文章编号:1671-6833(2025)01-0034-08

基于图重构和子图挖掘的僵尸网络检测方法

景永俊[1,2], 吴 悔[2], 陈 旭[2], 宋吉飞[3]

(1. 合肥工业大学 计算机与信息学院,安徽 合肥 230601;2. 北方民族大学 计算机科学与工程学院,宁夏 银川 750021;3. 国家(中卫)新型互联网交换中心,宁夏 中卫 755000)

摘 要:针对伪装后僵尸网络主机难以检测的问题,提出一种基于图重构和子图挖掘的僵尸网络检测方法(GR-SGM)。首先,将网络数据转化为图数据,并对其进行重构以此增强主机节点特征表示;其次,基于重构图中拓扑结构、节点的特征和位置变化设计僵尸网络子图评分函数,以此捕捉伪装后的特征,提取出僵尸网络子图,并对原始图和重构图进行预检测,以提高检测的准确率和效率,减少重构误差;最后,对预检测结果和僵尸网络子图进行综合评分,以获取完整的僵尸网络信息。在 ISCX2014 僵尸网络数据集和 CICIDS2017 僵尸网络数据集上的实验结果表明:GR-SGM 的检测准确率分别达到 99.98%和 99.91%,$F1$ 分别达到 99.94%和 99.65%,相较于其他僵尸网络检测模型,GR-SGM 能更加高效准确地识别僵尸网络节点,同时具有更低的误报率。

关键词:僵尸网络; 子图挖掘; 图重构; 网络安全; 预检测

中图分类号:TP391;TP393 **文献标志码**:A **doi**:10.13705/j.issn.1671-6833.2024.04.004

僵尸网络作为一种常见的恶意软件攻击手段,利用恶意软件感染大量计算机,使之处于“僵尸”状态以随时接收攻击指令[1]。为有效地检测僵尸网络,安全研究人员提出和应用了各种方法。目前,僵尸网络检测方法可以分为两大类:基于流量的僵尸网络检测方法和基于图的僵尸网络检测方法。

基于流量的僵尸网络检测方法主要通过分析单个的流量特征来进行检测。例如,Srinivasan 等[2]提出一种带有堆叠过程的集成分类器算法,通过混合不同的机器学习模型以改进预测性能。然而,机器学习需要大量的特征工程和相关领域知识,研究人员把目光转向了深度学习[3]。Haq[4]提出一种基于深度神经网络僵尸网络检测模型,在长短记忆网络的基础上进行改进,以极小的计算开销实现了较好的效果。然而,基于流量的僵尸网络检测主要关注单个数据流,未能充分利用数据流之间隐藏的结构信息,这限制了其在实际中的适用性。因此,研究人员开始转向基于图的僵尸网络检测方法。Zhao 等[5]通过构建一个多属性异构信息图,将僵尸网络检测问题转换为图上的节点分类问题。Joshi 等[6]提出一种基于强化的方法,解决了在动态通信图中的僵尸网络检测问题,提高了检测的精度和召回率,适用于大型通信图。然而,在僵尸网络中存在大量隐藏的感染主机,攻击者模仿正常主机的通信模式或者伪造 DNS 等方式来达到隐藏效果,使得在传统图神经网络检测中,这些主机的节点特征、边的特征都表现正常,但实际为僵尸网络节点,这导致传统图神经网络检测精度不佳,并存在大量的漏检情况。

为了解决上述问题,本文提出一种基于图重构和子图挖掘的僵尸网络检测方法(GR-SGM)。首先,通过提取位置感知特征以及对僵尸网络图数据中的节点和边信息进行编码和解码,生成新的图;其次,通过僵尸网络子图评分函数得到僵尸网络子图,对原始流量数据图和重构后的图进行初步检测;最后,综合利用僵尸网络子图和预检测的结果检测出完整的僵尸网络信息。在公开数据集 ISCX2014 僵尸网络数据集和 CICIDS2017 僵尸网络数据集上对本文模型进行验证。

1 相关工作

1.1 子图神经网络

图神经网络在节点级和图级任务中均表现出卓

收稿日期:2024-02-18;**修订日期**:2024-04-25

基金项目:宁夏回族自治区重点研发计划(2023BDE02017);中央高校基本科研业务费专项资金(2022PT_S04)

作者简介:景永俊(1977—),男,甘肃天祝人,合肥工业大学博士生,主要从事图数据挖掘与信息安全方面的研究,E-mail:jingyj@nmu.edu.cn。

引用本文:景永俊,吴悔,陈旭,等. 基于图重构和子图挖掘的僵尸网络检测方法[J]. 郑州大学学报(工学版),2025,46(1):34-41. (JING Y J, WU H, CHEN X, et al. Botnet detection method based on graph reconstruction and subgraph mining[J]. Journal of Zhengzhou University (Engineering Science),2025,46(1):34-41.)

