软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2024,35(10):4837−4858 [doi: 10.13328/j.cnki.jos.006964]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于图对比学习的恶意域名检测方法*

张 震[1], 张三峰[1,2], 杨 望[1,2]

[1](东南大学 网络空间安全学院, 江苏 南京 211189)

[2](教育部计算机网络和信息集成重点实验室 (东南大学), 江苏 南京 211189)

通信作者: 张三峰, E-mail: sfzhang@seu.edu.cn

摘 要: 域名是实施网络犯罪行为的重要环节, 现有的恶意域名检测方法一方面难以利用丰富的拓扑和属性信息, 另一方面需要大量的标签数据, 检测效果受限而成本较高. 针对该问题, 提出一种基于图对比学习的恶意域名检测方法, 以域名和 IP 地址作为异构图的两类节点并根据其属性建立对应节点的特征矩阵, 依据域名之间的包含关系、相似度度量以及域名和 IP 地址之间对应关系构建 3 种元路径; 在预训练阶段, 使用基于非对称编码器的对比学习模型, 避免图数据增强操作对图结构和语义的破坏, 也降低对计算资源的需求; 使用归纳式的图神经网络图编码器 HeteroSAGE 和 HeteroGAT, 采用以节点为中心的小批量训练模式来挖掘目标节点和邻居节点的聚合关系, 避免直推式图神经网络在动态场景下适用性较差的问题; 下游分类检测任务则对比使用了逻辑回归、随机森林等算法. 在公开数据上的实验结果表明检测性能相比已有工作提高 2–6 个百分点.

关键词: 恶意域名检测; 属性异构图; 图神经网络; 非对称编码; 自监督学习

中图法分类号: TP393

中文引用格式: 张震, 张三峰, 杨望. 基于图对比学习的恶意域名检测方法. 软件学报, 2024, 35(10): 4837–4858. http://www.jos.org.cn/1000-9825/6964.htm

英文引用格式: Zhang Z, Zhang SF, Yang W. Malicious Domain Name Detection Method Based on Graph Contrastive Learning. Ruan Jian Xue Bao/Journal of Software, 2024, 35(10): 4837–4858 (in Chinese). http://www.jos.org.cn/1000-9825/6964.htm

Malicious Domain Name Detection Method Based on Graph Contrastive Learning

ZHANG Zhen[1], ZHANG San-Feng[1,2], YANG Wang[1,2]

[1](School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China)

[2](Key Laboratory of Computer Network and Information Integration of Ministry of Education (Southeast University), Nanjing 211189, China)

Abstract: The domain name plays an important role in cybercrimes. Existing malicious domain name detection methods are not only difficult to use with rich topology and attribute information but also require a large amount of label data, resulting in limited detection effects and high costs. To address this problem, this study proposes a malicious domain name detection method based on graph contrastive learning. The domain name and IP address are taken as two types of nodes in a heterogeneous graph, and the feature matrix of corresponding nodes is established according to their attributes. Three types of meta paths are constructed based on the inclusion relationship between domain names, the measure of similarity, and the correspondence between domain names and IP addresses. In the pre-training stage, the contrast learning model based on the asymmetric encoder is applied to avoid the damage to graph structure and semantics caused by graph data augmentation operation and reduce the demand for computing resources. By using the inductive graph neural network graph encoders HeteroSAGE and HeteroGAT, a node-centric mini-batch training strategy is adopted to explore the aggregation relationship between the target node and its neighbor nodes, which solves the problem of poor applicability of the transductive graph neural networks in dynamic scenarios. The downstream classification detection task contrastively utilizes logistic regression and

* 基金项目: 国家自然科学基金 (62272100)
收稿时间: 2022-09-06; 修改时间: 2023-01-17, 2023-04-03; 采用时间: 2023-05-12; jos 在线出版时间: 2023-09-13
CNKI 网络首发时间: 2023-09-14

random forest algorithms. Experimental results on publicly available data sets show that detection performance is improved by two to six percentage points compared with that of related works.

Key words: malicious domain name detection; attribute heterogeneous graph; graph neural network (GNN); asymmetric coding; self-supervised learning

1 引 言

域名系统 (domain name system, DNS) 提供域名解析功能, 是互联网的关键基础设施[1]. 攻击者的恶意行为常常伴随着恶意域名的生成和使用, 譬如构造恶意域名诱使用户点击钓鱼网站, 或使用域名生成技术控制僵尸网络发起 DDoS 攻击[2]. 攻击者还可以利用 DGA[3]、Domain-flux[4]、Fast-flux[5]等技术, 隐匿 DNS 相关的恶意行为, 以绕过防火墙和入侵检测系统等传统防护手段. 钓鱼、赌博诈骗等网络黑产已经成为最活跃的网络犯罪行为, 恶意域名是实施这类行为的重要环节, 因此, 如何准确高效地对恶意域名进行检测已成为网络安全领域亟待解决的重要课题.

早期, 由于网络本身的规模较小、攻击的手段较为落后, 恶意域名主要通过黑白名单对照的方式进行检测[6]. 但随着 Domain-flux 以及 Fast-flux 等规避检测技术的发展, 攻击者不但可以每天产生数以万计的恶意域名, 而且可以改变域名和 IP 地址之间的映射模式, 黑名单技术由于无法及时覆盖所有恶意域名, 已不能够满足现实场景中复杂多变的检测需求. 检测技术研究转向基于特征提取、统计分析的分类方法, 基于机器学习的分类算法对提取 DNS 流量、日志、字符等特征并进行统计分析, 构建标记数据, 学习、训练恶意域名分类器[7,8]. 这种方法一方面需要专家知识提取、筛选特征, 另一方面需要大量的标签数据, 实践起来成本较高. 此外, 以上两种方法都只关注域名的个体特征, 忽略了不同域名之间以及域名和主机之间复杂的拓扑关系, 不仅分类精度受限, 也更容易受各种规避技术的影响.

基于图的检测方法可以弥补上述方法的不足. 该类方法从 DNS 流量和日志等数据中提取域名和 IP 地址之间的映射关系来构建“域名-IP 地址”异构图, 基于元路径建立节点之间的关联, 并通过图推理的方式发现二部图中的恶意域名节点[9−11]. 攻击者可以轻易伪造域名, 但篡改域名之间、域名和 IP 地址之间关联关系的成本更高, 因此基于图的检测方法理论上具有更强的检测能力. 但是, 现有的基于图推理的检测算法大多只使用节点的结构信息, 忽略了节点的属性信息, 并需要将二部异构图转换为同构图, 这种做法有很大的局限性: 首先, 强制对齐不同类型的节点将导致信息丢失; 其次, 阻碍了对域名系统更深层次的信息挖掘. 针对上述缺陷, 基于异构图神经网络的检测方法通过学习图的拓扑信息和属性信息, 完成对不同类型节点的嵌入, 在有监督场景下取得了更优的检测性能. 然而, 充分注释节点数据成本高昂且实现困难, 因此该类方法并不适用于半监督或者无监督的现实场景.

针对以上问题, 本文提出一种基于异构属性图对比学习的恶意域名检测模型 MD-GCL, 通过挖掘数据内在信息作为监督信号, 缓解了缺乏标签数据的问题并提高了模型的可迁移性. MD-GCL 将域名和 IP 地址看作属性异构图中两种不同类型的节点, 将域名和 IP 地址的属性视作图中对应节点的特征矩阵, 并依据域名之间的包含关系、相似度度量以及域名和 IP 地址之间解析与被解析关系构建了 3 种不同的元路径. 在此基础上, 采用无监督表征学习的训练策略对 DNS 属性异构图数据进行学习和训练, 先使用预先设计好的辅助任务对编码器进行预训练, 之后冻结编码器的参数, 执行下游分类任务. 在预训练阶段, 首先将属性异构图数据送入两个不同的图编码器进行嵌入以得到两种不同的表示, 之后利用损失函数迫使它们之间的互相关矩阵趋近于单位矩阵, 使得从相同数据样本提取出的多个特征表示高度相似, 并最小化特征向量各个分量之间的冗余. 此外, MD-GCL 使用的两个图编码器 HeteroSAGE 和 HeteroGAT 均属于归纳式图神经网络, 采用以节点为中心的小批量训练模式来挖掘目标节点和邻居节点的聚合关系, 有效地解决了 GCN[12]等直推式图神经网络不适用于动态场景的问题. 在下游任务阶段, 使用逻辑回归、随机森林等算法对预训练阶段学习到的表征进行分类. 实验结果表明, MD-GCL 模型在大规模公开数据集 DNS[13]和 mDNS[13]上的恶意域名检测性能相比已知的公开文献分别提高了 2–6 个百分点.

本文第 2 节介绍了恶意域名检测、异构图神经网络以及图对比学习的相关工作. 第 3 节提供了基于属性异构

图的恶意域名检测的相关定义. 第 4 节则阐述了 MD-GCL 模型的具体细节. 第 5 节对实验的结果进行了对比、分析和评估. 最后, 总结全文并探讨了下一步的工作方向.

2 相关工作

2.1 恶意域名检测

目前, 恶意域名检测研究按据数据处理方式可以分为以下 3 类: 基于特征分析的方法、基于图推理的方法以及基于异构图神经网络的方法. 基于特征分析的方法依赖于特征工程. 例如, Antonakakis 等人[14]提出了动态域名评分系统 Notos, 旨在通过分析域名的文本特征和空间分布特征来建立域名分类模型. 但这种方法需要大量的历史数据进行训练, 且无法识别新出现的恶意域名. 针对该问题, Bilge 等人[15]则设计了 Exposure 系统, 它使用决策树算法, 根据从 DNS 流量中提取的 15 个特征 (如时间特征和 DNS 响应特征) 对恶意域名进行分类. 这两种基于特征的检测算法首先要依赖于特征工程提取的有效域名特征, 但随着攻击手段的不断进步, 过时的特征不仅可能导致模型鲁棒性退化, 也因无法充分利用域名解析场景中丰富的拓扑和属性信息, 导致检测性能有限. 因此, 研究人员开始关注域名之间的关联信息. 彭成维等人[16]提出一种基于域名请求伴生关系的恶意域名检测方法 CoDetector, Sun 等人[10]则提出基于异质信息网络的 HinDom, 在恶意域名场景中定义了 3 种不同类型的实体和 6 种不同的元路径关系, 并基于元路径图推理对域名进行分类, 实验结果表明, 该模型在小样本检测场景中取得了良好的检测效果. 以上两种基于图推理的方法仅依据关系密切的域名通常属于同一类的假设, 忽略了图中丰富的属性信息和攻击者伪造关联的可能, 而基于异构图神经网络的检测方法可以挖掘更深层次的信息. Deepdom[11]将可扩展的异构图卷积神经网络模型与异构信息网络相结合, 通过学习节点特征和拓扑关系来完成域名分类. Zhang 等人[17]则在半监督场景下提出了一种基于属性异构图神经网络的恶意域名检测模型, 通过节点类型感知的特征转换机制、边缘类型感知的聚合机制同时融合节点属性和拓扑信息, 从而完成 DNS 异构图的推理和分类. 上述方法虽进一步提高了恶意域名的检测精度, 但仍无法摆脱对标签数据的依赖, 因此难以应用于现实场景.

