工程科学学报
Chinese Journal of Engineering

基于图神经网络增强通信特征的僵尸网络异常通信检测

王云浩 胡堰 皇甫伟 霍佳皓

Detection of Botnet anomalous communication based on GNN-enhanced communication features

WANG Yunhao, HU Yan, HUANGFU Wei, HUO Jiahao

引用本文:
王云浩, 胡堰, 皇甫伟, 霍佳皓. 基于图神经网络增强通信特征的僵尸网络异常通信检测[J]. 北科大：工程科学学报, 2025, 47(10): 2027–2037. doi: 10.13374/j.issn2095–9389.2024.12.20.001
WANG Yunhao, HU Yan, HUANGFU Wei, HUO Jiahao. Detection of Botnet anomalous communication based on GNN–enhanced communication features[J]. *Chinese Journal of Engineering*, 2025, 47(10): 2027–2037. doi: 10.13374/j.issn2095–9389.2024.12.20.001

在线阅读 View online: https://doi.org/10.13374/j.issn2095–9389.2024.12.20.001

工程科学学报, 第 47 卷, 第 10 期: 2027−2037, 2025 年 10 月
Chinese Journal of Engineering, Vol. 47, No. 10: 2027−2037, October 2025
https://doi.org/10.13374/j.issn2095-9389.2024.12.20.001; http://cje.ustb.edu.cn

基于图神经网络增强通信特征的僵尸网络异常通信检测

王云浩[1,2)], 胡　堰[1,2)✉], 皇甫伟[1,2)], 霍佳皓[1,2)]

1) 北京科技大学计算机与通信工程学院, 北京 100083　2) 北京科技大学顺德创新学院, 佛山 528399
✉通信作者, E-mail: huyan@ustb.edu.cn

摘　要　工业互联网中的传统工业设备存在大量安全漏洞, 在联网过程中易受僵尸网络攻击, 其通过恶意控制大量联网设备, 实现对目标网络的大规模协同攻击. 传统基于规则或阈值的检测方法过度依赖静态签名或人工阈值设定, 很难适应动态变化的网络环境; 传统机器学习技术对复杂网络高维通信特征的处理能力有限, 导致检测能力受限; 基于深度学习的检测技术通常将网络流量视为时间序列或空间数据进行处理, 无法对设备拓扑依赖关系进行建模, 因而难以识别僵尸网络协同攻击. 为了解决上述局限性, 本文采用图结构准确建模复杂通信网络拓扑结构, 并提出一种基于图神经网络增强通信特征的僵尸网络异常通信检测技术. 首先从网络流量数据中挖掘细粒度的节点特征与通信特征; 然后通过图神经网络的信息传播与聚合机制, 获得准确的节点聚合特征表示; 再用节点聚合特征增强通信特征, 实现准确的异常通信检测; 最后在大型公开数据集 CTU-13 上进行了综合实验, 验证所提出方法的有效性. 实验结果表明所提出的方案与现有的卷积神经网络、长短时记忆网络及其融合模型等异常检测算法, 以及最新提出的 Bot-DM 僵尸网络检测方法相比, 能更准确地检测僵尸网络异常通信.

关键词　工业互联网安全; 僵尸网络; 异常通信检测; 图神经网络; 深度学习
分类号　TN915.08

Detection of Botnet anomalous communication based on GNN-enhanced communication features

WANG Yunhao[1,2)], *HU Yan*[1,2)✉], *HUANGFU Wei*[1,2)], *HUO Jiahao*[1,2)]

1) School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
2) Shunde Innovation School, University of Science and Technology Beijing, Foshan 528399, China
✉Corresponding author, E-mail: huyan@ustb.edu.cn

ABSTRACT　The Industrial Internet is an important part of the national critical information infrastructure. Enabling comprehensive interconnectivity among humans, machines, and Internet of Things devices allows the formation of a new architecture of industrial production, manufacturing, and service. However, a great number of security vulnerabilities exist in industrial devices, especially legacy industrial devices. They can be maliciously exploited during device interconnection, causing severe security incidents or economic losses. Among the major security threats facing the Industrial Internet today, botnet attacks are particularly concerning. By exploiting zero-day vulnerabilities (e.g., buffer overflows in the programmable logic controller firmware) and propagating and deploying polymorphic malware, attackers can covertly hijack a large number of networked devices and recruit compromised devices into botnets to launch coordinated large-scale attacks on target networks. However, traditional botnet detection methods (e.g., rule-, threshold-, and machine learning-based methods) have significant limitations. Rule- and threshold-based botnet detection techniques, which depend heavily on static signatures (e.g., known malicious Internet Protocol lists) or predefined detection thresholds, face challenges in adapting

收稿日期: 2024−12−20
基金项目: 国家自然科学基金区域基金重点资助项目(U22A2005); 北京科技大学青年教师学科交叉研究(中央高校基本科研业务费专项资金)资助项目(FRF-IDRY-24-015)

to the dynamic nature of complex network environments, ultimately leading to constrained detection capabilities. Meanwhile, it is not easy for traditional machine learning-based detection techniques to process complex and high-dimensional network communication features effectively, resulting in poor detection performance. Deep learning-based detection techniques, which generally treat network traffic as isolated time-series or spatial data, fail to model the topological dependencies between devices in complex communication networks; this is a key limitation in identifying coordinated botnet behaviors (e.g., synchronized command-and-control communications). To address these challenges, we leverage the pervasive device-to-device connectivity in the Industrial Internet by modeling the communication network as a graph structure, where nodes represent devices and edges represent communication relationships between devices to achieve accurate topology representation. Based on the graph model, we propose a novel approach for detecting botnet anomalous communication based on graph neural network (GNN)-enhanced communication features. First, our method extracts fine-grained node and communication features from network traffic data and employs a GNN to propagate and aggregate node information across the entire network. By capturing topological dependencies, the method can generate more accurate aggregated node feature representations. In this step, the multihead attention mechanism is integrated with the GNN to perform weighted aggregation of node features in diverse ways, enhancing the flexibility of node feature representation. Afterward, the aggregated node features are used to enhance communication features. Finally, a multilayer perceptron model is used to classify the enhanced communication features into the normal or abnormal categories, thus achieving automatic detection of botnet anomalous communication. To validate the effectiveness of the proposed approach, we conducted a series of experiments on a public large-scale dataset, CTU-13, which includes 13 distinct botnet attack scenarios. We compared the proposed approach against a group of baseline methods, including a convolutional neural network (CNN), long short-term memory (LSTM), CNN-LSTM, and the recently proposed Bot-DM method, across a comprehensive set of metrics such as accuracy, recall, precision, and F1-score. The experimental results demonstrate that our approach outperforms existing botnet detection methods in detection performance.