越的性能,但其忽略了子图的重要性。子图能更有效地捕获和挖掘图数据中的结构信息,并且能更好地处理数据中的噪声和不完整信息。例如,Alsentzer 等[7]提出子图神经网络,采用一种独特的子图路由机制,在子图的组件和底层图中随机采样的锚块之间传播神经消息,从而到达高精度。为了更加高效地挖掘适合的子图数据,Li 等[8]提出一种新型的自适应子图神经网络,设计一个增强子图检测模块,即使在没有预定义相应规则的情况下也能自适应地去搜索需要的子图。为了增强图中节点特征表示,Zhang 等[9]提出一种异常子图自编码器,以无监督和弱监督的方式去提取异常子图。

1.2　僵尸网络检测

近年来,各类僵尸网络检测方法被陆续提出。例如,Velasco-Mata 等[10]提出一种基于机器学习的检测方法,利用决策树和一些简单特征快速分析大规模网络宽带的流量数据,达到了较高的速率和准确率。然而,人工特征工程量大、模型通用性较差等缺点使得机器学习受限,因此 Shahhosseini 等[11]提出一种基于深度学习的网络流量分析器用于僵尸网络检测,该方法可以自动提取相应的特征,取得了 97.13%的高准确率。为了加强检测性能,Tulasi Ratnakar 等[12]提出一种基于双向门控循环单元的僵尸网络检测模型,其性能优于传统的 GRU 模型。为了更全面地考虑被感染主机之间的拓扑结构,Zhou 等[13]应用图神经网络来检测僵尸网络,能有效捕获集中式僵尸网络的重要层次结构和分散式僵尸网络的快速混合结构。

2　本文方法

本文提出的基于图重构和子图挖掘的僵尸网络检测方法 GR-SGM 的基本框架如图 1 所示。首先,采集网络流量的原始数据,并通过一系列预处理步骤来提取出关键特征;其次,利用数据构建网络流量图,并基于图自编码器进行图重构;再次,本文设计僵尸网络子图评分函数,准确地挖掘出僵尸网络子图,并对原始网络流量图和重构图进行预检测;最后,将预检测结果和子图进行综合评估,以获取完整的僵尸网络信息。

2.1　数据预处理

在网络流量分析中特征的选择显得尤为重要。本文使用流量分析工具 CICFlowmeter 对原始的 pcap 文件进行深度解析,成功地提取出一系列统计特征,包括正向总包数、反向总包数、正向数据包的总大小、正向数据包的最大大小、正向包的最小大小、正向数据包的平均大小等 80 多个特征,这些统计特征仅基于对数据包数量的分析,简化了处理过程。该方法不依赖于数据包的具体通信内容,因此对单个节点内容的变更不敏感,有效规避攻击者通过加密僵尸网络通信内容对检测准确性可能造成的负面影响。此外,由于避免了使用通信内容,该方法也起到了保护数据用户隐私的作用。

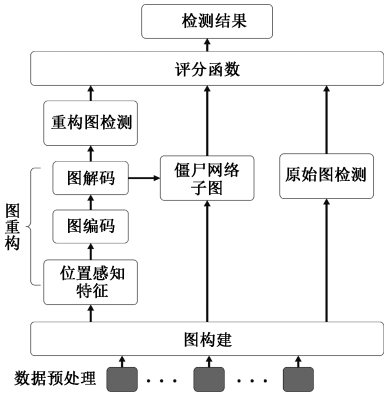


图 1　GR-SGM 模型框架结构

Figure 1　GR-SGM model framework structure

在提取完统计特征后,本文对其进行了筛选,最终选定了 5 个特征用于僵尸网络检测:数据包数量、数据包大小、数据包平均大小、数据包头部字节数和数据包标准偏差大小。正常的网络数据包的数量往往在一个稳定的范围内,而对于僵尸网络而言,数据包数量会有明显的波动或者巨大的峰值。另外,部分僵尸网络可能会通过发送过小或者过大的数据包来进行攻击,这类特征能用于识别那些通过发送相似大小数据包来进行伪装的攻击,因此,本文选取数据包标准偏差大小、数据包大小和平均大小作为特征。在僵尸网络流量中,数据包的头部字节数通常比正常网络流量的少,这是因为攻击者为了躲避安全检测,减小被发现的可能性,会精简数据包头部信息,并且可能通过篡改数据包头部来进行欺骗攻击,因此,本文选取数据包头部字节数作为特征。在特征选择完成后,本文将预处理后的流量数据转换成一个三元组的列表 $\boldsymbol{X}$:

$$\boldsymbol{X}=[\boldsymbol{Srcip},\boldsymbol{Dstip},\boldsymbol{F}]。\tag{1}$$

式中:$\boldsymbol{Srcip}$ 为源 IP 地址;$\boldsymbol{Dstip}$ 为目标 IP 地址;$\boldsymbol{F}=[f_1,f_2,\cdots,f_N]$ 为主机特征。

2.2　流量图构建

在网络连接关系错综复杂的环境下,将数据构建成图可以提供更为直观、专业和全面的视角,从而更充分地揭示僵尸网络的特点,并更有效地发现隐藏在网络数据中的规律和异常行为。因此,本文采用基于图的表示方法,将网络流量数据构建成图数

据。具体而言,本文将原始数据包和列表 $\boldsymbol{X}$ 结合,构建成有向图 $\boldsymbol{G}$:

$$\boldsymbol{G}=[\boldsymbol{V},\boldsymbol{E},\boldsymbol{F}]。\tag{2}$$

式中:$\boldsymbol{V}=[v_1,v_2,\cdots,v_N]$ 为图中所有节点,每个节点代表一个 IP 地址;$\boldsymbol{E}=[e_1,e_2,\cdots,e_M]$ 为节点之间的边,每条边代表两个 IP 地址之间存在着通信关系。