2.2 异构网络嵌入

异构网络可以看作是一种图结构. 图嵌入旨在从高维稀疏的图数据中学习到低维的节点表示, 通过训练得到嵌入表示, 然后用于下游任务, 是基于图的分类检测方法的重要环节[18]. 在恶意域名检测场景下, 异构网络的表示学习需要保留异构结构和语义信息以用于下游任务. Shi 等人[19]首先给出了异构信息网络 (heterogeneous information network, HIN) 的明确定义, Li 等人[20]率先将节点的属性信息引入到异构信息网络中, 并提出一种用于嵌入静态属性异构信息网络 (AHIN) 的离线模型. HIN2Vec[21]则提出一种基于元路径随机游走的异构网络嵌入方法, 其联合执行多个关系预测任务以共同学习节点和元路径的嵌入表示. 近几年, 随着 GCN[12]、GAT[22]等图神经网络模型的快速发展, 一些异构图神经网络也相继被提出并用于异构网络嵌入: RGCN[23]尝试将 GCN 的框架扩展至关系数据建模; HAN[24]则设计了一种分层注意机制来捕捉异构网络的图结构和语义信息; HetGNN[25]认为现有工作方法不能较好地整合异构网络中的丰富信息, 提出异构邻居采样策略、异构内容编码策略和异构邻居聚合策略, 在节点分类、聚类以及链路预测等任务上取得了更好的实验结果. Hu 等人[26]则认为现有的大多数工作依赖于精心设计的元路径, 并且忽略了异构网络的动态特性, 通过引入基于异构图的 Transformer 机制和相对时间编码技术来解决上述问题. 上述方法通过区分不同的节点类型和边类型, 在一定程度上解决了异构网络的嵌入问题, 但模型的泛化性和可扩展性不强, 在恶意域名检测场景中的性能表现均较差. 为解决上述问题, 本文基于邻居采样的聚合机制和图注意力机制, 分别对各个元路径进行编码以充分捕捉异构语义信息, 并进一步引入对比学习机制来挖掘域名检测场景中更深层次的信息.

2.3 图对比学习

大多数异构网络嵌入方法具有以下两点局限性: 首先, 过度依赖标记数据; 其次, 在有监督场景下学习到的知识可迁移性不强. 针对上述问题, 图对比学习使用辅助任务指导模型从大规模无监督数据中学习可迁移的知识, 并将学到的知识应用到具有特定监督信号的下游任务中, 现已被广泛应用于社交网络、推荐系统等领域. DGI[27]最

早将对比学习的思想引入到图神经网络模型中, 其核心思想是通过最大化节点表征和全局表征之间的互信息来训练编码器. 不同于 DGI 的跨尺度对比, GRACE[28]则提出了一个在节点层级进行同尺度对比的目标函数, 在通过数据增强方法生成的增强视图上构建正负样本来进行对比学习, 并在 Cora、PPI 等多个数据集上取得了优于 DGI 的性能. 除了对比任务的设计方法, 数据增强策略的选择也至关重要. You 等人[29]在 GraphCL 中设计了 4 种不同的数据增强策略, 并对不同场景以及不同数据增强策略下的模型性能进行了评估. Zhu 等人[30]则通过节点和边的中心性度量来分别识别重要的节点属性和拓扑连接, 提出了一种可以进行自适应数据增强的图对比学习方法. 然而, 上述基于负例的对比学习模型往往依赖于设计良好的数据增强策略和高质量的负例样本, 对计算机的算力和内存要求较高. 针对该问题, BGRL[31]受文献 [32] 启发, 提出一种基于非对称孪生网络的图对比学习方法, 分别使用在线编码器和目标编码器对输入数据进行编码, 并通过约束编码表示间的均方误差来训练目标编码器. BGRL 不再去关注正负样本之间是否具有不同的表征, 而是仅仅使得正样本彼此之间的表征相似, 实验评估表明 BGRL 性能已超越上述的 DGI、GRACE 等模型. 类似地, GBT[33]受文献 [34,35] 启发, 利用损失函数迫使输入数据的两个增广视图的互相关矩阵接近于单位矩阵, 使得不同维度的特征尽可能地表示不同的信息, 因此能够对不同类别的数据进行良好的区分.

受上述工作启发, 本文设计了一种基于非对称异构编码器架构的对比学习模型 MD-GCL. 和相关工作对比, MD-GCL 有如下优势.

(1) 使用结构和语义更丰富的异构网络表示学习, 提高了检测能力.

(2) 使用图对比学习降低了对标签数据的依赖.

(3) 在对比学习方法上, 使用非对称编码器, 消除了对负样本的显式需求, 避免了可能对破坏图数据语义的数据增强工作, 也降低了对计算资源的需求.

(4) 使用的编码器为归纳式图神经网络, 本质上适用于新节点的分类.

3 问题场景建模

3.1 设计目标和问题建模

域名常被攻击者用于僵尸网络生成、黑灰产传播等违法犯罪活动, 准确识别恶意域名有助于及时遏制犯罪行为、减少社会经济损失. 本文旨在通过网络 DNS 日志数据构建由域名和 IP 地址所组成的异构信息网络并利用安全情报开源平台对数据进行扩充, 之后使用属性异构图对比学习方法学习节点嵌入, 进而完成对域名标签的分类预测. 根据识别到的恶意域名, 安全从业人员可给出告警信息并进一步采取防御策略. 本文设计目标如下: (1) 利用异构图丰富的节点和边类型提升恶意域名检测性能, 并分析其相对同构图建模方法的优势; (2) 利用无监督的对比学习方法改进异构图表征学习, 在少量训练批次下生成高质量嵌入, 并通过 TSNE 图和基线模型对比进行效果验证; (3) 利用预训练好的模型对全图数据进行嵌入, 并将嵌入结果输入至分类器中预测域名节点标签, 验证其相对基线模型的优越检测性能.

基于图对比学习的恶意域名检测场景建模主要包含节点抽取、关系抽取、特征提取以及标签获取四大步骤, 依次详细阐述如下: 域名和 IP 地址被视为异构信息网络的基础节点; 基于 DNS 日志数据抽取域名和 IP 地址间的解析关系、基于安全情报平台开源数据抽取域名间共享同一顶点域的子域名关系、基于域名的文本嵌入计算相似度抽取域名间的相似关系; 抽取域名的统计特征 (如域名长度、深度等)、语言特征 (如是否包含数字、重复字符比例等) 以及 WHOIS 信息作为域名节点的属性, 抽取 IP 地址的 ASN 号、地理位置等信息作为 IP 地址节点的属性; 基于 Alexa top 1m 列表和安全情报平台开放接口获取节点标签.

3.2 相关定义

● 属性异构图

当图中节点类型数和边类型数之和大于 2 时, 该图即为异构图. 带属性的异构图可以定义为 $G=\{\mathcal{V},\mathcal{E},\mathcal{T},\mathcal{X}\}$, 其中 $\mathcal{V}$ 和 $\mathcal{E}$ 分别表示节点集合和边集合, $\mathcal{T}$ 为类型映射函数集合, 节点到节点类型的映射函数记为 $\phi(\mathcal{V}):\mathcal{V}\to\mathcal{T}_{\mathcal{V}}$,

边到边类型的映射函数记为 $\varphi(\mathcal{E}):\mathcal{E}\to\mathcal{T}_{\mathcal{E}}$, 特征矩阵 $\mathcal{X}\in R^{N\times M}$, M 代表节点的特征维数, $x_i\in R^M$ 代表单个节点 v_i 的特征. 此外, 每个节点都有一个属于自己的标签类别 $y_i\in Y$, $Y=\{y_i\mid y_i=0,1,2,\ldots,c-1\}$, c 代表图中节点标签的类别数.

以域名系统为例, IP 地址、域名可分别视为异构图中的节点类型 v_{ip}、v_{domain}, IP 地址所属的子网及自治系统等信息作为 IP 地址的特征矩阵 x_{ip}, 域名的全限定域名的域名级数以及信息熵等统计信息可以作为域名的特征矩阵 x_{domain}. 此外, 异构图中的每个节点都有反映自身类别的标签属性 y, 对于 IP 地址, 我们只标记其类型 $y_{\text{ip}}\equiv\mathbb{C}$, 对于域名, 我们将其类型细分为恶意域名和正常域名两类, 即 $y_{\text{domain}}\in\mathbb{R},\mathbb{Q}$, 上述公式中的 $\mathbb{C},\mathbb{R},\mathbb{Q}$ 均为常数.

此外, 异构图中的结构通常是语义依赖的, 对不同关系类型所构成的局部结构, 引入元路径的概念进行描述: $m=\mathcal{T}_{\mathcal{V}}^1\xrightarrow{\mathcal{T}_{\mathcal{E}}^1}\mathcal{T}_{\mathcal{V}}^2\xrightarrow{\mathcal{T}_{\mathcal{E}}^2}\ldots\xrightarrow{\mathcal{T}_{\mathcal{E}}^l}\mathcal{T}_{\mathcal{V}}^{l+1}$, 其中, $\mathcal{T}_{\mathcal{V}}^i$ 表示不同的节点类型, $\mathcal{T}_{\mathcal{E}}^i$ 表示不同的边类型. 由不同元路径所构成的元路径集合定义为: $\mathcal{M}=\{m_i\mid i\in N_{\text{meta}}\}$, N_{meta} 代表异构图中元路径的数量. 此时, 属性异构图的形式化定义可重写为 $G=(\mathcal{V},\mathcal{E},\mathcal{T},\mathcal{X},\mathcal{M})$.

同样以域名系统为例, 域名节点和 IP 节点间存在着 3 种类型的关系: 域名和 IP 地址间的解析关系、域名间共享同一顶点域的子域名关系、域名字符间的相似度关系. 其中, 相似度关系是指使用 N-Gram 算法和 TF-IDF 算法对域名进行文本嵌入并基于嵌入间的余弦相似度、预先设置的相似度阈值建立起来的边关系. 基于域名节点和 IP 节点间的 3 种不同关系, 3 种不同元路径的符号定义如下.

(1) $m_1=V_{\text{domain}}\xrightarrow{\text{similar}}V_{\text{domain}}$;

(2) $m_2=V_{\text{domain}}\xrightarrow{\text{subdomain}}V_{\text{domain}}$;

(3) $m_3=V_{\text{domain}}\xrightarrow{\text{resolve}}V_{\text{ip}}\xrightarrow{\text{resolve}}V_{\text{domain}}$.

