻击方通过发送不完整的 HTTP 请求,并长期占有 TCP 连接,以此耗尽目标服务器的并发连接池资源,最终导致合法用户无法建立新连接。

本文实验中,重放了 1000 条由本文方法生成的慢速 DDoS 攻击流量。为量化评估攻击效果,采用 httping 工具对目标服务器的 HTTP 服务连通性进行持续监测。实验开始约两分钟后,目标服务器的 Web 页面加载出现延迟。通过 httping 工具测试发现,页面的平均加载时间从基准状态的 0.298 ms 急剧攀升至 17.4 s,同时 HTTP 请求失败率达到了 40%。约 6 min 后,目标 Web 服务完全中断,无法访问。该结果明确证实,本文方法生成的慢速 DDoS 流量成功地对目标服务器造成了拒绝服务效果,验证了其攻击语义的有效性。

2)HTTP 泛洪攻击验证

随后,对 HTTP 泛洪攻击流量进行了验证。此类攻击通过发送海量的、看似合法的 HTTP 请求,耗尽服务器的处理与网络资源。实验中,重放了 8000 条由本文方法生成的 HTTP 泛洪攻击流量。

重放约 2 min 后,观测到目标服务器的 Web 页面加载时间从 0.298 ms 增加到 1.24 s,服务性能出现明显下降。同时,通过 nload 工具对目标服务器的网络流量进行监控,发现在攻击期间,其入网流量相较于正常状态出现了激增。这一现象印证了生成流量成功模拟了 HTTP 泛洪攻击大量消耗网络带宽的资源耗尽型特征。

结束语 为解决网络安全领域中恶意流量获取困难、标注成本高等关键问题,本文提出并实现了一种基于预训练-微调范式的细粒度网络流量生成方法。通过设计一种保留结构信息的静态分词方法,并结合两阶段学习策略,本文方法在领域专家知识检查、分布相似性、下游任务性能及生成多样性等多个维度上,均展现出相较于主流基准模型的综合优势。实验结果表明,本文方法不仅能够生成高质量的、符合协议规范的流量数据,其生成数据更可应用于下游恶意流量识别任务,并提升下游模型性能,验证了该技术路线的可行性。

尽管本文方法取得了一定的研究成果,但仍存在可供探索与改进的方向。首先,在方法能力层面,当前工作主要集中于网络流量头部的生成,未来的研究可尝试对具有特定语义的载荷进行建模与生成,以进一步提高合成数据在高级安全分析场景中的适用性。其次,在评估体系层面,尽管本文采用了多维度的评估框架,生成流量的语义真实性度量本身仍是一个开放的研究挑战,未来工作可以探索引入语义评估指标。

参 考 文 献

[1] FU C, 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.

[2] HAN D, WANG Z, CHEN W, et al. Anomaly Detection in the Open World: Normality Shift Detection, Explanation, and Adaptation[C]// 30th Annual Network and Distributed System Security Symposium (NDSS). 2023: 1-18.

[3] LIAN X, CAO C, LIU Y, et al. Facing Anomalies Head-On: Net-

work Traffic Anomaly Detection via Uncertainty-Inspired Inter-Sample Differences[C]// Proceedings of the ACM on Web Conference 2025. New York:ACM,2025:3908-3917.

[4] ZHAO Z,LI Z,SONG Z,et al. Trident:A universal framework for fine-grained and class-incremental unknown traffic detection [C]// Proceedings of the ACM Web Conference 2024. New York:ACM,2024:1608-1619.

[5] ZHOU G,GUO X,LIU Z,et al. TrafficFormer:An Efficient Pre-trained Model for Traffic Data[C]// 2025 IEEE Symposium on Security and Privacy(SP). San Francisco:IEEE Computer Society,2024:102-118.

[6] ADELEKE O A,BASTIN N,GURKAN D. Network traffic generation:A survey and methodology[J]. ACM Computing Surveys,2022,2:1-23.

[7] DU Z,QIAN Y,LIU X,et al. GLM:General Language Model Pretraining with Autoregressive Blank Infilling[C]// Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics. ACL,2022:320-335.