2.3 图重构

为了有效处理僵尸网络的图数据,学习节点间的相似性与差异性并挖掘隐藏的僵尸网络主机,本文提出一种基于图自编码器的图重构方法。

2.3.1 位置感知特征提取

为了优化重构图的准确性并尽可能减小在编码解码过程中产生的信息损失,在编码器中引入位置感知特征。如图 2 所示,本文选取节点的最短路径以及节点的度作为位置感知特征。在计算最短路径的过程中,为了有效降低图中节点最短路径的计算复杂度并提升计算效率,按式(3)设定子图划分阈值 Y,并将 $\boldsymbol{G}$ 图分为 Y 个子图 $[\boldsymbol{G}_1,\boldsymbol{G}_2,\cdots,\boldsymbol{G}_Y]$。

$$Y=N/Q。\tag{3}$$

式中:N 为节点数;Q 为可学习的超参数;阈值 Y 可以根据实际数据和需求进行调整。这种设定方式在保证灵活性的同时,可以保留节点的关键位置信息,又能有效降低计算复杂度,提高图重构的精度和效率。在每个子图中,选择子图的中心节点作为锚节点,并计算子图中每个节点至锚节点的最短路径 $\boldsymbol{P}=[P_1,P_2,\cdots,P_{v_i}]$,其元素为

$$P_{v_i}=\varphi\{v_{Y_i},v_{Y_k}\mid v_{Y_i},v_{Y_k}\in\boldsymbol{G}_Y\}。\tag{4}$$

式中:v_{Y_i} 为子图 $\boldsymbol{G}_Y$ 中的节点;v_{Y_k} 为子图 $\boldsymbol{G}_Y$ 中选取的中心锚节点;φ 为 Dijkstra 算子。同时,提取图中每个节点的度 $\boldsymbol{D}=[out\_d,in\_d]$,其中 out_d 和 in_d 分别表示节点的出度和入度。节点的度是一项关键指标,用以量化节点在图中连接结构的重要性。

2.3.2 图编码

在图重构过程中,将位置感知特征 $\boldsymbol{L}=[\boldsymbol{P},\boldsymbol{D}]$ 与图 $\boldsymbol{G}$ 的节点特征矩阵 $\boldsymbol{F}$ 整合在一起,一同输入到编码器 S 中进行编码,并将它们映射到低维的潜在表示向量 $\boldsymbol{Z}=[Z_1,Z_2,\cdots,Z_i]$:

$$\boldsymbol{Z}=S(\boldsymbol{G}\mid\boldsymbol{F},\boldsymbol{L})。\tag{5}$$

式中:S 为图编码器,用于将特征矩阵 $\boldsymbol{F}$ 与位置感知特征 $\boldsymbol{L}$ 一同映射到图 $\boldsymbol{G}$ 节点上进行编码,得到低维的潜在表示向量 $\boldsymbol{Z}$。

编码过程如图 3 所示,从当前节点的邻居节点中进行随机采样,并将当前节点自身的表示以及其邻居节点的表示进行聚合,形成每一层的节点表示。

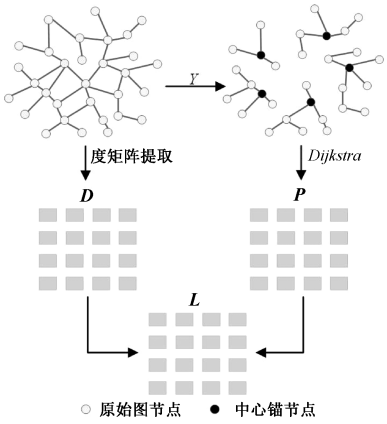


图 2 位置感知特征提取

Figure 2 Location-aware feature extraction

只对邻居节点进行采样,不仅避免了对整个图的计算,减少计算复杂度,而且使得编码器能够针对不同规模类型的僵尸网络节点进行处理。具体公式为

$$\boldsymbol{h}_{v_i}^{k}=\sigma(\boldsymbol{W}\cdot C(\boldsymbol{h}_{N(v_i)}^{k},\boldsymbol{h}_{v_i}^{k-1}))。\tag{6}$$

式中:$\boldsymbol{h}_{v_i}^{k-1}$ 为节点 v_i 在第 $k-1$ 层聚合特征的表示;σ 为激活函数;$\boldsymbol{W}$ 为该层的权重矩阵;$N(v_i)$ 为节点 v_i 的邻居节点集合;$\boldsymbol{h}_{N(v_i)}^{k}$ 为节点 v_i 的邻居节点在 k 层的聚合结果表示;C 为拼接操作。

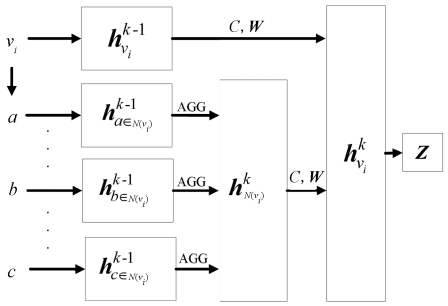


图 3 编码过程

Figure 3 Encoding process

随着聚合次数的增加,每个节点能够积累并聚合更多高阶邻居节点信息,从而更有效地捕捉节点间的相似性。使用平均聚合函数进行特征聚合:

$$\boldsymbol{h}_{N(v_i)}^{k}=\sum_{a\in N(v_i)}\frac{\boldsymbol{h}_{a}^{k-1}}{|N(v_i)|}。\tag{7}$$

式中:$\boldsymbol{h}_{a}^{k-1}$ 为节点 v_i 的邻居节点 a 在 $k-1$ 层的表示。

2.3.3 图解码

编码完成后对低维的潜在表示向量 $\boldsymbol{Z}$ 进行解码得到重构图。通过解码器 De,可以从中提取出重构图所需要的邻接矩阵 $\boldsymbol{A}'$ 和节点特征矩阵 $\boldsymbol{F}'$:

$$\boldsymbol{A}'=\mathrm{De}(\boldsymbol{Z}\mid\boldsymbol{W}_A);\tag{8}$$

$$\boldsymbol{F}'=\mathrm{De}(\boldsymbol{Z}\mid\boldsymbol{W}_F)。\tag{9}$$

式中:$\boldsymbol{W}_A$ 和 $\boldsymbol{W}_F$ 均为权重矩阵。解码过程如图 4 所示,采用多层隐藏层进行解码。首先,每个隐藏层将