KEY WORDS security of Industrial Internet; botnet; anomalous communication detection; graph neural networks; deep learning

工业互联网的蓬勃发展使得冶金、采矿、能源等智能制造领域的信息化与智能化水平不断提升[1]. 持续推进我国由制造大国向制造强国转变. 智能制造设备的互联互通、工业控制网络与生产管理网络的高度集成、工厂网络与互联网的广泛融合, 使新一代智能制造体系所依赖的信息网络基础设施具备灵活的网络架构, 并促进了实时工业控制、智慧化管控、产业链协同等创新制造模式的发展. 然而, 由于智能制造设备本身存在较多的安全漏洞, 大批设备的联网使工业互联网面临严峻的安全形势. 一旦设备遭受恶意攻击, 很可能导致严重的安全事故或经济损失[2]. 僵尸网络是目前工业互联网面临的主要安全威胁之一[3], 其通过控制大量受感染设备对目标工业网络发起协同攻击, 从而破坏生产流程或窃取机密数据. 因此, 准确检测僵尸网络异常通信是后续采取有效防护措施的前提, 也是保障工业互联网安全的重要手段.

僵尸网络产生的异常通信通常混杂在大量正常网络通信中, 且呈现突发性、动态性、伪装性等特征[4−5], 导致传统基于阈值或规则匹配[6−7] 的方法很难对其进行有效检测. 因此, 研究者们提出了基于机器学习的检测技术, 如决策树和支持向量机[8]、多层机器学习分类器[9] 等, 但此类方法对高维复杂通信特征的自动选择和处理能力有限, 导致检测效果不佳. 另一些学者将深度学习技术应用于僵尸网络检测, 如卷积神经网络(Convolutional neural network, CNN)[10]、长短期记忆网络(Long short-term memory, LSTM)[11] 以及融合模型 CNN–LSTM[12] 等, 以实现高维特征的自动选择和非线性处理[13−14], 提升检测效果. 但大部分深度学习技术仅关注设备通信产生的网络流量特征[15], 忽略了设备之间的网络拓扑关系, 使僵尸网络通信特征的表示仍不够准确, 导致模型检测能力依然受限.

由于设备互联形成典型的图结构, 设备及其通信关系可被建模为图的节点和边, 因此图模型非常适用于描述设备拓扑关系[16−17]. Zhou 等[18] 采用图结构建模设备拓扑关系, 并利用图神经网络(Graph neural network, GNN) 和图卷积操作, 实现网络中节点信息的传播与聚合, 从而识别僵尸网络节点. 该方法主要关注网络节点特征, 而未考虑通信特征及其与节点特征的关联关系, 导致检测精度受限. Meng 等[19] 同样利用图神经网络检测僵尸网络, 但仅关注连接数和平均通信时长等基本通信特征, 未深入挖掘更多细粒度的网络通信特征, 从而影响检测精度. Zhao 等[20] 指出入度和出度等基本图特征仅能粗粒度地表示设备间连接关系,

无法细致刻画设备间的复杂通信模式. Wu 等[21]将网络流量转换成图像表示, 并结合数据包负载分析, 实现僵尸网络异常通信检测. 然而, 解析大量数据包的负载信息, 必然会消耗大量的计算资源和时间, 显著增加检测成本. 针对上述问题, 本文提出了一种基于图神经网络增强通信特征的僵尸网络异常通信检测模型 (Botnet anomalous communication detection based on GNN-enhanced communication features, Bot-GECF). 首先用图模型建模设备拓扑结构, 并抽取细粒度的节点特征和通信特征, 然后利用图神经网络实现网络内节点特征的传播和聚合, 得到准确的节点聚合特征表示, 再用其增强通信特征, 形成准确的通信特征表示, 最后利用多层感知机 (Multilayer perceptron, MLP) 模型对增强的通信特征进行分类, 实现僵尸网络异常通信检测. 本文的主要创新点如下:

(1) 基于原始网络通信数据提取多维细粒度节点特征与通信特征, 较全面地刻画设备历史行为模式与单次通信模式;

(2) 运用图模型准确建模设备拓扑结构, 基于图神经网络中节点信息的传播与聚合机制, 有效捕捉复杂通信网络中设备之间的局部和全局依赖关系;

(3) 利用通信双方节点的历史行为特征增强单次通信特征, 有效提升检测模型对单次通信异常的识别能力.

1 问题建模

僵尸网络由一系列受恶意操控的设备组成, 并混杂在大量正常设备中形成复杂网络. 图 1 展示了僵尸网络通信模型, 红色与黑色节点分别代表僵尸设备和正常设备, 僵尸设备能与其他僵尸设备或正常设备进行恶意通信. 本文将该网络建模为通信图模型$G(\mathcal{V},\boldsymbol{A})$, 其中$\mathcal{V}$为节点集合, 共包含$N$个独立节点$\{v_1,\cdots,v_N\}$, 每个节点对应一台联网设备. $\boldsymbol{A}\in\mathbf{R}^{N\times N}$为邻接矩阵, 其所有对角线元素均为 1. 当节点$v_m$和$v_n(1\leqslant m,n\leqslant N)$之间存在至少一次通信, 则$\boldsymbol{A}$的元素$a_{mn}$为 1, 否则为 0. 同时, 用$\boldsymbol{D}=\mathrm{diag}(d_1,d_2,\cdots,d_N)$表示节点的度矩阵, 其中$d_m=\sum_{n=1}^{N}a_{mn}$. 用$\boldsymbol{A}$和$\boldsymbol{D}$便可表示通信网络的拓扑结构. 此外, 网络中任意源节点v_m到任意目的节点v_n的第$l(l\geqslant 1)$次通信表示为e_{nm}^{l}, 并对应一个标签$y_{mn}^{l}\in\{0,1\}$. $y_{mn}^{l}=0$表示e_{nm}^{l}为正常通信, 否则为僵尸网络异常通信.

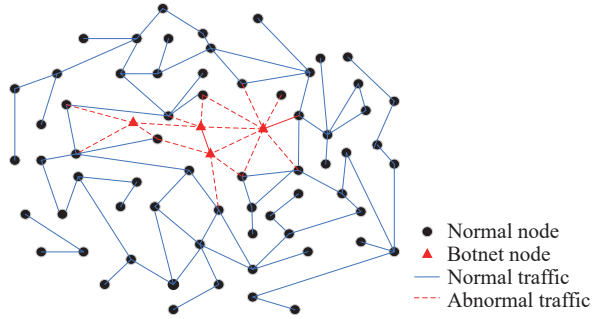

图 1 正常网络嵌入僵尸网络的示意图

Fig.1 Example of a normal network embedded with a P2P botnet

为了准确识别僵尸网络异常通信, 本文构建了一个分类器C, 实现单次通信到类别的映射. 构建分类器的过程可建模为以下优化问题:

$$\hat{\theta}=\underset{\theta}{\mathrm{argmin}}\,\mathcal{L}(\theta,G) \tag{1}$$

其中, θ为模型C的所有可训练参数, G是待识别的网络通信图. 优化的目标是通过最小化损失函数$\mathcal{L}$来获得模型C的最佳参数$\hat{\theta}$, 从而构建最优分类器.

2 基于图神经网络增强通信特征的僵尸网络异常通信检测模型

2.1 模型概览

基于上述问题模型, 本文提出一种基于图神经网络增强通信特征的僵尸网络异常通信检测模型 Bot-GECF, 其总体架构如图 2 所示. 首先, 基于原始通信数据, 以网络设备为节点、设备间通信关系为边, 构建网络通信图, 并为每个节点和每次通信分别抽取细粒度的初始特征向量. 然后, 采用图神经网络和多头注意力机制生成节点聚合特征向量, 并用其增强通信特征. 最后, 利用 MLP 模型对增强的通信特征向量进行自动分类, 实现僵尸网络异常通信的准确检测.