[8] DONG Y,DING J,JIANG X,et al. Codescore:Evaluating code generation by learning code execution[J]. ACM Transactions on Software Engineering and Methodology,2025,3:1-22.

[9] LI J,LI G,LI Y,et al. Structured chain-of-thought prompting for code generation[J]. ACM Transactions on Software Engineering and Methodology,2025,2:1-23.

[10] HENDERSON T R,LACAGE M,RILEY G F,et al. Network simulations with the ns-3 simulator[J]. SIGCOMMDemonstration,2008,4:527-527.

[11] BÜHLER T,SCHMID R,LUTZ S,et al. Generating representative,live network traffic out of millions of code repositories [C]// Proceedings of the 21st ACM Workshop on Hot Topics in Networks. New York:ACM,2022:1-7.

[12] ROLLAND C,RIDOUX J,BAYNAT B. LiTGen,a lightweight traffic generator:application to P2P and mail wireless traffic [C]// Passive and Active Network Measurement:8th Internatinoal Conference. Berlin:Springer,2007:52-62.

[13] Naval Research Laboratory. Multi-Generator(MGEN)[EB/OL]. (2021-08-25)[2025-09-19]. https://www.nrl.navy.mil/itd/ncs/products/mgen.

[14] CHU A,JIANG X,LIU S,et al. Feasibility of state space models for network traffic generation[C]// Proceedings of the 2024 SIGCOMM Workshop on Networks for AI Computing. New York:ACM,2024:9-17.

[15] RING M,SCHLÖR D,LANDES D,et al. Flow-based network traffic generation using generative adversarial networks[J]. Computers & Security,2019,82:156-172.

[16] LIN Z,JAIN A,WANG C,et al. Using gans for sharing networked time series data:Challenges,initial promise,and open questions[C]// Proceedings of the ACM Internet Measurement Conference. New York:ACM,2020:464-483.

[17] YIN Y,LIN Z,JIN M,et al. Practical gan-based synthetic ip header trace generation using netshare[C]// Proceedings of the ACM SIGCOMM 2022 Conference. New York:ACM,2022:458-472.

[18] VASWANI A,SHAZEER N,PARMAR N,et al. Attention is all you need[J]. Advances inNeural Information Processing Systems,2017,30:6000-6010.

[19] RADFORD A,WU J,CHILD R,et al. Language models are unsupervised multitask learners[J]. OpenAI Blog,2019,1(8):9.

[20] MENG X,LIN C,WANG Y,et al. Netgpt:Generative pretrained transformer for network traffic[J]. arXiv:2304.09513,2023.

[21] WANG Q,QIAN C,LI X,et al. Lens:A foundation model for network traffic in cybersecurity[J]. arXiv:2402.03646,2024.

[22] HO J,JAIN A,ABBEEL P. Denoising diffusion probabilistic models[J]. Advances inNeural Information Processing Systems,2020,33:6840-6851.

[23] SIVAROOPAN N,BANDARA D,MADARASINGHA C,et al. Netdiffus:Network traffic generation by diffusion models through time-series imaging[J]. Computer Networks,2024,251:1-13.

[24] ZHANG S,LI T,JIN D,et al. NetDiff:A service-guided hierarchical diffusion model for network flow trace generation[C]// Proceedings of the ACM on Networking. 2024:1-21.

[25] JIANG X,LIU S,GEMBER-JACOBSON A,et al. Netdiffusion:Network data augmentation through protocol-constrained traffic generation[J]. Proceedings of the ACM on Measurement and Analysis of Computing Systems,2024,8(1):1-32.

[26] GOODFELLOW I J,POUGET-ABADIE J,MIRZA M,et al. Generative adversarial nets[J]. Advances in Neural Information Processing Systems,2014,27:2672-2680.

[27] NETOEC 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):5941-5967.

ZHANG Can,born in 2004,Ph.D candidate. Her main research interest is cybersecurity.

HAN Dongqi,born in 1997,Ph.D,associate professor/reseacher,master's supervisor,is a member of CCF(No. 93935M). His main research interests include network and artificial intelligence security.

(责任编辑:喻藜)