输入与权重矩阵和偏差值相结合,并通过线性函数计算。这使得解码器 De 通常只需要较少的训练时间和计算资源,同时也更易于调整和优化。然后,将计算结果输入到激活函数中以添加非线性,使得解码器 De 可以学习到更复杂、更丰富的节点特征表示,从而提高重构图的可靠性,具体公式为

$$\boldsymbol{H}_i = \sigma(\boldsymbol{W}_i\boldsymbol{H}_{i-1} + \boldsymbol{b}_i); \tag{10}$$

$$\boldsymbol{O} = \boldsymbol{W}_o\boldsymbol{H}_i + \boldsymbol{b}_o。 \tag{11}$$

式中:$\boldsymbol{H}_i$ 为编码器第 i 层的特征表示;$\boldsymbol{W}_i$ 和 $\boldsymbol{b}_i$ 分别为解码器第 i 层的权重矩阵和偏差值;$\boldsymbol{W}_o$ 和 $\boldsymbol{b}_o$ 分别为输出层权重矩阵和偏差值;$\boldsymbol{O}$ 为输出结果。最后,通过邻接矩阵 $\boldsymbol{A}'$ 和节点特征矩阵 $\boldsymbol{F}'$ 可以得到重构图 $\boldsymbol{G}'$,为下一步的僵尸网络检测提供可靠基础。

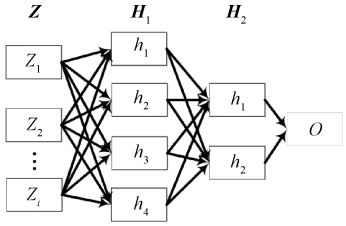


图 4　解码过程

Figure 4　Decoding process

2.4　僵尸网络子图提取

本文提出的僵尸网络子图提取方法通过评估原始图节点和重构图节点之间的差异来判断节点是否存在于僵尸网络子图中。首先,为量化原始图节点和重构图节点之间的差异,设计 1 个对数函数来计算僵尸网络子图分数 $\boldsymbol{b} = \{b_1, b_2, \cdots, b_i\}$,其元素为

$$b_i = y\log(f_i - f'_i) + \sum_{j\in M}^{M}(1-y)\log(A_{ij} - A'_{ij})。 \tag{12}$$

式中:M 为邻接矩阵列数;y 为子图权重,用于调节以及衡量节点自身特征与拓扑结构信息特征之间的重要性。然后,利用 $\boldsymbol{b}$ 来判断节点是否存在于僵尸网络子图中:

$$\boldsymbol{V}_{\text{Botnet}} = \{\bar{v}_i \mid \bar{v}_i \in \boldsymbol{V}, b_i > \theta\}。 \tag{13}$$

式中:$\boldsymbol{V}_{\text{Botnet}}$ 为僵尸网络子图节点;$\theta = \dfrac{1}{N}\sum_{i\in N}^{N} b_i$,$N$ 为节点数。僵尸网络子图 $\boldsymbol{G}_{\text{Botnet}} = [\boldsymbol{V}_{\text{Botnet}}, \boldsymbol{E}_{\text{Botnet}}]$,其中 $\boldsymbol{E}_{\text{Botnet}} = [\bar{e}_1, \bar{e}_2, \cdots, \bar{e}_i]$ 表示 $\boldsymbol{V}_{\text{Botnet}}$ 中节点之间存在的边。

2.5　预检测

图自编码器在进行图重构时可能会由于数据噪声、模型限制和特征丢失等因素导致重构有偏差,影响最后的检测效果。为缓解这种偏差的影响,在原始图和重构图上采用图卷积神经网络进行预检测,快速地对一些僵尸网络特征显著的节点进行初步检测,减小重构后信息丢失造成的影响。具体公式为

$$\boldsymbol{T} = \text{FC}[\text{GCN}_2(\text{GCN}_1(\boldsymbol{A}_T, \boldsymbol{F}_T))]。 \tag{14}$$

式中:$\boldsymbol{T}$ 包含 $\{\boldsymbol{H}_T, \boldsymbol{H'}_T\}$,$\boldsymbol{H}_T$ 为原始图的预检测结果,$\boldsymbol{H'}_T$ 为重构图的预检测结果;$\boldsymbol{A}_T$ 包含 $\{\boldsymbol{A}, \boldsymbol{A}'\}$ 为原始图和重构图的邻接矩阵;$\boldsymbol{F}_T$ 包含 $\{\boldsymbol{F}, \boldsymbol{F}'\}$ 为原始图和重构图的特征矩阵;GCN_1 和 GCN_2 为两层 GCN 层;FC[·]为全连接层。在 GCN 中,每层隐藏层都对主机节点特征进行特征变换,具体公式为

$$\boldsymbol{H}^{l+1} = \sigma(\boldsymbol{A}\boldsymbol{H}^l\boldsymbol{W}^l)。 \tag{15}$$

式中:$\boldsymbol{W}^l$ 为图节点在第 l 层的权重矩阵;$\boldsymbol{H}^l$ 为第 l 层的节点特征。为了充分利用原始图和重构图中节点自身信息以及邻居节点信息,将邻接矩阵加上自连接,这样在特征聚合时不仅聚合邻居节点信息,而且同时聚合了自身节点信息,具体公式为

$$\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}。 \tag{16}$$

式中:$\boldsymbol{I}$ 为单位矩阵;$\tilde{\boldsymbol{A}}$ 为添加自连接后的邻接矩阵。为了防止在消息传递的过程中出现过大或者过小的值,保证节点特征变换的稳定,对邻接矩阵进行对称归一化,从而将式(15)进一步优化为

$$\boldsymbol{H}^{l+1} = \sigma(\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}H^l\boldsymbol{W}^l)。 \tag{17}$$