2.2 初始特征构建

本文从原始通信数据中抽取与节点相关的细粒度特征, 构建节点初始特征向量. 由于原始通信数据集$\mathcal{T}_{\mathrm{o}}$中异常通信占比很小, 呈现明显的不平衡特性. 因此, 本文采用 SMOTE–Tomek 方法[22]对其进行过采样和欠采样的组合处理, 得到均衡化的数据集$\mathcal{T}$. 节点主要包含连接特征和流量特征. 其中, 连接特征包括当前节点分别作为源节点和目的节点时, 其自身通信端口的多样性X_{SP}、对方节点的多样性X_{AV}、对方端口的多样性X_{AP}、一次通信的平均时间T_{M}及方差D_{T}、相邻两次通信的平均时间间隔T_{I}及方差D_{I}、使用最频繁的协议占比P_{R}、协议的变化频率F_{R}和连接状态的变化频率

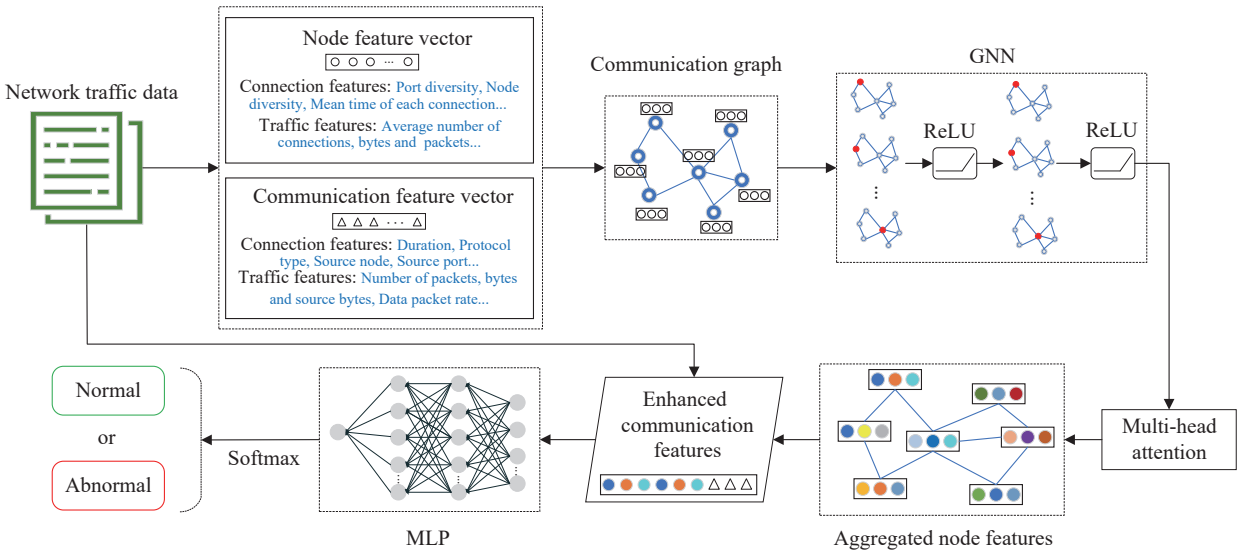

图 2　Bot-GECF 模型总体架构

Fig.2　Architecture of Bot-GECF

F_S. 本文通过熵值来衡量多样性, 低熵值通常表示存在集中攻击的可能性. 流量特征包括当前节点在单位时间窗口内的平均通信连接数N_C、传输字节数N_B和数据包数N_P. 然后, 对不同类型的特征进行规范化处理, 便于模型训练. 对于类别型特征, 采用标签编码技术[23]将其转换为数值特征. 对于数值型特征, 采用 Z-分数[24]进行标准化处理. 最后, 将规范后的特征进行拼接, 为每个节点v_n生成对应的初始特征向量$\boldsymbol{v}_n \in \mathbf{R}^{1\times d_v}$, 如公式 (2) 所示.

$$\begin{aligned}\boldsymbol{v}_n =&\text{Norm}((X_{SP}, X_{AV}, X_{AP}, T_M, D_T, T_I, D_I, P_R,\\ &F_R, F_S, N_C, N_B, N_P))\end{aligned} \tag{2}$$

其中, d_v为节点初始特征维度, Norm(·)为向量规范化函数. 此外, 还需要构建通信特征向量, 描述任意一次通信的属性. 通信特征同样包括连接特征和流量特征. 其中, 连接特征包括此次通信的持续时间T、协议类型Z、源节点V_S和端口号Q_S、目的节点V_D和端口号Q_D、连接状态S_C和服务类型U_S. 流量特征包括此次通信中传输的数据包总数N_P、字节总数N_B、源字节数N_{SB}、数据包速率R_P、字节速率R_B、数据包大小的方差D_P和偏度J_P、字节和数据包的突发指示O_B和O_P. 其中, 数据包方差表示此次通信过程中数据包大小的波动性, 数据包偏度衡量数据包大小分布的偏向性, 明显的波动和偏度都表示可能存在异常. 突发指示为二值变量, 用于指示此次通信产生的字节数和数据包数是否超过其对应节点单次通信平均字节数或平均数据包数的特定倍数. 最后, 对这些特征进行规范化处理和拼接, 为每次通信e^l_{nm}生成对应的初始特征向量$\boldsymbol{c}^l_{mn} \in \mathbf{R}^{1\times d_c}$, 如公式 (3) 所示, 其中$d_c$为通信初始特征维度.

$$\begin{aligned}\boldsymbol{c}^l_{mn} =&\text{Norm}((T, Z, V_S, Q_S, V_D, Q_D, S_C, U_S, N_P,\\ &N_B, N_{SB}, R_P, R_B, D_P, J_P, O_B, O_P))\end{aligned} \tag{3}$$

2.3　基于 GNN 的节点聚合特征学习

GNN 是一种专门处理图结构数据的深度学习模型, 通过图卷积操作实现相邻节点间的信息传播与聚合, 从而形成融合重要上下文信息的节点特征. 本文基于 GNN 模型学习通信网络中节点的聚合特征表示. 具体地, 用$\boldsymbol{V}^k \in \mathbf{R}^{N\times d_k}$ $(1 \leqslant k \leqslant K)$表示第$k$层所有节点构成的特征矩阵, 其任意一行$\boldsymbol{V}^k_n$ $(1 \leqslant n \leqslant N)$表示节点$v_n$的特征向量, d_k为第k层的特征维度, K为图卷积层总数. 所有节点的初始特征向量$\boldsymbol{v}_n$ $(1 \leqslant n \leqslant N)$构成了初始特征矩阵$\boldsymbol{V}^0$. 因此, 基于 GNN 的节点特征更新过程表示为:

$$\boldsymbol{V}^k = \sigma\left(\boldsymbol{V}^{k-1}\boldsymbol{W}^k_s + \tilde{\boldsymbol{A}}\boldsymbol{V}^{k-1}\boldsymbol{W}^k_c + \boldsymbol{B}^k\right) \tag{4}$$