其中, m_1 表示通过相似度来度量的不同域名之间的关系, m_2 表示域名之间可能存在的从属关系, m_3 则表示域名和 IP 地址之间解析和被解析的关系. 上述 3 种元路径构成元路径集合 $\mathcal{M}=\{m_i\mid i=0,1,2\}$. 一个典型的“域名-IP”属性异构图如图 1 所示.

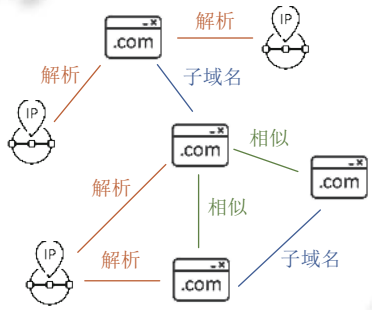

图 1 “域名-IP”属性异构图

● 图编码器

图编码器通常基于图神经网络进行构建, 其旨在将拓扑图中的高维稀疏数据转换为低维向量表示, 同时尽可能地保留图的结构信息和节点语义信息. 在涵盖多种节点类型和边类型的属性异构图中, 使用图编码器对不同类型的节点进行嵌入, 有助于属性异构图数据的深度发掘.

图编码器一般通过聚合邻居节点的表示来更新自我节点表示, 公式表述为: $H^{(l)}=\delta(H^{(l-1)},G)$, 其中 δ 代表邻域聚合函数, $H^{(l)}$ 指的是节点在模型第 l ($l\geqslant 1$) 层的表征, 但特别地, $H^{(0)}$ 表示图中节点初始的特征表示. 在向量层面, 上述聚合过程在可进一步阐述为: $h_u^{(l)}=f_{\text{combine}}\left(h_u^{(l-1)},f_{\text{aggregate}}\left(\{h_i^{(l-1)}\mid i\in N_u\}\right)\right)$. 为得到节点 u 在第 l 层的表征, GNN 编码器首先在第 $l-1$ 层对节点 u 的所有邻居节点的表征进行聚合, 然后将聚合到的表征和节点 u 自身在第 $l-1$ 层的表征组合到一起. 在计算得到节点 u 全部 L 层的表征之后, GNN 编码器通过一个读出函数来获得最终的表征, 即 $h_u=f_{\text{readout}}(h_u^{(l-1)}\mid l=[0,1,2,\ldots,L])$.

在域名系统中, 对由域名节点和 IP 节点构成的异构图 $G=(\mathcal{V},\mathcal{E},\mathcal{T},\mathcal{X},\mathcal{M})$, 本文定义两种不同的异构图编码

器对 G 进行嵌入, 公式表述为: $h_1 = g_1(G)$, $h_2 = g_2(G)$, 其中 $g_1(\cdot)$ 和 $g_2(\cdot)$ 指代实现上述图编码器消息传递和聚合过程的函数. 在得到上述表征之后, 计算 h_1 和 h_2 的互相关矩阵 $C = mm(h_1 h_2^{\mathrm{T}})$, 然后使用损失函数 $\mathcal{L}$ 来不断优化模型参数 θ, 使得 C 和单位矩阵 E 尽可能地相似, 即 $\theta = \arg\min_{\theta} \mathcal{L}(C, E)$.

3.3 系统架构

本文基于图对比学习的恶意域名检测模型的系统架构如图 2 所示, 主要包含以下 3 个模块: (1) 问题场景建模. 如第 3.1 节所述, 可基于 DNS 日志、安全情报分析平台开源数据以及 WHOIS、PDNS 记录构造“域名-IP”二部异构网络, 并为域名节点和 IP 地址节点抽取特征和获取标签. (2) 元路径抽取. 在构造的异构网络基础上, 定义 3 种不同的元路径关系来挖掘图中域名和 IP 地址节点间的高级语义依赖关系. (3) 基于非对称异构图对比学习模型的恶意域名检测. 将图数据输入到本文 MD-GCL 模型中进行预训练, 并将输出的节点嵌入输入到分类预测器中完成对图中域名节点的标签预测, 安全从业人员可根据判定结果给出早期预警并进一步采取主动防御策略.

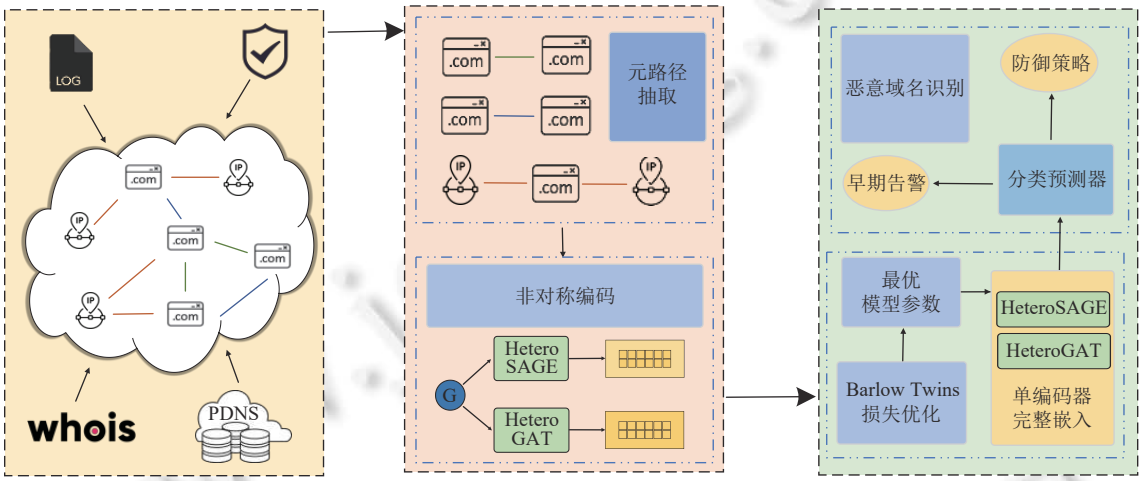

图 2 系统模型架构示意图

4 模型设计

监督学习的数据标签成本高、过渡拟合泛化能力差、在面对标签相关的对抗攻击时鲁棒性较弱, 自监督学习在解决这些问题方面有明显优势, 因此在恶意域名检测方面有更好的适用性. 另外, 已有的基于负例的对比学习模型的复杂度和训练成本较高, 用于图数据处理时面临较大的挑战, 同时基于数据增强的对比方法还会导致图数据语义损失. 针对这些问题, 本文设计了基于异构编码器的属性异构图自监督对比学习模型 MD-GCL, 用于恶意域名检测. 图 3 展示了该模型对比预训练阶段的工作流程.

如图 3 所示, 图中不同颜色形状的节点分别代表域名节点和 IP 节点, 不同颜色的边分别代表 3 种不同类型的边关系. MD-GCL 首先将输入的图数据发送到两个不同的异构图编码器 HeteroSAGE 和 HeteroGAT 中, 使用双层的 GraphSAGE 和 GAT 对抽取出的不同元路径视图分别进行编码并进行聚合, 在得到相对应的嵌入表征 h_1 和 h_2 后, 利用损失函数迫使 h_1 和 h_2 的互相关矩阵趋近于单位矩阵, 即对编码器学习到的表征的各个分量做解耦, 达到区分开不同节点类型的目的.

与基于负例的对比学习模型相比, MD-GCL 不再显式地构造正负样本, 而是在无监督场景下使用不同的编码器编码输入数据. 由于图嵌入旨在低维空间表达拓扑结构、节点属性等语义信息, 对相同的图数据, 不同图编码器所捕获的语义信息应当相似, 因此, MD-GCL 通过训练使得相同数据的不同编码表示相互接近, 并减少冗余表示, 使编码器能够学习到数据的深层特征, 从而更好地辅助于下游任务. MD-GCL 模型主要由非对称编码器、对比损失函数设计和下游分类器 3 部分组成, 本节将依次对这几个部分进行详细阐述.

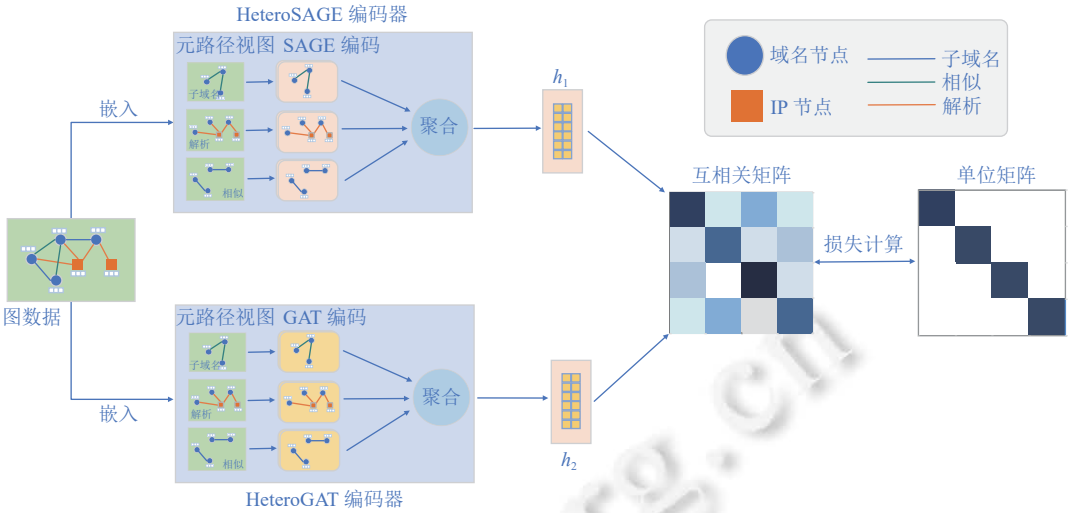

图 3 MD-GCL 模型框架示意图

4.1 非对称编码

对比学习是一种表示学习方法, 通常采用数据增强的策略, 即通过语义不变的变换得到同一原始样本的不同副本, 然后训练神经网络模型参数使得输入的不同副本输出相似的表示, 以此获得表示学习模型. 数据增强的关键是不能损坏数据的原始语义信息, 在计算机视觉和自然语言处理领域, 图像和语句的语义往往是显式的, 对于图像旋转或者语句同义词替换等数据增强方式而言, 研究人员可以很明显地得到原生数据的语义没有被破坏的结论. 但是, 对图数据来说, 由于其抽象性, 所蕴含的语义往往是隐式的、不容易被人们直观感受到的, 现有的图对比学习数据增强方法, 比如在拓扑层次或者特征层次上进行随机扰动的方式, 往往会破坏图数据的语义. 例如, 一些原本重要的数据可能会被删除, 而原本不重要的数据会被留下, 这导致对比学习无法学习到图数据的本质特征, 因此下游任务的性能变差.