式中:$\tilde{\boldsymbol{D}}$ 为 $\tilde{\boldsymbol{A}}$ 的度矩阵。

2.6　综合评分

$\boldsymbol{G}$ 中的僵尸网络节点在编码解码中的嵌入和重建特征很可能因为它们的属性或结构异常而被扭曲;此外,单一的预检测可能无法完全检测出所有的僵尸网络。因此,本文将挖掘出的僵尸网络子图和预检测结果进行结合,通过分配不同的权重来衡量僵尸网络子图和预检测结果的重要性,使二者能够有机地结合在一起,以此获得僵尸网络节点特征表示 $\boldsymbol{b}' = \{b'_1, b'_2, \cdots, b'_i\}$,其元素为

$$b'_i = \sigma(\boldsymbol{W}_H\boldsymbol{H}_T + \boldsymbol{W}_{H'}\boldsymbol{H'}_T + \boldsymbol{W}_b b_i)。 \tag{18}$$

式中:$\boldsymbol{W}_H$、$\boldsymbol{W}_{H'}$ 和 $\boldsymbol{W}_b$ 均为权重矩阵。最后,通过 softmax 进行节点分类,将特征表示向量转化为[0,1]的概率分布,令 $\boldsymbol{R} = \{R_1, R_2, \cdots, R_i\}$ 为该节点被判定属于僵尸网络概率,$\boldsymbol{R}$ 值越高,概率越大:

$$\boldsymbol{R} = \text{softmax}(\text{FC}(\boldsymbol{b}'))。 \tag{19}$$

式中:$\boldsymbol{R}$ 为僵尸网络检测的概率向量。训练时用组合损失函数来优化僵尸网络的检测:

$$Loss = Loss_1 + \alpha F(\boldsymbol{G}_{\text{Botnet}})。 \tag{20}$$

式中:$Loss_1 = -\sum_{i=1}^{N} L_i\log R_i$,为交叉熵损失,用于确保模型能够准确分类僵尸网络节点,起到评估模型预测和实际标签之间一致性的关键作用,其中 L_i 为

网络流量实际标签；$F(\boldsymbol{G}_{\text{Botnet}})=\frac{1}{N}\sum_{i=1}^{N}\|b_i\|$，用于评估模型在重构僵尸网络子图时的性能；$\alpha$ 为损失权重，用于衡量重构损失相对于交叉熵损失的重要性，可以确保模型在保持高准确性分类的同时，还能有效重建和识别僵尸网络的细微特征。

3 实验与分析

3.1 实验设置

本文的 GR-SGM 模型基于 DGL 框架和 PyTorch 框架实现，通过使用一台配备两张 Quadro RTX 6000 显卡的实验计算机进行训练，模型实验只用到了计算机小部分的计算资源。实验采用公开网络安全数据集：ISCX2014 僵尸网络数据集[14]和 CICIDS2017 僵尸网络数据集[15]。这两个数据集被广泛认可并用来评估僵尸网络检测模型的性能，可以确保研究的模型在符合真实环境的情况下进行检测。本文将实验数据集划分为训练集、验证集和测试集，比例为 6∶2∶2。

3.2 评估指标

本文采用 4 种评价指标分析检测效果：召回率 *REC*、准确率 *ACC*、精确率 *PRE* 和 *F*1。公式如下。

$$REC=\frac{TP}{TP+FN};\tag{21}$$

$$ACC=\frac{TP+TN}{TP+TN+FP+FN};\tag{22}$$

$$PRE=\frac{TP}{TP+FP};\tag{23}$$

$$F1=\frac{2\times PRE\times REC}{PRE+REC}。\tag{24}$$

式中：*TP* 为该节点是僵尸网络节点，并且被预测为僵尸网络节点数量；*FN* 为该节点是僵尸网络节点，但被预测为正常节点数量；*TN* 为该节点是正常节点数量，并且被预测为正常节点数量；*FP* 为该节点是正常节点，但被预测为僵尸网络节点数量。

3.3 对比实验

为了验证本文所提方法的有效性，将 GR-SGM 与几种先进方法进行对比实验，包括 LSTM. MI[16]、N-BaIoT[17]、GRU[18]、GCN[13]、ABD-GN[19]、GAT[20]和 GraphSAGE[21]。GR-SGM 与对比方法在 ISCX2014 僵尸网络数据集和 CICIDS2017 僵尸网络数据集上进行实验，从图 5 可以看出，在训练过程中，GR-SGM 模型展现出较高的准确率与较低的损失值，并且数值波动小，充分证明了模型的稳定性。通过图 6 中的 TSNE 图可以直观地观察到 GR-SGM 模型学习到的节点表达之间的明显差异性，这表明模型能有效区分僵尸网络节点与正常流量节点。从图 7 可以看出，相比其他对比模型，GR-SGM 模型的 ROC 曲线呈现快速上升趋势，AUC 值高达 0.99，这表明在极低的假正率阈值下，真正率能迅速达到较高水平，说明 GR-SGM 模型能够高效准确地识别僵尸网络节点，同时将误报率维持在极低水平。