其中, $\boldsymbol{W}^k_s$、$\boldsymbol{W}^k_c$和$\boldsymbol{B}^k$分别为第k层可训练的残差连接矩阵、卷积权重矩阵和偏置向量, 每一层保留节点部分原始特征, 避免特征同质化, $\sigma(\cdot)$为激活函数 (如 ReLU 函数等), $\tilde{\boldsymbol{A}}$是归一化的邻接矩阵:

$$\tilde{\boldsymbol{A}} = \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2} \tag{5}$$

2.4　构建增强的通信特征进行异常识别

此外, 本文在 GNN 训练过程中引入多头注意力机制[25], 用不同的方式对输入特征进行加权聚合, 以实现更加灵活的特征表示. 首先, 设置h个独立的注意力头, 分别对第K层图卷积层输出的节点特征进行注意力操作, 得到输出$\boldsymbol{A}_i = \text{Att}_i(\boldsymbol{V}^K)$, 其中$1 \leqslant i \leqslant h$. 然后, 将$h$个独立注意力头的输出进行拼

接和线性变换得到最终的节点特征矩阵:

$$\boldsymbol{V}=(\boldsymbol{A}_1,\cdots,\boldsymbol{A}_h)\boldsymbol{W}_{\mathrm{O}} \tag{6}$$

其中, $\boldsymbol{W}_{\mathrm{O}}$是线性变换矩阵, 用于将拼接后的特征空间映射至原特征空间.

然后, 基于节点聚合特征向量构造增强的通信特征向量. 对于任意一个通信初始特征向量$\boldsymbol{c}_{mn}^{l}$, 首先对其进行线性变换和 LayerNorm 归一化[26], 来规范特征分布, 提升模型稳定性, 并通过激活函数增强特征的非线性表达能力:

$$\boldsymbol{c}_{mn}^{l}{}'=\mathrm{PReLU}\left(\mathrm{LayerNorm}\left(\boldsymbol{c}_{mn}^{l}\boldsymbol{W}_{\mathrm{e}}+\boldsymbol{b}_{\mathrm{e}}\right)\right) \tag{7}$$

其中, $\boldsymbol{W}_{\mathrm{e}}$和$\boldsymbol{b}_{\mathrm{e}}$分别为可学习的权重矩阵和偏置向量, $\mathrm{PReLU}(x)=\max(x,0)+a\cdot\min(x,0)$为非线性激活函数[27], a通常取 0.25. 然后, 将$\boldsymbol{c}_{mn}^{l}{}'$与其对应的源节点和目标节点的聚合特征向量$\boldsymbol{V}_m$和$\boldsymbol{V}_n$(分别对应矩阵$\boldsymbol{V}$的第$m$行和第$n$行) 进行融合, 得到增强的通信特征向量$\bar{\boldsymbol{c}}_{mn}^{l}$:

$$\bar{\boldsymbol{c}}_{mn}^{l}=\left(\boldsymbol{V}_m,\boldsymbol{V}_n,\boldsymbol{c}_{mn}^{l}{}'\right) \tag{8}$$

最后, 本文使用 MLP 模型[28] 结合softmax函数对$\bar{\boldsymbol{c}}_{mn}^{l}$进行异常识别:

$$p_{mn}^{l}=\mathrm{softmax}(\mathrm{MLP}(\bar{\boldsymbol{c}}_{mn}^{l})) \tag{9}$$

其中, p_{mn}^{l}表示$\bar{\boldsymbol{c}}_{mn}^{l}$被预测属于僵尸网络通信的概率. 若大于特定阈值, 则判定为异常通信.

2.5 模型训练

由于原始通信数据集呈现明显的不平衡性, 因此在模型训练过程中采用改进的交叉熵损失函数(Focal loss) 来提高模型对少数类样本的学习效果:

$$\begin{aligned}\mathcal{L}\left(p_{mn}^{l},y_{mn}^{l}\right)=&-\alpha\left(1-p_{mn}^{l}\right)^{\gamma}y_{mn}^{l}\lg\left(p_{mn}^{l}\right)-\\&(1-\alpha)\,p_{mn}^{l}{}^{\gamma}\left(1-y_{mn}^{l}\right)\lg\left(1-p_{mn}^{l}\right)\end{aligned} \tag{10}$$

其中, y_{mn}^{l}为样本$\bar{\boldsymbol{c}}_{mn}^{l}$的标签, p_{mn}^{l}表示$\bar{\boldsymbol{c}}_{mn}^{l}$被预测为异常样本的概率, $\alpha(0\leqslant\alpha\leqslant1)$为平衡因子, 用于平衡异常和正常样本的权重, 通常取 0.25, $\gamma(\gamma\geqslant1)$为调节因子, 较大的γ值更能降低简单样本的损失贡献, 使训练时更关注分类困难的样本, 通常设置为 2. 基于单个样本的分类损失函数值, 可以计算全局平均损失函数值:

$$\mathcal{L}_{\mathrm{avg}}=\frac{1}{|\mathcal{T}|}\sum_{m,n,l}\mathcal{L}\left(p_{mn}^{l},y_{mn}^{l}\right) \tag{11}$$

其中, $|\cdot|$表示集合的势. 最后, 使用 AdamW 优化器[29] 来优化全局平均损失函数, 获得最优模型参数. 优化过程表示为$\theta=\theta-\eta\nabla_{\theta}\mathcal{L}_{\mathrm{avg}}$, 其中$\theta$为待训练参数, η为学习率, ∇为梯度算子.

本文所提出的 Bot-GECF 模型训练算法如表 1 所示. 算法的输入为原始网络通信数据集T_{o}、模型初始参数θ_{init}、图卷积层总数K、最大训练轮数M、早停轮数NUM、早停阈值δ, 超参数α、h、γ和η, 输出为模型最优参数$\hat{\theta}$. 第 1 ~ 2 行对训练集进行组合采样, 并构建通信图模型和初始特征向量. 第 3 行进行参数初始化. 第 4 ~ 19 行为 GNN 训练过程, 其中第 5 ~ 8 行生成节点聚合特征, 第 9 ~ 10 行生成增强的通信特征, 第 11 行识别异常通信, 第 12 ~ 13 行计算归一化的全局平均损失函数, 第 14 ~ 18 行依据损失函数更新模型参数直至训练结束. 第 20 行返回模型最优参数.

本算法主要分为数据预处理、特征构建及模型训练三部分. 假设网络中共包含N个设备, 在数据预处理阶段, 对原始网络通信数据集T_{o}进行遍历采样, 若任意两个节点之间最大通信次数不超过M($M\ll N$), 则网络中通信总数不超过MN^2, 因此时间复杂度为$O(N^2)$; 在特征构建阶段, 需要提取N个节点和最多MN^2次通信的初始特征向量, 因此时间复杂度为$O(N^2)$; 在模型训练阶段, 每个节点特征的更新操作最多涉及N个邻居节点的信息聚合, 因此时间复杂度为$O(N^2)$. 综合以上分析, 算法 1 的总体时间复杂度为$O(N^2)$.