为解决上述问题, MD-GCL 模型使用了基于非对称异构图编码器的对比学习策略, 采用 HeteroSAGE 和 HeteroGAT 对输入的图数据进行嵌入. 由于两个图编码器的消息传递机制和聚合机制略有不同, 因此相同的图数据最终将得到不同的表示. 此外, 根据文献 [36], 经典的同构 GNN 编码器的性能常被严重低估, 配置良好且输入恰当的图注意力网络在各种场景下的性能表现要优于现有的异构图神经网络模型. 因此, MD-GCL 基于经典的同构图神经网络 GAT[22]、GraphSAGE[37]设计并进行改进, 以满足异构场景中不同类型节点和边的嵌入需求. 针对属性异构图数据, HeteroSAGE 和 HeteroGAT 为每种关系类型单独实现消息传递、聚合以及更新的功能, 即分别在图中相同和不同关系类型的连接上进行消息传递, 并不断更新各个节点类型的嵌入.

图 4 展示了具有两层架构的 HeteroSAGE 编码器, 阐明了基于域名节点和 IP 节点之间解析与被解析的元路径关系, 不同节点类型之间的消息传递过程. HeteroSAGE 编码器的编码过程可以分为以下 5 步.

(1) 域名节点初始表征 x_{domain} 和 IP 节点初始表征 x_{ip} 被分别输入到 SAGE 卷积层中进行嵌入.

(2) 基于元路径 $m_3 = V_{\text{domain}} \xrightarrow{\text{resolve}} V_{\text{ip}} \xrightarrow{\text{resolve}} V_{\text{domain}}$, 域名节点和 IP 节点的嵌入除在节点当前分支继续传递外, 也会被输送到对应的邻居节点分支服务于其邻域聚合.

(3) LeakyReLU 激活函数对 SAGE 卷积层学习到的域名节点表示和 IP 节点表示进行激活处理.

(4) 批规范化即 BatchNorm 层是神经网络两个隐藏层之间的网络层, 其对收到的激活表示进行规范化, 使得输出数据的各个维度满足于正态分布.

(5) 对于 BatchNorm 层的输出, 重复上述步骤 (1)–(3), 得到域名和 IP 节点的最终嵌入表示, 分别为 h_{domain} 和 h_{ip}.

$$h^l_{\mathcal{N}(v)} = f_{\text{aggregate}}(\{h^{l-1}_u \mid u \in \mathcal{N}(v)\}) \tag{1}$$

$$h_v^l = \beta\left(LeakyReLU\left(f_{\text{concat}}\left(h_v^{l-1}, h_{\mathcal{N}(v)}^l\right)\right)\right) \tag{2}$$

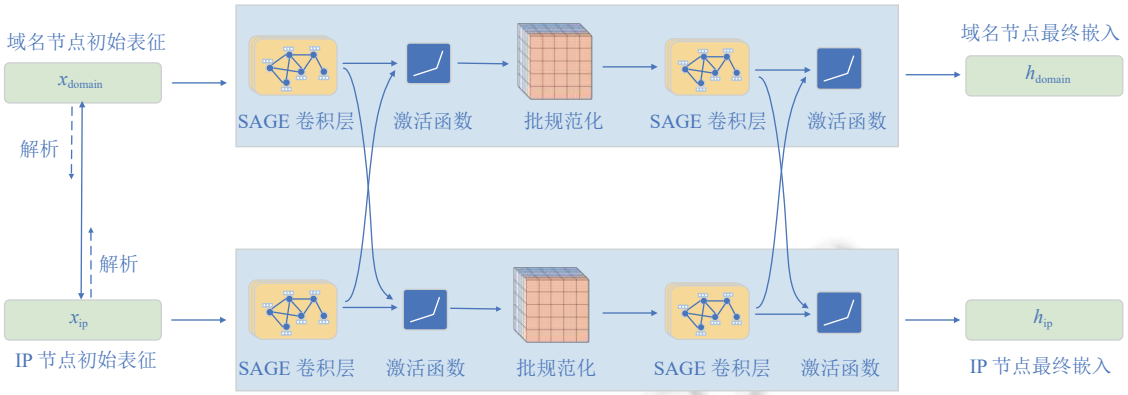

图 4 MD-GCL 非对称编码架构示意图

HeteroSAGE 编码器的编码过程的形式化定义如公式 (1) 和公式 (2) 所示, 其中 $h_v^{(l)}$ 指的是节点 v 在模型第 l 层的表征, β 指的是 BatchNorm 层实现批标准化的执行函数. 对于不同的元路径关系, 公式 (1) 首先对当前节点采样一定数量的邻居节点, 并将其嵌入进行聚合得到 $h_{\mathcal{N}(v)}^l$, 之后在公式 (2) 中, $h_{\mathcal{N}(v)}^l$ 和节点 v 在第 $l-1$ 层的表示 h_v^{l-1} 被拼接在一起, 经激活函数非线性变换和批标准化处理后, 产生节点 v 在第 l 层的表征 h_v^l. 值得注意的是, 在模型的最终层, HeteroGNN 只使用激活函数对表征进行激活而不再进行批标准化处理.

$$e_{vu} = LeakyReLU(a[(Wh_v \| Wh_u]) \tag{3}$$

$$\alpha_{uv} = \frac{\exp(e_{uv})}{\sum_{u\in\mathcal{N}(v)}\exp(e_{vu})} \tag{4}$$

HeteroGAT 与 HeteroSAGE 之间的差异主要表现在生成目标节点 v 的邻域聚合表示的不同机制上, 即 HeteroGAT 不再在聚合过程中对目标节点的邻居节点进行采样, 而是使用注意力机制. 对于在属性异构图中互为邻居节点的节点 v 和节点 u, HeteroGAT 首先使用公式 (3) 计算一个节点对于另外一个节点的重要程度, 之后在公式 (4) 中使用 Softmax 对当前目标节点 v 的所有一跳邻居节点的注意力权重进行归一化.

h_v 和 h_u 分别代表节点 v 和节点 u 的特征, W 代表 h_v 和 h_u 所共享的映射矩阵, $\|$ 代表拼接操作; $a(\cdot)$ 则代表将拼接后的节点高维特征映射为实数的映射函数, $LeakyReLU(\cdot)$ 代表激活函数. 在上述基础上, 根据计算好的注意力系数, HeteroGAT 对输入的邻居节点的表示进行聚合以产生节点 v 所有邻居节点的聚合表示 $h_{\mathcal{N}(v)}^l$, 如公式 (5) 所示:

$$h_{\mathcal{N}(v)}^l = f_{\text{aggregate}}\left(\alpha_{vu} h_u^{l-1} | u \in \mathcal{N}(v)\right) \tag{5}$$

4.2 对比损失

在训练过程中, MD-GCL 通过不断优化模型参数, 使不同嵌入表示间的互相关矩阵不断接近于恒等矩阵, 该过程可以分为嵌入标准化、互相关矩阵计算以及损失函数计算这 3 个部分. 在训练过程中, 模型首先对 HeteroSAGE 和 HeteroGAT 编码获得的表征 h_1 和 h_2 进行 z-score 标准化处理, 即将每个 batch 中的数据减去其均值并除以其方差, 使得处理后的数据符合标准正态分布, 因此消除了因数据量级差异带来的不良影响. 之后, 对标准化处理后的表征 h_1 和 h_2, 模型使用公式 (6) 来计算它们之间的互相关矩阵. 其中, b 代表不同训练批次的索引, i 和 j 分别代表 h_1 和 h_2 中特征分量的维度索引.

$$C_{ij} = \frac{\sum_b h_{b,i}^1 h_{b,j}^2}{\sqrt{\sum_b (h_{b,i}^1)^2}\sqrt{\sum_b (h_{b,j}^2)^2}} \tag{6}$$

公式 (7) 展示了对比损失函数的细节, 等式的右侧由两部分组成: (1) 不变项 $\sum(1-C_{ii})^2$, 即迫使互相关矩阵 C 的对角线元素恒等于 1, 以消除不同属性异构图编码器对图数据的嵌入带来的影响, 即要求不同属性异构图编码器对相同输入数据的编码应当相似; (2) 冗余减小项 $\lambda\sum\sum C_{ij}^2$, 即使得互相关矩阵 C 的非对角线元素趋近于 0, 目的是对特征表示进行解耦并减少特征向量各个分量之间的冗余. 此外, 在公式 (7) 中, λ 是控制不变项和冗余减小项之间相对重要性的权重系数.

$$\mathcal{L}_{ij}=\sum_i(1-C_{ii})^2+\lambda\sum_i\sum_{j\neq i}C_{ij}^2 \tag{7}$$

在模型具体训练的过程中, 由于向量间的互相关矩阵的计算不符合交换律. 为提高模型鲁棒性, 本文采用了对称反向传播梯度, 即在计算互相关矩阵时, 分别以 h_1 和 h_2 为锚点向量, 通过交换表征 h_1 和 h_2 的前后次序分别计算得到 C 和 C', 并在此基础上分别计算对比损失 $\mathcal{L}_{ij}$ 和 $\mathcal{L}_{ji}$, 之后取均值作为模型最终的损失函数. 对比损失函数可进一步改写为公式 (8):

$$\mathcal{L}=\frac{1}{2}(\mathcal{L}_{ij}+\mathcal{L}_{ji}) \tag{8}$$

算法 1. MD-GCL 训练算法.

输入: “域名-IP”属性异构图 $G(\mathcal{V},\mathcal{E},\mathcal{T},\mathcal{M})$;
输出: Barlow Twins 对比损失 $\mathcal{L}_{C\mathcal{L}}$.

1. 初始化图编码器 g_1 和 g_2
2. 加载 G 中不同类型节点的特征矩阵 *x_dict* 和不同元路径关系下的邻接矩阵 *edge_index_dict*
3. **for** *iter* = 1; $t\leqslant max\_iter$; *iter*++ **do**
4. // 非对称编码器编码, 并规范化处理
5. $h_1=g_1(x\_dict,edge\_index\_dict)$
6. $h_2=g_2(x\_dict,edge\_index\_dict)$
7. $h_1=(h_1-h_1.mean(0)/(h_1.std(0))$
8. $h_2=(h_2-h_2.mean(0)/(h_2.std(0))$
9. // Barlow Twins 损失函数计算
10. $C=mm(h_1h_2^{\mathrm{T}})/batch\_size$
11. $\mathcal{L}_{C\mathcal{L}}=\min(\mathcal{L}(C,E))$
12. 使用梯度估计器对 AdamW 优化器进行更新
11. **end for**
12. **return** $\mathcal{L}_{C\mathcal{L}}$

综上, 基于第 4.1 节和第 4.2 节, 本文给出 MD-GCL 模型的训练过程, 如算法 1 所示. 首先, 使用非对称的异构图编码器 g_1 和 g_2 对输入的异构图数据 G 进行嵌入, 分别得到节点嵌入 h_1 和 h_2 并进行标准化处理; 其次, 计算 h_1 和 h_2 的互相关矩阵 C, 并基于 Barlow Twins 损失函数不断优化模型参数.