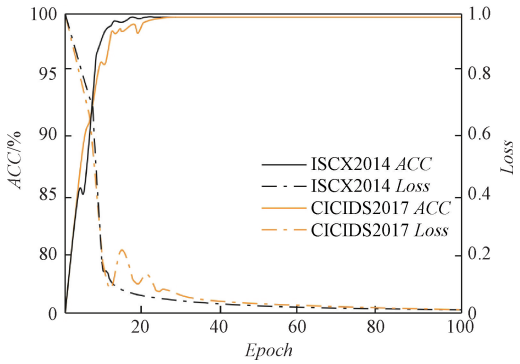
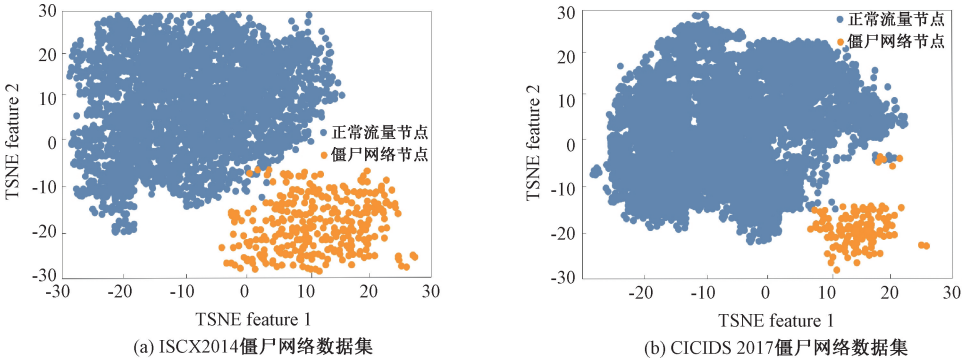


图 5 在两个数据集上训练 GR-SGM 模型的准确率和损失

Figure 5 *ACC* and *Loss* of GR-SGM model trained on two datasets

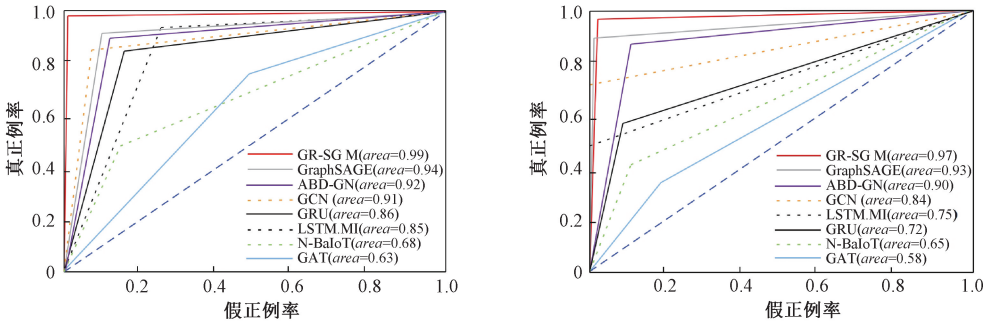
从表 1 看出，GR-SGM 在 ISCX2014 僵尸网络数据集上的准确率、召回率、精确率和 *F*1 都到达了最优。在 CICIDS2017 僵尸网络数据集上训练时大多数模型的准确率都偏高，而精确度和 *F*1 都偏低，这是由其僵尸网络样本数量远远小于正常样本数量，数据样本不平衡问题所导致。尽管如此，GR-SGM



(a) ISCX2014僵尸网络数据集 (b) CICIDS 2017僵尸网络数据集

图 6 GR-SGM 模型在两个数据集上的 TSNE 图

Figure 6 TSNE plot of GR-SGM model on two datasets



(a) ISCX2014僵尸网络数据集　　(b) CICIDS 2017僵尸网络数据集

图 7　在两个数据集上各模型的 ROC 曲线图

Figure 7　ROC curves of GR-SGM model on each dataset

表 1　对比实验结果

Table 1　Comparing experimental results

模型	ISCX2014 僵尸网络数据集				CICIDS2017 僵尸网络数据集			
	ACC/%	*REC*/%	*PRE*/%	*F*1	*ACC*/%	*REC*/%	*PRE*/%	*F*1
LSTM. MI	95. 43	98. 59	95. 66	0. 970 6	97. 70	94. 68	91. 45	0. 935 0
N-BaIoT	93. 52	94. 49	94. 05	0. 942 7	94. 68	99. 07	86. 80	0. 923 8
GRU	94. 99	96. 04	95. 78	0. 959 1	99. 56	81. 77	81. 18	0. 814 7
GCN	96. 34	93. 81	97. 81	0. 957 5	98. 86	98. 23	50. 57	0. 668 7
ABD-GN	98. 87	98. 09	97. 76	0. 980 2	99. 10	99. 20	91. 00	0. 947 0
GAT	81. 26	89. 88	84. 85	0. 867 8	98. 33	62. 61	26. 96	0. 373 4
GraphSAGE	96. 87	98. 09	97. 76	0. 980 4	99. 10	99. 23	80. 99	0. 890 1
GR-SGM	**99. 98**	**99. 90**	**99. 95**	**0. 999 4**	**99. 91**	**99. 72**	**99. 61**	**0. 996 5**

仍然有着最优的性能,准确率达到了最高的 99. 91%,召回率、精确度和 *F*1 等都远超于其他方法。GR-SGM 的性能优于传统深度学习方法以及传统图神经网络方法,其主要原因是 LSTM. MI、N-BaIoT 和 GRU 等传统深度学习检测方法主要关注单个数据流,未能充分利用数据流之间隐藏的结构信息,导致检测效果降低。相比之下,GR-SGM 不仅能学习单个节点特征,还能学习不同节点之间的结构关系特征。GCN、ABD-GN、GAT 和 GraphSAGE 等基于传统图神经网络的方法虽然利用了结构信息,但当伪装后僵尸网络流量特征和正常流量特征相似以及结构信息差异不大时,无法准确区分,导致检测精度不佳。而 GR-SGM 基于子图,能更加精确地捕捉局部特征,准确区分伪装后僵尸网络节点和正常流量节点。