3 实验及分析

3.1 评估指标

为了综合评估 Bot-GECF 模型对僵尸网络异常通信的检测效果, 本文采用准确率 (Accuracy)、精确率 (Precision)、召回率 (Recall) 和F1分数 (F1-Score) 等指标来评价检测精度, 各指标定义如下.

(1)准确率 (Accuracy) 表示正确预测类别的样本数占总样本数的比例:

$$\mathrm{Accuracy}=\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \tag{12}$$

其中, TP和TN分别表示被正确预测为异常通信和正常通信的样本数, FP和FN分别表示被错误预测为异常通信和正常通信的样本数.

(2)精确率 (Precision) 表示正确预测为异常通信的样本数占所有预测为异常通信的样本数的比例:

$$\mathrm{Precision}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{13}$$

(3)召回率 (Recall) 表示正确预测为异常通信的样本数占异常通信总样本数的比例:

$$\mathrm{Recall}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{14}$$

(4)F1分数 (F1-Score) 为精确率和召回率的调

表 1 Bot-GECF 模型训练算法

Table 1 Bot-GECF model training algorithm

Algorithm 1. Bot-GECF model training algorithm

Input: the original dataset $\mathcal{T}_{\mathrm{o}}$, initial model parameters θ_{init}, the number of graph convolutional layers K, the maximum number of training epochs M, early stopping rounds NUM, the early stopping threshold δ, hyperparameters α、h、γ and η.

Output: the optimal model parameters $\hat{\theta}$.

Step1: Perform combination sampling on $\mathcal{T}_{\mathrm{o}}$ to obtain a balanced training set $\mathcal{T}$;

Step2: Construct $G(\mathcal{V},\boldsymbol{A})$ based on $\mathcal{T}$, and extract the initial node and communication features.

Step3: Initialize parameters: $\theta \leftarrow \theta_{\mathrm{init}}$, $\mathcal{L}_{\mathrm{avg}} \leftarrow 0$, $\mathcal{L}_{\mathrm{avg\text{-}pre}} \leftarrow 0$;

Step4: for epoch = 1 to M **do**:

  Step5: for k = 1 to K **do**:

    $\boldsymbol{V}^k \leftarrow \sigma\left(\boldsymbol{V}^{k-1}\boldsymbol{W}_{\mathrm{s}}^k + \tilde{\boldsymbol{A}}\boldsymbol{V}^{k-1}\boldsymbol{W}_{\mathrm{c}}^k + \boldsymbol{B}^k\right)$; //Node feature aggregation and update via GNN propagation.

  end

  Step6: $\boldsymbol{V} \leftarrow \left(\mathrm{Att}_1(\boldsymbol{V}^K),\cdots,\mathrm{Att}_h(\boldsymbol{V}^K)\right)\boldsymbol{W}_{\mathrm{O}}$; //Node feature fusion via multihead attention.

  Step7: $\boldsymbol{c}_{mn}^{l\prime} \leftarrow \mathrm{PReLU}\left(\mathrm{LayerNorm}\left(\boldsymbol{c}_{mn}^{l}\boldsymbol{W}_{\mathrm{e}} + \boldsymbol{b}_{\mathrm{e}}\right)\right)$; //Normalize initial communication features.

  Step8: $\bar{\boldsymbol{c}}_{mn}^{l} \leftarrow \left(\boldsymbol{V}_m, \boldsymbol{V}_n, \boldsymbol{c}_{mn}^{l\prime}\right)$; //Enhance communication features via node features.

  Step9: $p_{mn}^{l} \leftarrow \mathrm{softmax}(\mathrm{MLP}(\bar{\boldsymbol{c}}_{mn}^{l}))$; //Abnormal communication identification via MLP

  Step10: $\mathcal{L}\left(p_{mn}^{l}, y_{mn}^{l}\right) \leftarrow -\alpha\left(1-p_{mn}^{l}\right)^{\gamma} y_{mn}^{l} \log\left(p_{mn}^{l}\right) - (1-\alpha)\, p_{mn}^{l\,\gamma}\left(1-y_{mn}^{l}\right)\log\left(1-p_{mn}^{l}\right)$; //Calculate the loss value.

  Step11: $\mathcal{L}_{\mathrm{avg\text{-}pre}} \leftarrow \mathcal{L}_{\mathrm{avg}}$, $\mathcal{L}_{\mathrm{avg}} \leftarrow (1/|\mathcal{T}|)\sum_{m,n,l}\mathcal{L}\left(p_{mn}^{l}, y_{mn}^{l}\right)$; //Loss normalization.

  Step12: if $\left|\mathcal{L}_{\mathrm{avg}} - \mathcal{L}_{\mathrm{avg\text{-}pre}}\right|$remains below δ for NUM consecutive rounds **then**: //Check for early stop

    break;

  else:

    $\theta \leftarrow \theta - \eta\nabla_{\theta}\mathcal{L}_{\mathrm{avg}}$; //Update model parameters via gradient descent.

    end

  end

Step13: return $\hat{\theta} \leftarrow \theta$; //Return the optimal model parameters.

和平均值, 用于衡量模型的综合性能:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{15}$$

3.2 数据集和实验环境

本实验在捷克技术大学 (Czech Technical University, CTU) 网络安全实验室提供的公开僵尸网络通信数据集 CTU-13[30] 上进行. 该数据集包含 13 个独立场景, 每个场景模拟了不同规模网络感染特定类型僵尸病毒后的通信情况. 表 2 展示了各场景下僵尸网络通信的数量和占比情况, 可以看出原始数据集呈现明显的不平衡性. 此外, 本文实验所基于的硬件环境为 Intel Xeon Silver 4210R (2.40 GHz) CPU, NVIDIA GeForce RTX 2080 Ti GPU, 11 GB 显存, 软件环境为 Ubuntu 操作系统, PyTorch 2.5.1 深度学习框架, CUDA 版本为 11.4.

3.3 实验对比与分析

首先, 为了消除异常流量极端稀疏的场景对实验结果的影响, 本文将场景 7 剔除. 然后, 在剩余 12 个场景下, 对原始数据集进行组合采样, 并划分为 80% 的训练集和 20% 的测试集. 最后, 在训练集上训练模型, 并在测试集上测试模型. 表 3 的结果表明本文提出的 Bot-GECF 模型对 12 个场景中僵尸网络异常通信的检测性能都很优秀, 各项指标值都非常接近于 1.

(1) 消融实验.