4.3 下游任务

深度学习可分为表示学习和归纳偏好学习两部分. 表示学习是指模型通过学习更优的样本编码形式来降低下游任务难度; 归纳偏好学习是指根据任务目标选择合理的模型结构和参数, 使模型适应任务的分布和要求. 本文提出的恶意域名检测模型采用无监督表征学习策略: 在表示学习阶段, 利用非对称异构编码器和冗余减少损失函数训练无标签“域名-IP”二部图数据, 以挖掘异构图数据中的隐藏结构和规律; 在归纳偏好阶段, 固定住预训练好的编码器参数并嵌入异构图数据, 仅在下游分类任务的监督下训练分类器来预测域名标签.

具体实验过程如下: 首先, 利用 MD-GCL 训练算法预训练无标签“域名-IP”二部异构图数据; 其次, “冻结”预训

练好的模型参数, 输入完整的图数据, 编码得到“冻结”的域名节点嵌入; 最后, 遵循 Velickovic 等人[27]所提出的标准线性评估协议, 使用 Scikit-Learn 库初始化一个基于 L2 正则化的逻辑回归分类器 (或者随机森林分类器), 将域名节点嵌入根据域名样本数切分为训练、测试和验证集, 并用训练集中的域名嵌入和标签数据来拟合分类器, 然后预测测试集中的域名节点标签. 基于预训练的异构图编码器, 我们能够在少量训练标签下精准预测大多数域名的标签. 以逻辑回归分类器为例.

上述过程的数学原理如公式 (9) 所示, 其中 θ 代表预训练任务需优化的编码器参数, h_θ 代表特定参数下的异构图编码器, $\mathcal{L}_{\mathrm{BT}}$ 代表 Barlow Twins 损失函数, C 和 E 分别代表不同分支嵌入间的互相关矩阵和单位矩阵; LR 代表逻辑回归分类器, w 代表其参数, $\mathcal{L}_{\mathrm{cls}}$ 代表负对数似然损失函数.

$$\theta=\arg\min_{\theta}\mathcal{L}_{\mathrm{BT}}(C,E),\ w=\arg\min_{w}\mathcal{L}_{\mathrm{cls}}(h_\theta,LR_w) \tag{9}$$

本文使用逻辑回归和随机森林分类器的原因有以下两点: (1) 逻辑回归算法是评估图对比学习模型嵌入质量的标准线性评估模型; (2) 同时引入非线性的随机森林分类器对模型性能进行评估, 确保良好的检测性能依赖于良好的编码质量而不依赖于特定的分类模型. 下面简单介绍本文使用的逻辑回归和随机森林算法.

● 逻辑回归

逻辑回归建立在线性回归的基础之上. 线性回归解决的是回归问题, 逻辑回归在线性回归的基础上添加 Sigmoid 函数, 将输入的域名节点嵌入映射为取值在 0 和 1 之间的离散型分布, 相对应的离散数值即为数据样本属于良性或恶意域名的概率值, 因此逻辑回归可应用于本文恶意域名检测问题.

● 随机森林

随机森林隶属于机器学习分支之一集成学习的 bagging 方法. 森林的基本单位是决策树, 不同的决策树之间互相独立没有关联. 执行分类任务时, 每当有新的域名节点嵌入被输入, 森林中的每一颗决策树都独立地做出判断并最终以多数决策树的判别作为对当前域名类别标签的预测. 随机森林的优势在于实现简单、无需降维和特征选择, 且训练速度快、不容易过拟合.

5 实验与评估

5.1 实验数据

PDNS-Net[13]是一个大规模的域名系统数据集, 包含了通过被动 DNS 技术收集到的 2021 年 10 月份的互联网域名解析数据. 该数据集包含了两个不同规模的子数据集, 分别为 DNS 和 mDNS 数据集, 详情如表 1 所示. DNS 和 mDNS 数据集都包含了域名节点和 IP 节点, 本文对 PDNS-Net 数据集进行预处理, 基于域名之间的从属关系、相似度度量以及域名和 IP 地址之间相互解析的关系, 构建了 3 种不同的元路径.

表 1 PDNS-Net 数据集概况

数据集	域名总数	IP地址总数	边数	恶意域名数	良性域名数
mDNS	7495	4505	37285	2827	4668
DNS	373475	73593	897588	20354	4963

本文自监督学习任务接受表 1 所示的全量数据作为输入: mDNS 数据集共有 12000 个节点和 37285 条边参与预训练; DNS 数据集则共有 447028 个节点和 897588 条边参与预训练. 此外, 基于无监督表征学习训练模式的下游任务分类器仅接受具有标签标记的域名节点嵌入作为输入, 本文将该过程训练集的比例设置为 70%, 因此 mDNS 数据集和 DNS 数据集分别有 5205、17722 个域名节点嵌入参与下游任务分类器的训练过程.

5.2 实验环境以及评价指标

本文实验环境的相关参数如表 2 所示.

本文使用混淆矩阵评估分类预测结果, 以验证模型有效性. 混淆矩阵按真实类别和预测类别汇总样本来反映

模型在各类别上的性能. 混淆矩阵的取值分为以下 4 种情况: *TP* (正例预测为正例), *FN* (正例预测为负例), *FP* (负例预测为正例), *TN* (负例预测为负例). 基于此, 我们计算了精确率、召回率等评估指标, 它们的定义如下: (1) 精确率 (*Precision*). 指模型正确预测为正例的样本数与模型总共预测为正例的样本数之比, 反映了模型在预测正例时的准确性; (2) 召回率 (*Recall*). 指模型正确预测为正例的样本数与真实正例的样本数之比, 反映了模型在覆盖正例时的完整性; (3) *F*1 分值 (*F*1). 指精确率和召回率的调和平均值, 反映了模型在平衡精确性和完整性方面的综合性能. 公式依次表示如下:

$$Precision = \frac{TP}{TP+FP} \tag{10}$$

$$Recall = \frac{TP}{TP+FN} \tag{11}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{12}$$

表 2 实验环境与配置

类别	配置
系统版本	Ubuntu 20.04
CPU	i7-10700F@2.90 GHz×16
GPU	RTX 3060Ti
内存	31.3 GB
Anaconda3	4.9.2v
Python	3.9.7v
PyTorch	1.9.0v+cu111

5.3 实验设置与基准模型

首先, 为验证 MD-GCL 的有效性, 本文基于 PDNS-Net 数据集, 与涵盖了近几年各种主要设计思路的检测模型就检测性能、收敛速度、计算资源消耗等方面进行了广泛充分的对比实验: (1) 基于异构图神经网络 HeteroSAGE 和 HeteroGAT 的恶意域名检测方法与基于同构图神经网络 GraphSAGE[34]和 GAT[22]的恶意域名检测方法间的对比实验; (2) 各基线模型与 MD-GCL 的 TSNE 聚类分析对比实验; (3) 基于异构图神经网络 HAN[24]、HGT[26]的恶意域名检测与 MD-GCL 之间的对比实验; (4) 基于同构图神经网络 GCN[12]、GAT、GraphSAGE 的恶意域名检测与 MD-GCL 之间的对比实验; (5) 基于图对比学习方法的检测模型 DGI[27]、BGRL[31]与 MD-GCL 之间的比较, 以验证模型所使用的对比学习策略的先进性; (6) 各基线模型和 MD-GCL 在少标签场景下的恶意域名检测性能对比实验. 其中, 实验 (1) 和实验 (2) 分别是对设计目标 (1) 和目标 (2) 的验证, 实验 (3)–(5) 则是对设计目标 (3) 的验证.

本文所选取的对比模型简要介绍如下.

● 异构图神经网络模型

异构图注意力网络 HAN 在节点层级和元路径层级分别引入注意力机制, 以充分考虑不同类型或相同类型但不同个体的邻居节点以及不同元路径对于目标节点的重要性. HGT 将 Transformer 机制引入到异构场景中, 在节点层次和边层级上使用自注意力机制来自动抽取对下游任务有用的元路径进行学习. 此外, HGT 使用相对时间编码机制以处理异构图的动态性, 并设计了异构子图采样算法来处理 Web 规模的庞大数据.

● 同构图神经网络模型

本文选取了 GCN、GAT 以及 GraphSAGE 这 3 种经典的同构图神经网络模型作为 MD-GCL 的对比模型. 图卷积神经网络 GCN 将定义在欧几里得空间上的拉普拉斯算子和傅里叶变换应用于非欧几里得结构的拓扑结构上, 提出图的频域卷积操作. GraphSAGE 通过对邻居节点进行采样并不断聚合, 并使用融合后的信息对节点的标签进行预测, 解决了 GCN 全图训练方式下的耗时严重的问题. Veličković 等人于 2018 年提出图注意力网络 GAT[22], 其使用自注意力机制来计算图里面的节点相对于其邻居节点的注意力, 并将节点自身的特征和注意力特征拼接在

一起作为该节点的特征, 以解决 GCN 不能直接用于有向图以及不能够为每个邻居节点分配不同的权重等缺陷.

● 图对比学习模型

DGI 使用 GCN 作为编码器, 并在预训练过程中最大化节点表征和图的全局表征之间的互信息. BGRL 则使用非对称孪生网络的架构, 在预训练过程中先进行数据增强, 然后最大化原始数据的两个视图表示之间的相似性. 在下游任务阶段, 同 MD-GCL 模型一致, 本文使用逻辑回归和随机森林算法作为 DGI 和 BGRL 算法的下游任务分类器.

其次, 本文通过实验研究了不同组件和超参数设置对 MD-GCL 模型效果的影响, 主要通过消融实验完成: (1) 探究不同图神经网络编码器对 MD-GCL 检测性能的可能影响; (2) 探究下游任务中不同的分类器如逻辑回归、随机森林对 MD-GCL 检测性能的具体影响; (3) 探究学习率参数设置对模型检测性能的影响; (4) 探究训练集、测试集以及验证集比例划分对模型检测性能的影响.

5.4 对比实验结果与分析

● 异构场景建模有效性分析

如表 3 所示, 本文将恶意域名检测场景分别建模为同构图和异构图, 并使用相应的图神经网络模型在有监督场景下对恶意域名进行检测.