3. 4　消融实验

为了验证本文模型中各个组件的有效性和合理性,在 ISCX2014 僵尸网络数据集和 CICIDS2017 僵尸网络数据集上对预检测模块和僵尸网络子图模块进行消融实验,结果如表 2 所示,其中,No-Pre-detection 中去除预检测模块,其他模型的组成部分保持不变;No-Subgraph 中去除僵尸网络子图模块,只对原始流量图与重构图进行僵尸网络检测。

从表 2 可以看出,No-Pre-detection 在 ISCX2014 僵尸网络数据集和 CICIDS2017 僵尸网络数据集上的各项评估指标都有所下降,这证明了预检测模块的重要性。对于一些僵尸网络特征显著的节点,它们的嵌入和重建特征很可能因为它们的属性或结构异常而被扭曲,因此需要预检测来修正,防止出现漏检的情况。No-Subgraph 在两个数据集上的各项评估指标都有所下降,这说明了僵尸网络子图模块的重要性。由于伪装后的僵尸网络节点特征与正常流量节点特征极为相似,因此需要子图的辅助,子图可

表 2　消融实验结果

Table 2　Results of ablation experiments

模型	ISCX2014 僵尸网络数据集				CICIDS2017 僵尸网络数据集			
	ACC/%	*REC*/%	*PRE*/%	*F*1	*ACC*/%	*REC*/%	*PRE*/%	*F*1
No-Pre-detection	93. 80	86. 46	98. 93	0. 923 9	91. 21	90. 30	95. 01	0. 927 8
No-Subgraph	96. 21	94. 50	97. 31	0. 957 3	97. 39	96. 13	80. 35	0. 874 9
GR-SGM	**99. 98**	**99. 90**	**99. 95**	**0. 999 4**	**99. 91**	**99. 72**	**99. 61**	**0. 996 5**

以准确地发现伪装后的僵尸网络节点,通过节点特征和图的结构来综合检测,解决僵尸网络特征表现不明显的问题。

3.5 参数敏感性实验分析

本节分析了在计算最短路径子图划分阈值 Y 时参数 Q 对检测的影响。由于 Y 值的计算除了与 Q 有关外,还与节点数相关联。因此,本文使用不同节点数的数据集进行实验,来验证不同情况下 Q 的大小对模型有效性的影响。在 ISCX2014 僵尸网络数据集上分别选取 1 000、5 000、10 000、20 000 个节点来进行实验。考虑到节点最短路径的计算复杂度,以及最小数据节点为 1 000,因此,将 Y 值从 50 更改到 250。根据图 8 的趋势总结出以下结论:当模型检测较小数据集(节点数<10 000)时,模型性能峰值出现在 $Y \leqslant 150$,即 $Q \geqslant 67$;当模型检测较大数据集(节点数≥10 000)时,模型性能峰值出现在 $Y \geqslant 200$,即 $Q \leqslant 50$。

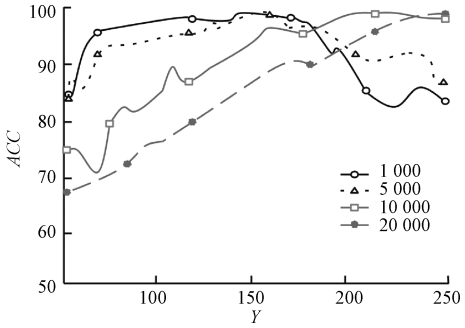


图 8 不同 Y 值情况下的检测准确率

Figure 8 Detection accuracy with different Y values

对于较小规模的数据集而言,较高的 Q 值意味着更少的子图划分,从而有效地保持每个子图的信息完整性。这是由于较小的网络数据集通常具有更紧密的连接模式,减少子图划分可以让每个子图仍然保持足够的信息密度和连接性;如果 Q 值设定过小,则可能导致子图过于分散,进而削弱对关键网络结构和通信模式的识别,降低检测的准确性。对于较大规模的数据集而言,较低的 Q 值意味着更多的子图划分,由于大规模的网络数据通常涉及更复杂的分布和通信模式,通过增加子图数量可以有效地降低单个子图中最短路径的计算复杂度,提高整体检测效率;如果 Q 值过高可能导致单个子图过于庞大,使得计算过于复杂和耗时,从而影响整体的检测效率。

4 结论

针对伪装后僵尸网络特征难以检测的问题,提出一种基于图重构和子图挖掘的僵尸网络检测方法(GR-SGM)。将网络流量数据构建成图,并设计图重构模块,通过对图进行编码和解码,获取新的向量表示,生成重构图;通过分析重构图,提取出僵尸网络子图,同时对原始网络流量图和重构图进行预检测,并将预检测结果和子图进行综合评估,以获取完整的僵尸网络信息。在 ISCX2014 僵尸网络数据集和 CICIDS2017 僵尸网络数据集上的实验结果表明,GR-SGM 优于现有的深度学习方法。在未来工作中,将引入时间关系等其他信息,进行更有效的特征学习,并进一步提高检测的速度和实现模型的轻量化。

参考文献:

[1] CHEN S C, CHEN Y R, TZENG W G. Effective botnet detection through neural networks on convolutional features[C]//2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/ 12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE). Piscataway: IEEE, 2018: 372-378.

[2] SRINIVASAN S, P D. Enhancing the security in cyberworld by detecting the botnets using ensemble classification based machine learning[J]. Measurement: Sensors, 2023, 25: 100624.