为了验证所提出的细粒度节点特征和通信特征对僵尸网络异常通信检测的重要性, 本文进行了对比实验. 首先, 对节点和通信初始特征进行简化, 仅保留基础通用特征. 节点特征仅保留通信连接数, 通信特征仅保留持续时间、协议类型、源端口、目标端口、连接状态、服务类型、数据包总数、字节总数和源字节数. 然后, 我们选取了 6 个具有不同僵尸网络类型和不同异常流量占比的场景, 对基于基础通用特征和细粒度特征的算法检

表 2 CTU-13 数据集概览

Table 2 Overview of CTU-13 dataset

Scenario	Total flows	Botnet flows (Percentage of total flows)	Bot	Scenario	Total flows	Botnet flows (Percentage of total flows)	Bot
1	2824636	39933 (1.41%)	Neris	8	2954230	5052 (0.17%)	Murlo
2	1808122	18839 (1.04%)	Neris	9	2753884	179880 (6.53%)	Neris
3	4710638	26759 (0.57%)	Rbot	10	1309791	106315 (8.12%)	Rbot
4	1121076	1719 (0.15%)	Rbot	11	107251	8161 (7.61%)	Virut
5	129832	695 (0.54%)	Virut	12	325471	2143 (0.66%)	NSIS.ay
6	558919	4431 (0.79%)	Menti	13	1925149	38791 (2.01%)	Virut
7	114077	37 (0.03%)	Sogou				

表 3 不同场景下的 Bot-GECF 模型检测性能

Table 3 Detection performance of Bot-GECF under different scenarios

Scenario	Precision	Recall	F1-Score	Accuracy	Scenario	Precision	Recall	F1-Score	Accuracy
1	0.998211	0.998698	0.998454	0.998608	8	1	0.998912	0.999456	0.998912
2	0.991306	0.998249	0.994765	0.998249	9	0.996960	0.998739	0.997849	0.998739
3	0.993201	0.998384	0.995786	0.998384	10	0.999875	0.999781	0.999828	0.999781
4	0.924731	1	0.960894	1	11	1	0.998775	0.999387	0.998775
5	1	1	1	1	12	0.990683	0.981538	0.986090	0.981538
6	0.996413	1	0.998203	1	13	0.998750	0.998417	0.998583	0.998417

测效果进行了对比, 结果如图 3 所示. 从实验结果中可以看出, 使用细粒度特征与使用基础通用特征相比, 模型的检测能力更强. 这表明细粒度特征能更准确且精细地表达节点历史行为模式和单次网络通信特征, 从而有效提升模型对异常通信的检测能力.

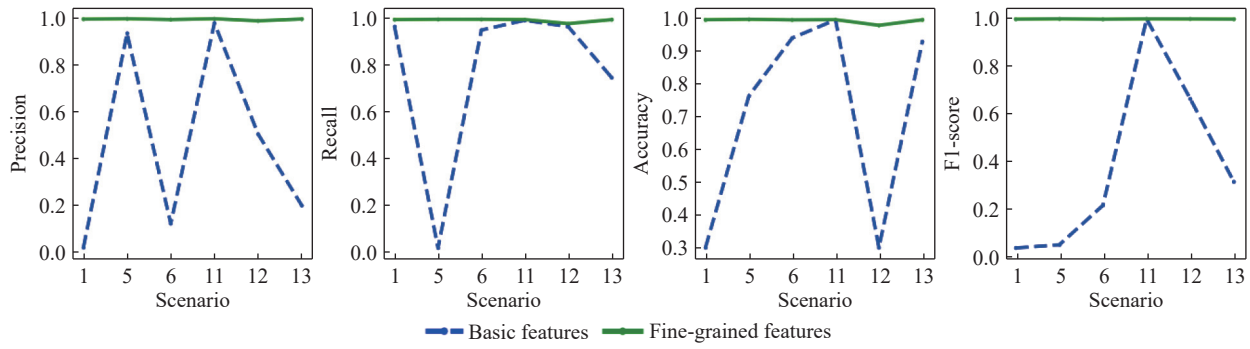

图 3 基于细粒度特征和基础通用特征的算法检测性能对比

Fig.3 Detection performance comparison of algorithms based on fine-grained and basic features

为验证节点特征对通信特征的增强作用, 本文对基于初始通信特征和增强通信特征的算法检测性能进行了对比. 图 4 的结果表明增强的通信特征能显著提升模型的检测精度, 尤其对Precision和F1-Score指标值的提升较为显著. 这表明在单次通信异常识别过程中, 充分考虑发起此次通信的源节点和目标节点的历史行为模式, 能够显著提升模型的检测能力.

针对原始数据集类别不平衡问题, 本文在 6 个不同场景中验证了组合采样技术对模型检测性能的影响. 表 4 的实验结果表明, 组合采样技术通过数据集中平衡正常样本和异常样本的比率, 有效缓解异常样本稀疏性对检测性能造成的不良影响, 从而提升各项性能指标值. 其中, 对Precision和F1-Score均值的提升效果非常显著, 从未进行组合采样时 0.888197 和 0.933977 提升到了组合采样时

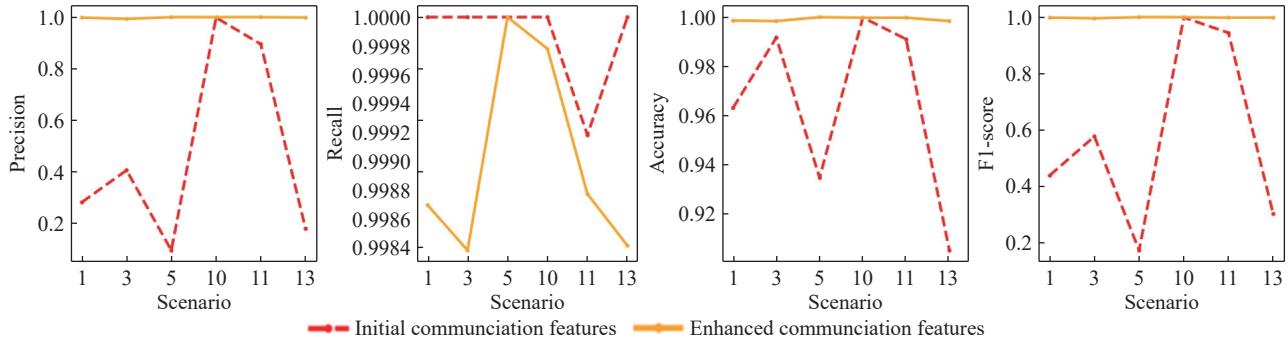

图 4　基于初始通信特征和增强通信特征的算法检测性能对比

Fig.4　Detection performance comparison of algorithms based on initial and enhanced communication features

的 0.983557 和 0.989477. 这也表明, 组合采样技术的引入能够显著降低模型的误报率和综合性能, 其中误报率的显著降低, 能够有效减轻人工排查负担, 更好地保障工业系统连续稳定地运行.

表 4　SMOTE-Tomek 组合采样对检测性能的影响

Table 4　Influence of SMOTE-Tomek sampling on detection performance

Scenario	Without SMOTE-Tomek sampling				With SMOTE-Tomek sampling			
	Precision	Recall	Accuracy	F1-score	Precision	Recall	Accuracy	F1-score
1	0.952924	0.999919	0.999283	0.975856	0.998211	0.998698	0.998608	0.998454
2	0.962611	1	0.999550	0.995	0.991306	0.998249	0.998249	0.994765
4	0.957921	1	0.999899	0.978508	0.924731	1	1	0.960894
6	0.990728	1	0.999922	0.995342	0.996413	1	1	0.998203
8	0.934385	0.999456	0.999853	0.965825	1	0.998912	0.998912	0.998456
12	0.530612	1	0.994111	0.693333	0.990683	0.981538	0.981538	0.986090
Average	**0.888197**	**0.999896**	**0.998770**	**0.933977**	**0.983557**	**0.996233**	**0.996218**	**0.989477**

表 5 展示了多头注意力机制的引入对模型检测性能的影响. 实验结果表明, 引入多头注意力机制能在一定程度上提升 Recall 和 F-Score 值, 其中 Recall 均值从 0.9925 提升至 0.9967, F1-Score 均值从 0.9899 提升至 0.9907. 这主要是由于多头注意力机制能以不同的方式对节点特征进行加权聚合, 挖掘相邻节点之间的多样化关联关系, 实现更加灵活、准确的节点特征表示, 促进提升模型攻击检测性能. Recall 值的提升表明模型能够有效降低恶意攻击漏报率. 而在工业环境中, 即使极少数的恶意攻击漏报, 也会造成重大安全事故或经济损失, 因此多头注意力机制的引入对降低系统安全风险有重要意义.