表 3 同异构场景下的恶意域名检测性能

模型	mDNS数据集			DNS数据集		
	整体$F1$	良性域名$F1$	恶意域名$F1$	整体$F1$	良性域名$F1$	恶意域名$F1$
GAT	0.8338	0.8762	0.7871	0.8636	0.4925	0.9207
GraphSAGE	0.8465	0.8839	0.8087	0.8537	0.4178	0.9253
HeteroGAT	**0.8485**	**0.8839**	**0.8082**	**0.9301**	**0.8027**	**0.9611**
HeteroSAGE	**0.8912**	**0.9136**	**0.8575**	**0.9268**	**0.8176**	**0.9543**

表 3 中, HeteroGAT 和 HeteroSAGE 是针对异构图的图模型, 它们为每个元路径视图单独定义了一个卷积层用于嵌入; GAT 和 GraphSAGE 是针对同构图的模型, 它们没有区分不同的元路径视图. 实验结果表明, 异构图模型在恶意域名检测上具有显著的优势. 在 mDNS 数据集上, HeteroGAT 和 HeteroSAGE 的检测性能均优于 GAT 和 GraphSAGE, HeteroGAT 在整体和恶意域名检测性能上分别比 GAT 高出 1.5% 和 2.11%, HeteroSAGE 则分别高出 4.5% 和 5%; 在 DNS 数据集上, HeteroGAT 和 HeteroSAGE 也表现出较大优势, HeteroGAT 在整体和恶意域名检测性能上分别比 GAT 高出约 6.7% 和 4.04%, HeteroSAGE 则分别高出 7.3% 和 2.9%. 上述实验分析表明异构图能够有效地表征恶意域名检测场景中的复杂关联和丰富信息, 从而有助于提高恶意域名检测的准确率.

本文还通过实验评估了不同元路径设置策略的合理有效性. 表 4 比较了 HeteroGAT 和 HeteroSAGE 在 4 种不同的元路径设置下的性能表现, m1、m2 和 m3 分别代表基于相似度度量关系、顶点域共享关系及解析关系所构建的异构图元路径. 本文以域名和 IP 地址间的解析关系为基准元路径关系, 并在此基础上将其与其他元路径关系进行组合, 探究模型在不同元路径设置下的性能变化. 从表 4 中可以看出, HeteroGAT 和 HeteroSAGE 在使用 3 种元路径组合策略时都达到了最优的检测性能, 在 mDNS 数据集上的 $F1$ 分值分别为 84.85%、89.12%, 在 DNS 数据集上的 $F1$ 分值分别为 93.4%、92.68%; 仅使用基准元路径时的检测性能则明显较差, 在 mDNS 数据集上分别低于最佳性能 6%、4.36%, 在 DNS 数据集上则分别低于 1.15%、2.83%. 这说明元路径组合策略可以有效地利用异构图中的多种关系来提高检测效果, 由于大规模数据集中的数据样本更加丰富和多样, 因此同异构模型在 DNS 数据集上的性能差距较 mDNS 数据集有所缩小.

此外, 在基准元路径基础上添加不同类型的元路径关系对模型性能的提升程度也有所不同. 以 mDNS 数据集为例, 基于顶点域共享的元路径关系 (m2) 比基于语义相似度的元路径关系 (m3) 对模型性能的提升更显著, “m2+m3”设置下的 HeteroGAT 和 HeteroSAGE 检测性能要分别优于“m1+m3”设置 2.06 和 2.93 个百分点. 综上, 上述对比实验对设计目标一进行了有效验证.

表 4　不同元路径设置下的恶意域名检测性能

模型	元路径选择	mDNS数据集			DNS数据集		
		整体$F1$	良性域名$F1$	恶意域名$F1$	整体$F1$	良性域名$F1$	恶意域名$F1$
HeteroGAT	m3	0.7885	0.8367	0.7192	0.9225	0.7654	0.9523
	m1+m3	0.8072	0.8571	0.7355	0.9285	0.7904	0.9568
	m2+m3	0.8278	0.8623	0.7869	0.9295	0.7935	0.9592
	m1+m2+m3	**0.8485**	**0.8839**	**0.8082**	**0.9340**	**0.8139**	**0.9618**
HeteroSAGE	m3	0.8476	0.8835	0.7921	0.8985	0.6856	0.9412
	m1+m3	0.8579	0.8928	0.8063	0.9091	0.7245	0.9508
	m2+m3	0.8872	0.9101	0.8517	0.9256	0.7989	0.9612
	m1+m2+m3	**0.8912**	**0.9136**	**0.8575**	**0.9268**	**0.8176**	**0.9543**

● 聚类分析

为验证 MD-GCL 模型对比学习策略下域名嵌入的有效性, 本文首先将其与基线模型在 mDNS 和 DNS 数据集上各训练 100 个批次, 之后采用 TSNE 算法将各模型得到的域名嵌入降维到二维空间并绘制 TSNE 可视化图, 分别如图 5 和图 6 所示. 从图中可以看出, MD-GCL 模型得到的域名嵌入在 TSNE 图上展现出了明显的聚类效果: (1) 不同和相同类别的域名在二维空间中表现出良好的分离和聚集性质, 表明 MD-GCL 模型能够有效学习域名的高阶特征, 区分不同类别的域名并捕捉相同类别域名之间的相似性; (2) 相比于基线模型, 没有出现明显的重叠或混乱区域, 表明 MD-GCL 方法能够有效地降低不同类别域名之间的干扰; (3) 每个聚类的大小与真实的域名分布基本一致, 表明 MD-GCL 较好地捕捉到了数据集中的域名分布状况.

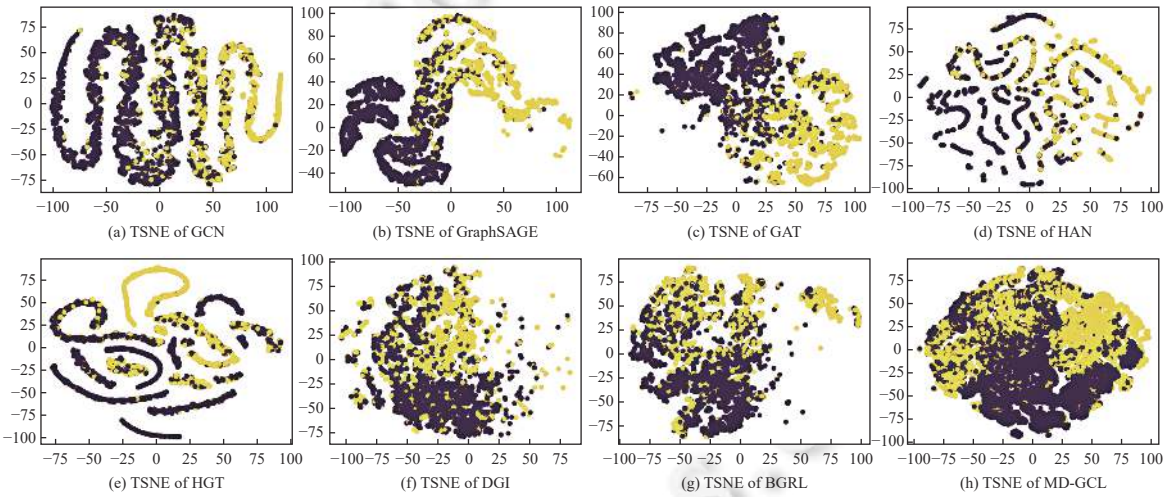

图 5　各模型 TSNE 可视化图 (mDNS)

相比之下, 其他基线模型得到的域名嵌入在 TSNE 图上表现出较弱的聚类效果: (1) 良性域名和恶意域名间的重叠或混淆区更加明显, 以 DNS 数据集为例, HAN 和 BGRL 模型的 TSNE 图中不同类型的域名重叠区域较多, 说明这些基线模型未能较好地区分不同类型的域名嵌入; (2) 相同类型的域名没有表现出较好的聚集特性, 如在 mDNS 数据集中, HAN、HGT 等模型的 TSNE 图呈线状分散分布, 说明这些基线模型未能有效地捕捉相同域名类型间的相似性. 综上, 通过 TSNE 聚类的可视化结果可以看出, MD-GCL 模型在较少训练批次下学习到的域名表示与基线模型相比具有更高的嵌入质量和更优的聚类效果, 验证了 MD-GCL 模型对比学习策略的有效性. 该聚类分析实验对设计目标 (2) 进行了有效验证.

● 检测性能对比

为验证设计目标 (3), 本文基于 mDNS 和 DNS 数据集详细评估了 MD-GCL 模型在不同分类场景下的性能表

现, 并与异构图神经网络算法、同构图神经网络算法以及图对比学习方法进行了详细的对比. 在实验中, 本文选取 $F1$ 分值作为主要的评价指标. 实验结果如表 5 所示. 从表 5 中观察可得, 本文提出的 MD-GCL 算法在 mDNS 和 DNS 数据集上均达到了最优的性能表现, 整体的 $F1$ 分值分别达到了 93.47%、95.85%, 大幅领先于其他算法.

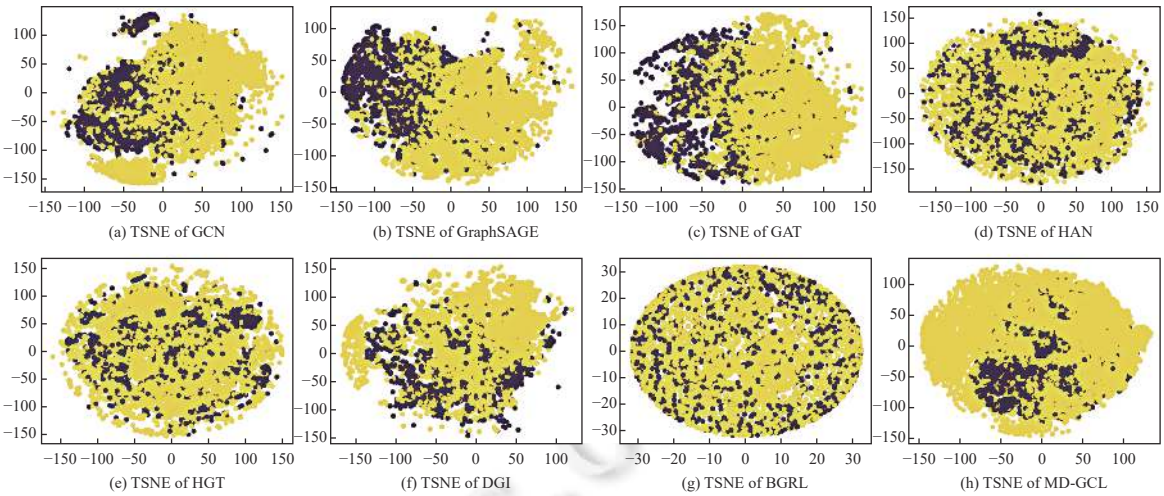

图 6 各模型 TSNE 可视化图 (DNS)

表 5 各算法在 mDNS 和 DNS 数据集上的性能表现

模型	mDNS数据集			DNS数据集		
	整体$F1$	良性域名$F1$	恶意域名$F1$	整体$F1$	良性域名$F1$	恶意域名$F1$
HAN	0.8127	0.8614	0.7163	**0.9079**	**0.7088**	**0.9579**
HGT	**0.8937**	**0.9102**	**0.8603**	**0.9192**	**0.7752**	**0.9503**
GCN	0.8439	0.8852	0.7838	0.8382	0.2965	0.9127
GAT	0.8634	0.8991	0.8207	0.8609	0.4529	0.9275
GraphSAGE	**0.8759**	**0.9012**	**0.8339**	0.8466	0.3541	0.9173
DGI	0.8356	0.8825	0.7568	0.8395	0.2841	0.9178
BGRL	0.8527	0.8862	0.8073	0.8773	0.6322	0.9285
MD-GCL	**0.9347**	**0.9467**	**0.9174**	**0.9585**	**0.8812**	**0.9826**