[3] 汪祖民, 王冬昊, 梁霞, 等. 基于 DBSCAN_GAN_XGBoost 的网络入侵检测方法[J]. 郑州大学学报(工学版), 2022, 43(3): 44-51.
WANG Z M, WANG D H, LIANG X, et al. Network intrusion detection method based on DBSCAN_GAN_XGBoost[J]. Journal of Zhengzhou University (Engineering Science), 2022, 43(3): 44-51.

[4] HAQ M A. DBoTPM: a deep neural network-based botnet prediction model[J]. Electronics, 2023, 12(5): 1159.

[5] ZHAO J, LIU X D, YAN Q B, et al. Multi-attributed heterogeneous graph convolutional network for bot detection[J]. Information Sciences, 2020, 537: 380-393.

[6] JOSHI H P, DUTTA R. A reinforcement approach for detecting P2P botnet communities in dynamic communication graphs[C]//ICC 2022-IEEE International Conference on Communications. Piscataway: IEEE, 2022: 56-61.

[7] ALSENTZER E, FINLAYSON S G, LI M M, et al. Subgraph neural networks[C]//Proceedings of the 34th International Conference on Neural Information Processing Systems. New York: ACM, 2020: 8017-8029.

[8] LI J X, SUN Q Y, PENG H, et al. Adaptive subgraph neural network with reinforced critical structure mining [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(7): 8063-8080.

[9] ZHANG Z, ZHAO L. Unsupervised deep subgraph anomaly detection[C]//2022 IEEE International Conference on

Data Mining (ICDM). Piscataway: IEEE, 2023: 753-762.

[10] VELASCO-MATA J, GONZÁLEZ-CASTRO V, FIDALGO E, et al. Real-time botnet detection on large network bandwidths using machine learning[J]. Scientific Reports, 2023, 13(1): 1-10.

[11] SHAHHOSSEINI M, MASHAYEKHI H, REZVANI M. A deep learning approach for botnet detection using raw network traffic data[J]. Journal of Network and Systems Management, 2022, 30(3): 1-23.

[12] TULASI RATNAKAR P, UDAY VISHAL N, SAI SIDDHARTH P, et al. Detection of IoT botnet using recurrent neural network[C]// Intelligent Data Communication Technologies and Internet of Things. Cham: Springer, 2022: 869-884.

[13] ZHOU J W, XU Z Y, RUSH A M, et al. Automating botnet detection with graph neural networks[EB/OL]. (2022-03-13)[2024-02-10]. https://arxiv.org/abs/2003.06344.

[14] BEIGI E B, JAZI H H, STAKHANOVA N, et al. Towards effective feature selection in machine learning-based botnet detection approaches[C]//2014 IEEE Conference on Communications and Network Security. Piscataway: IEEE, 2014: 247-255.

[15] ENGELEN G, RIMMER V, JOOSEN W. Troubleshooting an intrusion detection dataset: the CICIDS2017 case study[C]//2021 IEEE Security and Privacy Workshops (SPW). Piscataway: IEEE, 2021: 7-12.

[16] TRAN D, MAC H, TONG V, et al. A LSTM based framework for handling multiclass imbalance in DGA botnet detection[J]. Neurocomputing, 2018, 275: 2401-2413.

[17] MEIDAN Y, BOHADANA M, MATHOV Y, et al. N-BaIoT—network-based detection of IoT botnet attacks using deep autoencoders[J]. IEEE Pervasive Computing, 2018, 17(3): 12-22.

[18] CHO K, VAN MERRIENBOER B, GULCEHRE C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[EB/OL]. (2014-09-03)[2024-02-10]. https://arxiv.org/abs/1406.1078.

[19] CARPENTER J, LAYNE J, SERRA E, et al. Detecting botnet nodes via structural node representation learning[C]//2021 IEEE International Conference on Big Data (Big Data). Piscataway: IEEE, 2022: 5357-5364.

[20] VELIČKOVIĆ P, CUCURULL G, CASANOVA A, et al. Graph attention networks[EB/OL]. (2018-02-04)[2024-02-10]. https://arxiv.org/abs/1710.10903.

[21] HAMILTON W L, YING R, LESKOVEC J. Inductive representation learning on large graphs[EB/OL]. (2018-09-10)[2024-02-10]. https://arxiv.org/abs/1706.02216.

Botnet Detection Method Based on Graph Reconstruction and Subgraph Mining

JING Yongjun[1,2], WU Hui[2], CHEN Xu[2], SONG Jifei[3]

(1. School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, China; 2. School of Computer Science and Engineering, North Minzu University, Yinchuan 750021, China; 3. National (Zhongwei) New-type Internet Exchange Point, Zhongwei 755000, China)

Abstract: Aiming at the problem that disguised botnet hosts are difficult to detect, a botnet detection method based on graph reconstruction and subgraph mining (GR-SGM) was proposed. Firstly, network data was converted into graph data which was reconstructed to enhance the host node feature representation. Then, based on the topological structure, node characteristics, and position changes in the reconstructed graph, a botnet subgraph scoring function was designed. In this way, the camouflaged features were captured, the botnet subgraph was extracted, and the original and reconstructed graphs were pre-detected to improve detection accuracy and efficiency reducing reconstruction errors. Finally, the pre-detection results and botnet subgraphs were comprehensively scored to obtain complete botnet information. Experimental results on the ISCX2014 botnet dataset and CICIDS2017 botnet dataset showed that the detection accuracy of GR-SGM was 99.98% and 99.91%, respectively, and the $F1$ reached 99.94% and 99.65%, respectively. Compared with other botnet detection models, GR-SGM could identify botnet nodes more efficiently and accurately, while having a lower false alarm rate.

Keywords: botnet; subgraph mining; graph reconstruction; cybersecurity; pre-detection