表 5　多头注意力机制对模型检测性能的影响

Table 5　Influence of multihead attention on detection performance

Scenario	Without multi-head attention				With multi-head attention			
	Precision	Recall	Accuracy	F1-score	Precision	Recall	Accuracy	F1-score
4	0.932530	1	0.998368	0.965087	0.924731	1	1	0.960894
5	1	0.996296	0.999974	0.998145	1	1	1	1
6	1	0.994240	0.999952	0.997112	0.996413	1	1	0.998203
10	0.999875	0.999687	0.999964	0.999781	0.999875	0.999781	0.999781	0.999828
11	1	0.998775	0.999907	0.999387	1	0.998775	0.998775	0.999387
12	0.993671	0.966154	0.999734	0.979719	0.990683	0.981538	0.981538	0.986090
Average	**0.987679**	**0.992525**	**0.999650**	**0.989872**	**0.985284**	**0.996682**	**0.996682**	**0.990734**

(2) 与其他算法的对比实验.

此外, 本文将 Bot-GECF 和文献 [13] 中的 CNN、LSTM、CNN–LSTM 三种算法进行了检测性能对比. 这些算法是目前僵尸网络异常通信检测领域

中使用较为广泛的几种深度学习模型, 检测性能较好且具有一定的代表性. 图 5 的对比结果表明 Bot-GECF 在所有性能指标上均表现最佳. 其中, CNN 模型主要将网络流量数据转换为二维空间数据, 并通过卷积操作实现特征提取, LSTM 模型将网络流量数据建模为时间序列数据, 并挖掘流量数据的时序依赖关系, CNN–LSTM 模型融合了 CNN 空间特征提取与 LSTM 时序特征提取机制, 因此比前两者具有更强的特征表达能力. 然而, 这些模型都无法对节点之间的拓扑结构进行建模, 难以表征节点之间的局部和全局依赖关系, 导致节点特征表示不够准确, 使模型检测性能相对较低. 相比之下, 本文提出的 Bot-GECF 模型利用图模型对设备拓扑结构进行准确建模, 并用图神经网络实现节点信息的传播与聚合, 从而更准确地表示节点特征. 同时, 引入节点特征对通信特征的增强机制, 在检测单次通信过程中充分考虑发起此次通信的源节点和目标节点的历史行为信息, 从而有效提升了模型对异常通信的识别能力. 因此, Bot-GECF 比 CNN, LSTM 和 CNN–LSTM 具有更优秀的检测能力.

此外, 本文将所提出的方法与僵尸网络异常通信检测领域的新技术 Bot-DM[21] 进行对比. Bot-DM 通过挖掘流量负载信息来增强异常检测, 一方面利用多层 Transformer 编码器建模负载字节间隐式语义关系, 另一方面采用网络流量图像表征, 通过深度神经网络捕获不同字节的空间关联, 最终通过最大化两者互信息实现检测. Bot-DM 同样在

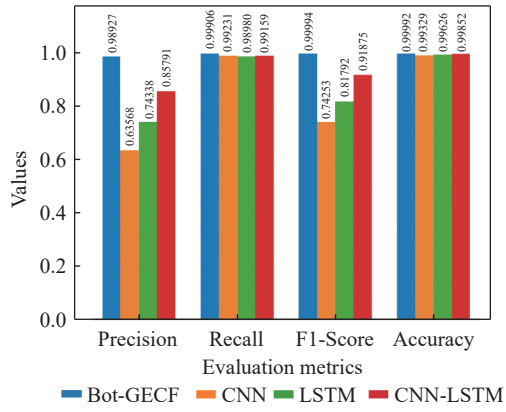

图 5 不同方法的检测性能平均值对比

Fig.5 Detection performance comparison of different approaches

CTU-13 数据集上进行实验验证, 并在该数据集提供的多个场景下都取得了较高的检测精度, 因此用于本文对比实验. 表 6 展示了在文献 [21] 给定的 5 个场景下, Bot-DM 与 Bot-GECF 的检测对比结果. 结果表明 Bot-GECF 在不同场景下的性能指标基本都优于 Bot-DM, 即各项指标的平均值都有较明显的提升. Bot-DM 方法虽然也表现出较强的检测能力, 但需要借助对大量数据包负载信息的分析, 显著加重了模型计算开销. 相比之下, Bot-GECF 无需进行负载分析, 而是借助设备拓扑结构建模和图神经网络的信息传播与聚合机制, 实现更准确的特征表达, 并能在保持高检测精度的同时, 减少模型计算开销, 有效降低实际工业场景中因异常漏报和误报带来的重大安全风险和人工核查负担.

表 6 Bot-GECF 与 Bot-DM 的检测性能对比

Table 6 Detection performance comparison of Bot-GECF and Bot-DM

Scenario	Bot-DM				Bot-GECF			
	Precision	Recall	Accuracy	F1-score	Precision	Recall	Accuracy	F1-score
1	0.992	0.9945	0.9947	0.9935	0.998211	0.998698	0.998608	0.998454
2	0.995	0.995	0.9964	0.995	0.991306	0.998249	0.998249	0.994765
3	0.997	0.9985	0.9984	0.998	0.993201	0.998384	0.998384	0.995786
9	0.987	0.9905	0.99	0.989	0.996960	0.998739	0.998739	0.997849
13	0.9795	0.9885	0.9896	0.9835	0.998750	0.998417	0.998417	0.998583
Average	**0.9901**	**0.9934**	**0.99382**	**0.9918**	**0.995686**	**0.998497**	**0.998479**	**0.997087**

4 结论

为了提高僵尸网络异常通信检测精度, 本文提出一种基于图神经网络增强通信特征的僵尸网络异常通信检测模型 Bot-GECF.

(1) Bot-GECF 充分挖掘网络通信中节点和通信的细粒度流量特征及连接特征, 全面描述网络通信行为, 为僵尸网络异常通信检测提供丰富的特征信息, 并用图模型建模设备拓扑结构, 充分利用设备之间的空间关系提升异常通信检测效果.

(2) 利用图神经网络的信息传播和聚合机制,

生成充分融合重要上下文信息的节点聚合特征, 并用节点聚合特征增强通信特征, 生成更准确的通信特征表示, 进一步利用多层感知机 MLP 模型实现对僵尸网络异常通信的准确识别.

(3) 综合实验结果表明, Bot-GECF 僵尸网络异常通信检测技术对不平衡数据集具有很高的异常识别精度, 模型综合检测能力优秀. 本文提出的 Bot-GECF 模型通过适当修改也适用于其他网络异常检测场景.