在 mDNS 数据集中, 恶意域名数和良性域名数之间的比例约为 1:1.65, 域名的各个类别分布较为均衡. 在该数据集上, MD-GCL 分类整体的性能表现分别要领先于次优的 HGT 和 GraphSAGE 算法约 4.1%、5.9%, 对于性能表现最差的 HAN 算法, MD-GCL 则领先其近 12%. 此外, 在 mDNS 数据集上, 绝大多数对比算法在识别正常域名上的准确率往往很高, 而在恶意域名的准确识别上表现较差. 譬如, HGT 算法在 7 种对比算法中取得了对恶意域名识别的最优性能表现, $F1$ 分值达到了 86.03%, 但仍落后于 MD-GCL 算法 5.71%.

mDNS 数据集是由 DNS 数据集抽样产生的一个小数据集, 其所涵盖的节点数和连接数都较少, 信息的匮乏使得大多数对比算法无法从数据中提取到对分类任务有用的表示. 本文提出的 MD-GCL 模型由于采用了设置恰当的对比学习策略, 因此可以挖掘到不同类型数据样本之间的本质特征, 从而实现了最优的分类性能. 从表 3 中还可以发现, 在 mDNS 数据集上 MD-GCL 模型的表现最优, 其次是同构图神经网络以及图对比学习方法, 而异构图神经网络算法的表现则最差. 我们推测, 由于现有的异构图神经网络算法过度依赖于预先设置好的元路径等高阶结构, 忽略了元路径外的节点关联, 因此在小规模数据集上提取信息的能力有限, 表现甚至不及同构图神经网络算法.

在 DNS 数据集中, 恶意域名数和良性、未标记域名数之间的比例极为不均衡, 它们之间的比例约为 1:17.3.

理论上而言, 在该数据集上准确识别出恶意域名的难度较大. 在该数据集上, MD-GCL 模型仍然取得最优的性能表现, 但各算法之间的性能差距进一步缩小. 在该数据集上, MD-GCL 的 $F1$ 分值达到了 95.85%, 领先于次优的 HGT 模型 3.93%, 领先于表现最差的 GCN 模型 12.03%. 此外, 在 mDNS 数据集上表现较差的 HAN, 此次在 DNS 数据集上则取得了较优的性能表现, 同 MD-GCL 相比差距仅有 5.06%. 除 HGT 和 HAN 外, 其他 5 种对比算法的 $F1$ 分值则散布于 82% 和 88% 之间, 整体差距不大. 其中, BGRL 模型取得了 87.73% 的 $F1$ 分值, 紧随其后的是同构图神经网络模型 GAT 和 GraphSAGE, 表现最差的则为 GCN 模型. 当数据集中包含更多的节点和连接时, 算法的性能在一定程度上会获得提升, 并且异构图神经网络算法所取得的提升要明显大于同构图神经网络算法和图对比学习算法, 这一点可以印证之前的猜想: 基于元路径获取信息的异构图神经网络算法只有在较大规模数据集上才能提取到充足的信息, 而同构图神经网络由于无法处理不同节点类型和不同边缘类型之间的关系, 因此基本没有提升.

由于 MD-GCL 专注于对比学习策略, 并通过对提取到的表征做解耦以区分开不同类型的节点, 既克服了基于元路径进行信息提取的异构图神经网络方法往往需要大量专业领域知识的不足, 也摆脱了同构图神经网络无法适用于异构图的缺陷, 因此无论在大规模还是小规模数据集中均取得了最优的性能表现. 需要指出的是, 在 DNS 数据集上 8 种算法识别恶意域名的 $F1$ 分值均超越了 90%, 尤其是 MD-GCL 模型取得了 97.89% 的分值, 接近于 100%. 但各个算法识别良性域名实体的 $F1$ 分值反而有所下降, 譬如, 传统的图神经网络模型 GCN、GAT 以及 GraphSAGE 在识别良性域名实体上的 $F1$ 分值均不足 50%, 这一现象背后的深层机理有待继续探索和验证.

● 收敛速度对比

在本文的实验条件下, 模型 MD-GCL 的收敛时间一般在 25–28 s 之间, 考虑到硬件算力配置对实际训练的时间影响较大, 篇幅所限, 这里不以秒数为主要衡量指标, 重点以训练批次分析模型的收敛性能. 以 mDNS 数据集为例, 图 7 和图 8 分别展示了各个模型的损失函数以及检测性能随训练次数变化的曲线图. 在图 5 中, 由于 GCN、GAT、GraphSAGE 以及 HAN、HGT 均使用交叉熵作为损失函数, 因此本文将上述图神经网络的损失函数变化曲线图合并到图 7(d) 中进行联合展示. 此外, 由于 MD-GCL、DGI、BGRL 和上述 GNNs 分别使用了不同的损失函数设计, 其损失函数值的分布并不在相似的区间, 因此本文为对比学习模型分别单独绘制了损失函数变化的曲线图, 分别如图 7(a)–(c) 所示.

观察图 7 可以发现, MD-GCL、BGRL、DGI 训练过程趋于收敛所用的训练批次数分为约为 350、500、300; GCN、GAT、GraphSAGE 这 3 个传统的图神经网络模型所花费的批次数均约为 800; HAN 收敛较快, 只花费了约 300 个训练批次; HGT 的损失函数变化曲线则较为波动, 模型未收敛. 此外, 结合图 8 可以发现, 收敛速度的快慢和检测性能的好坏并不简单地呈正相关关系, 如收敛较快的 DGI、BGRL 以及 HAN 的检测性能实际较差, 收敛速度次之的 GAT、GraphSAGE、GCN 实际检测性能达到了较优的水平, 未收敛的 HGT 检测性能仅次于 MD-GCL. 需要指出, MD-GCL 模型由于上下游任务相分离, 因此模型具体的检测性能会有所波动, 但整体表现均大幅领先于其他算法. 综上, MD-GCL 以较快的收敛速度同时实现了最优的检测性能, 充分证明了其对比学习策略设置的有效性.

● 内存开销对比

以 mDNS 数据集为例, 本文将 MD-GCL 及其对比基线模型的隐藏层维度均设置为 256, 将 GAT、HAN、HGT 和 MD-GCL 中的注意力头数均设置为 2, 在显存为 24 GB 的 RTX 4090 上进行实验并统计了各模型的内存开销情况. 如表 6 所示, BGRL 的显存开销最大, 约为其他模型的 2–3 倍; MD-GCL 的显存开销最低, 仅为 1 420 MB; GAT、HAN、HGT 等基于注意力机制的模型显存开销均超过了 2 GB, 平均显存占用高出 MD-GCL 约 900 MB. 需指出, GCN、GAT、GraphSAGE 以及 HAN、HGT、DGI 均为单编码器模型, BGRL、MD-GCL 均为双编码器模型. 本文提出的 MD-GCL 模型采用非对称异构编码架构, 将编码范围限制在各个元路径视图, 在较为复杂的双编码器设置下不仅实现了最优检测性能, 而且显存开销最低, 与同为双编码器设置的 BGRL 相比减小了近 70%. 上述实验分析论证了本文非对称异构编码器架构的优越性和高效性.

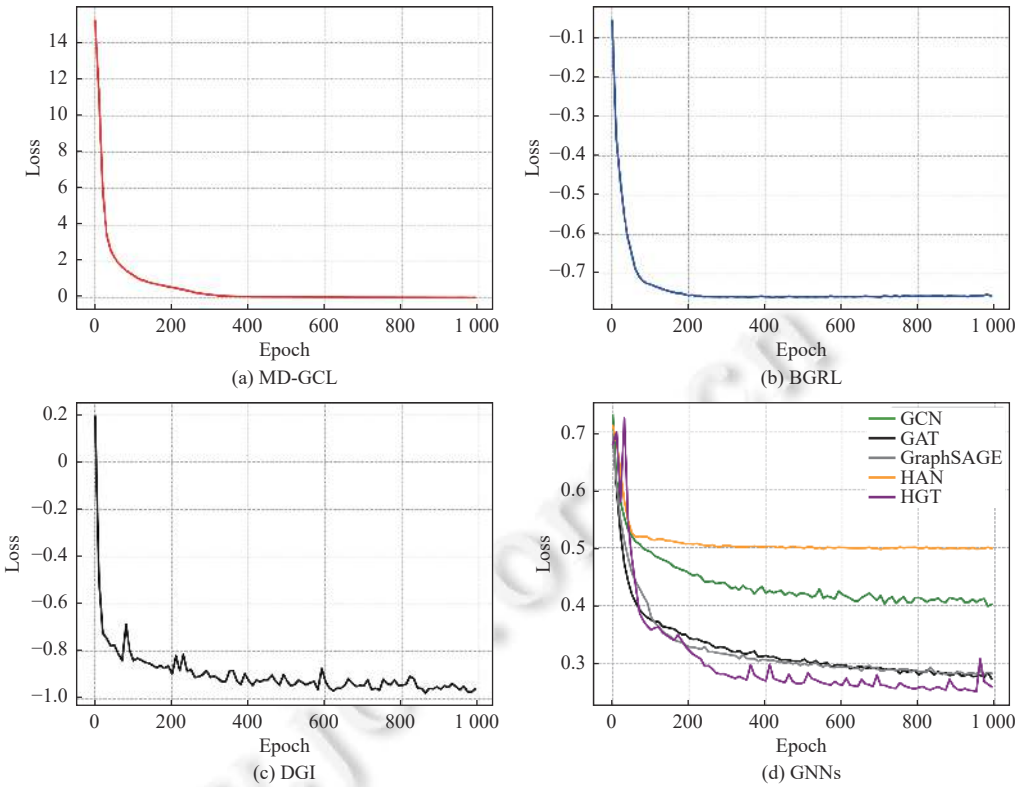

图 7 各模型损失函数变化曲线图

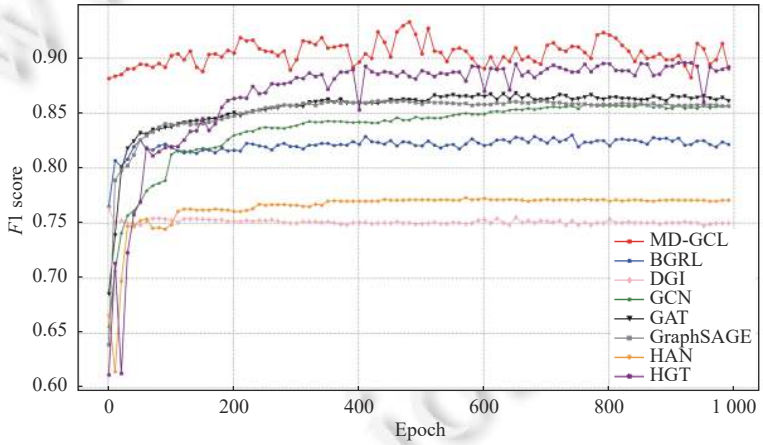

图 8 各模型检测性能变化曲线图

表 6 各模型内存开销对比 (mDNS)

模型	GCN	GAT	GraphSAGE	HAN	HGT	DGI	BGRL	MD-GCL
内存占用 (MiB)	1 594	2 486	1 448	2 138	2 358	2 512	4 608	1 420

● 少标签场景性能对比

本节通过实验验证了 MD-GCL 在标签稀缺情况下的优越性. 为评估不同模型在不同标签数量下的性能, 本文

在 mDNS 数据集上固定验证集比例为 0.1、训练集和测试集比例之和为 0.9, 并分别设置训练集比例为 0.001、0.01、0.02、0.05、0.1、0.2、0.4, 对应模型可见的域名节点标签数量分别约为 7、75、150、375、750、1499、2998. 图 8 展示了各个模型在不同标签数量下的 $F1$ 分值, 本文对标签稀缺情况下的性能差异进行了重点分析.