尽管本文提出的 Bot-GECF 方法能够有效地检测僵尸网络异常通信, 但仍存在以下不足之处:

(1) 本文所构建的复杂网络结构为静态结构, 无法适应复杂网络随时间动态变化的规律. 未来将探索动态图神经网络或时序感知机制在僵尸网络检测中的应用, 增强模型对时变通信行为的建模能力.

(2) 本文通过人工选取并计算节点及通信特征, 具有一定的主观性, 后续可探索自动特征选择机制, 进一步提升模型的对特征的自动处理和筛选能力.

(3) 本文实验基于 CTU-13 公开数据集, 虽在僵尸网络研究领域具有一定代表性, 但与真实工业互联网场景仍存在一定差距, 后续将进一步探索工业互联网仿真场景下的僵尸网络数据集获取或构建方式, 以更好地验证方法的有效性.

参 考 文 献

[1] Cacciuttolo C, Atencio E, Komarizadehasl S, et al. Internet of Things long-range-wide-area-network-based wireless sensors network for underground mine monitoring: Planning an efficient, safe, and sustainable labor environment. *Sensors*, 2024, 24(21): 6971

[2] Cacciuttolo C, Guzmán V, Catriñir P, et al. Sensor technologies for safety monitoring in mine tailings storage facilities: Solutions in the industry 4.0 era. *Minerals*, 2024, 14(5): 446

[3] Lange T, Kettani H. On security threats of botnets to cyber systems // *Proceedings of the 6th International Conference on Signal Processing and Integrated Networks* (*SPIN* 2019). Noida, 2019: 176

[4] Salem A H, Azzam S M, Emam O E, et al. Advancing cybersecurity: A comprehensive review of AI-driven detection techniques. *J Big Data*, 2024, 11(1): 105

[5] Wazzan M, Algazzawi D, Bamasaq O, et al. Internet of Things botnet detection approaches: Analysis and recommendations for future research. *Applied Sci*, 2021, 11(12): 5713

[6] Aljarrah S J, Cherbal S, Mashaleh A, et al. Optimizing network traffic anomaly detection with normalized features // *Proceedings of the* 2024 *International Jordanian Cybersecurity Conference* (*IJCC*). Amman, 2024: 143

[7] Al-Mashhadi S, Anbar M, Hasbullah I, et al. Hybrid rule-based botnet detection approach using machine learning for analysing DNS traffic. *PeerJ Comput Sci*, 2021, 7: e640

[8] Haq S, Singh Y. Botnet detection using machine learning // *Proceedings of the 5th International Conference on Parallel, Distributed and Grid Computing* (*PDGC* 2018). Solan, 2018: 240

[9] Khan R U, Zhang X S, Kumar R, et al. An adaptive multi-layer botnet detection technique using machine learning classifiers. *applied Sci*, 2019, 9(11): 2375

[10] Mohan H G, Kumar J, Rajesh I S, et al. A CNN based deep learning model for detecting P2P Botnets using flow features // 2024 *Second International Conference on Networks, Multimedia and Information Technology* (*NMITCON*). Bengaluru, 2024: 1

[11] Padmavathi B, Muthukumar B. A deep recursively learning LSTM model to improve cyber security botnet attack intrusion detection. *Int J Model Simul Sci Comput*, 2023, 14(2): 2341018

[12] Fu Y F, Du Y S, Cao Z J, et al. A deep learning model for network intrusion detection with imbalanced data. *Electronics*, 2022, 11(6): 898

[13] Nugraha B, Nambiar A, Bauschert T. Performance evaluation of botnet detection using deep learning techniques // *Proceedings of the* 11*th International Conference on Network of the Future* (*NoF* 2020). Bordeaux, 2020: 141

[14] Lu F Q, Chen D W. Botnet identification method based on improved CNN–LSTM fusion. *Comput Appl Softw*, 2024, 41(3): 328
(卢法权, 陈丹伟. 基于改进 CNN–LSTM 融合的僵尸网络识别方法. 计算机应用与软件, 2024, 41(3): 328)

[15] Pektaş A, Acarman T. Deep learning to detect botnet *via* network flow summaries. *Neural Comput Appl*, 2019, 31(11): 8021

[16] Azab A, Khasawneh M, Alrabaee S, et al. Network traffic classification: Techniques, datasets, and challenges. *Digital Commun Networks*, 2024, 10(3): 676

[17] Lagraa S, Husák M, Seba H, et al. A review on graph-based approaches for network security monitoring and botnet detection. *Int J Inf Secur*, 2024, 23(1): 119

[18] Zhou J, Xu Z, Rush A M, et al. Automating botnet detection with graph neural networks [J/OL]. *arXiv preprint* (2020−03−13) [2024−11−24]. https://arxiv.org/abs/2003.06344

[19] Meng X, Bo L, Yan Y, et al. Deeply fused flow and topology features for botnet detection based on a pretrained GCN [J/OL]. *arXiv preprint* (2024−05−25) [2024−11−24]. https://arxiv.org/abs/2307.10583

[20] Zhao J, Liu X D, Yan Q B, et al. Multi-attributed heterogeneous graph convolutional network for bot detection. *Inf Sci*, 2020, 537: 380

[21] Wu G L, Wang X Y, Lu Q, et al. Bot-DM: A dual-modal botnet detection method based on the combination of implicit semantic expression and graphical expression. *Expert Syst Appl*, 2024, 248: 123384

[22] Hairani H, Anggrawan A, Priyanto D. Improvement performance

of the random forest method on unbalanced diabetes data classification using smote-tomek link. *Int J Inf Visualization*, 2023, 7(1): 258

[23] Low M X, Yap T T V, Soo W K, et al. Comparison of label encoding and evidence counting for malware classification. *J Syst Manage Sci*, 2022, 12(6): 17

[24] Henderi H, Wahyuningsih T, Rahwanto E. Comparison of Min–Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer. *IJIIS*, 2021, 4(1): 13

[25] Shehzad A, Xia F, Abid S, et al. Graph transformers: A survey [J/OL]. *arXiv preprint* (2024−07−13) [2024−11−24]. https://arxiv.org/abs/2407.09777

[26] Huang L, Qin J, Zhou Y, et al. Normalization techniques in training DNNs: methodology, analysis and application. *IEEE Trans Pattern Anal Mach Intell*, 2023, 45(8): 10173

[27] Jiang T T, Cheng J Y. Target recognition based on CNN with LeakyReLU and PReLU activation functions // 2019 *International Conference on Sensing, Diagnostics, Prognostics, and Control* (*SDPC*). Beijing, 2019: 718

[28] Naskath J, Sivakamasundari G, Begum A A S. A study on different deep learning algorithms used in deep neural nets: MLP SOM and DBN. *Wirel Pers Commun*, 2023, 128(4): 2913

[29] Zhou P, Xie X Y, Lin Z C, et al. Towards understanding convergence and generalization of AdamW. *IEEE Trans Pattern Anal Mach Intell*, 2024, 46(9): 6486

[30] García S, Grill M, Stiborek J, et al. An empirical comparison of botnet detection methods. *Comput Secur*, 2014, 45: 100