如图 9 所示, MD-GCL 在标签数量较少时具有最佳的检测性能: 当标签占比仅为 0.1% 时, MD-GCL 的 $F1$ 分值达到了 84%, 领先次优的 BGRL、GAT 和 GraphSAGE 模型近 10 个百分点, 远高于其他模型; 当标签占比为 1% 和 2% 时, 异构图神经网络模型 HAN、HGT 及同构图神经网络模型 GraphSAGE 性能提升明显, 但其最优性能仍低于 MD-GCL 约 7 个百分点; 当标签占比为 5% 时, MD-GCL 的 $F1$ 分值超过了 90%, 次优的 HGT 性能仅为 86%; 之后, 随着训练集所占比例的增加, 各模型检测性能基本稳定, 与表 4 描述一致, 并没有基线模型的检测性能超过 89%. 上述实验结果表明 MD-GCL 在标签稀缺情况下能够充分利用图结构和节点属性信息来提高检测性能, 而其他模型则受到标签数量的限制. 因此, MD-GCL 在标签稀缺情况下具有明显的优势.

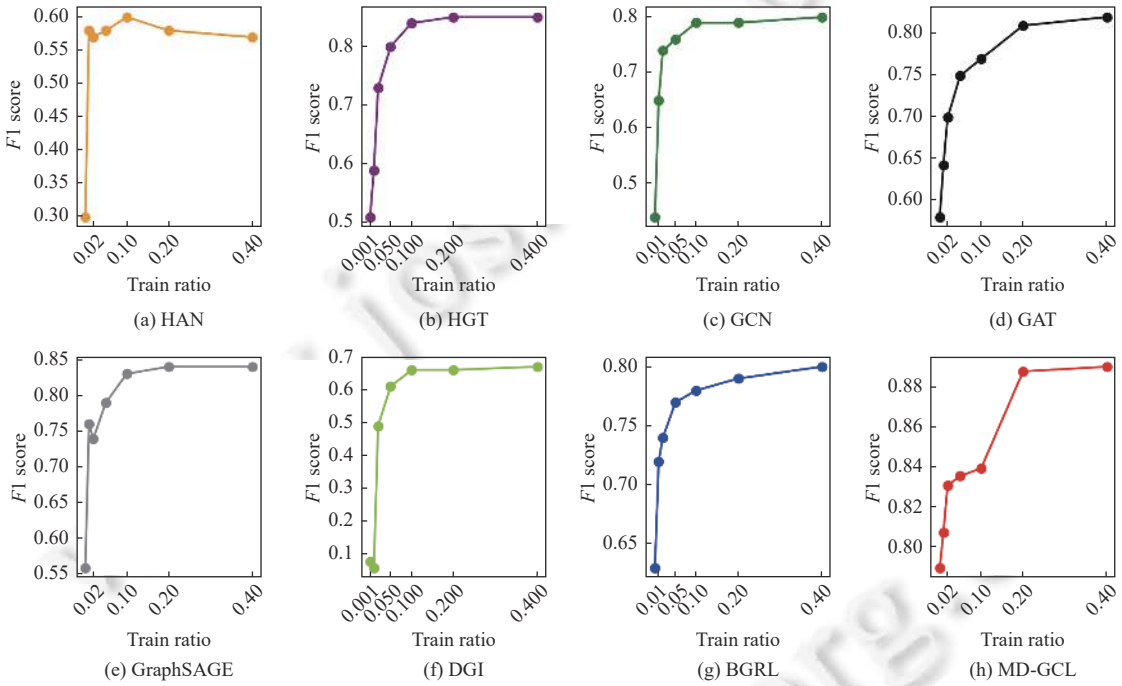

图 9 各模型在不同标签数量下的性能分布

此外, 本文绘制了各模型在少标签场景下的恶意域名检测性能变化曲线, 如图 10 所示. 观察可得, 在标签数量占比仅为 0.1% 时, MD-GCL 的恶意域名检测的 $F1$ 分值高达近 80%, 表现次优的 BGRL 的检测性能仅有 63%, 而其他基线模型更是低于 60%, 与 MD-GCL 相差甚远; 当标签数量占比增加到 2% 时, GraphSAGE 性能下降, 与 BGRL、GCN、HGT 等表现较优的模型持平, 其 $F1$ 分值均为 73%, 仍落后于 MD-GCL 约 10 个百分点. 实验结果表明, MD-GCL 在少标签场景下表现出优异的检测性能, 验证了非对称编码异构图对比学习策略在挖掘良性和恶意域名间差异方面的有效性, 为现实少标签场景下的恶意域名检测提供了新的解决方案.

5.5 消融实验结果与分析

在对比实验之外, 针对模型中不同参数和组件对实验结果可能带来的影响, 本文以 mDNS 数据集为例进行了消融实验, 实验结果阐述如下.

(1) HeteroGAT 和 HeteroSAGE 的编码组合要优于 HGT 和 HAN 的编码组合. 如表 7 所示, 本文以“A+B”的形式来表示不同的编码组合. 需要指出, “A+B”与“B+A”分别代表两种不同的编码器组合策略, 区别在于当 A 在前

时, 下游任务的输入为 A 所提取到的表征, B 在前时同理. 从表 7 中可以看到, 在模型其他设置都相同的条件下, “HeteroGAT+HeteroSAGE”和“HeteroSAGE+HeteroGAT”的编码器组合要明显优于“HAN+HGT”和“HGT+HAN”的编码器组合, 其中, “HeteroSAGE+HeteroGAT”的编码器组合在 mDNS 数据集上的性能最好, *F*1 分值达到了 93.47%, 从上到下依次领先于其他编码器组合策略 14.8%、9.74%、3.74%. 此外, 将表 5 与表 3 对比可得, 由“HAN+HGT”和“HGT+HAN”编码组合而成的对比学习模型, 其性能也要分别优于 HAN 和 HGT, 充分证明了自监督预训练模型的有效性和优越性. 最后, 受不同编码器编码能力的影响, 相同编码器组合的不同先后顺序对最终模型的分类性能也有着较大的影响, 在 MD-GCL 模型中, 我们最终选择了性能表现最优的“HeteroSAGE+HeteroGAT”编码器组合.

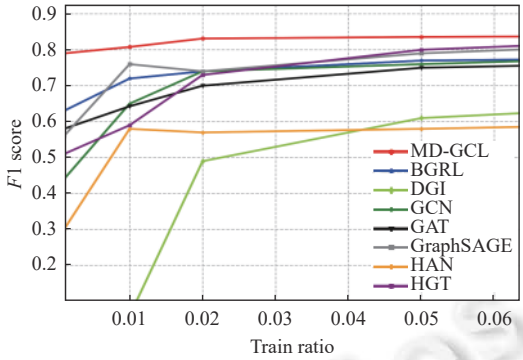

图 10 各模型在少标签场景下的恶意域名检测性能

表 7 不同编码器组合对 MD-GCL 模型性能的影响

编码器组合	整体*F*1	*F*1-Specific	
		0	1
HGT + HAN	0.7867	0.83	0.72
HAN + HGT	0.8373	0.88	0.74
HeteroGAT + HeteroSAGE	0.8973	0.91	0.86
HeteroSAGE + HeteroGAT	0.9347	0.94	0.91

(2) 选择下游任务分类器时, 随机森林算法比逻辑回归算法的性能更优. 如表 8 所示, 在 MD-GCL 模型其他参数设置都相同的条件下, 随机森林算法作为下游任务分类器时的 *F*1 分值, 要领先于逻辑回归算法 0.13%, 在对正常域名和恶意域名识别的效果上也分别领先逻辑回归算法 1%. 需要指出的是, 在进行消融实验时, 随机森林算法和逻辑回归算法均没有经过细致的调参, 该消融实验的结果旨在说明默认设置下的随机森林算法要优于逻辑回归算法.

表 8 不同下游任务分类器对 MD-GCL 模型性能的影响

下游任务分类器	整体*F*1	*F*1-Specific	
		0	1
随机森林	0.9186	0.92	0.89
逻辑回归	0.9173	0.91	0.88

(3) 不同学习率取值对模型效果影响不大. 如图 11 所示, 在使用逻辑回归作为下游任务分类器的条件下, 我们在区间 [0.0002, 0.003] 之间共选取了 9 种不同的学习率数值以评估模型的参数敏感性. 当学习率取值为 0.0004 和 0.002 时, MD-GCL 模型的 *F*1 分值均超过了 93%, 而当学习率取值为 0.003 的时候, 模型的检测性能最差, *F*1 分值为 91.87%, 和最优效果相比仅有 1.43% 的差值. 此外, 随着学习率取值的不断增大, 模型所取得的 *F*1 分值整体也呈现不断增长的趋势, 但当学习率取值过大时, 模型的检测性能会有所减弱. 学习率取值的合理区间应当设置为 [0.004, 0.002].

(4) 随着训练集所占比率的不断增加, MD-GCL 模型的检测性能整体呈上升趋势. 在实验中, 我们控制验证集的比例为 10% 且训练集和测试集所占比例之和为 90%, 并在此基础上不断调整训练集的比例, 实验结果如图 12 所示. 当训练集所占比例仅为 10% 时, MD-GCL 所取得的 *F*1 分值仅为 90.8%, 结合表 5 分析可得, 此时 MD-GCL 的检测性能仍然领先次优的 GraphSAGE 算法约 2.5 个百分点. 实验结果再次验证了 MD-GCL 模型的优越性. 此外, 随着训练样本所占比重的不断增加, 模型所取得的 *F*1 分值也不断提高, 在训练集所占比例为 70% 时, 模型的

检测性能最优, $F1$ 分值达到 93.34%. 然而, 当训练集所占比例过大时, 在较高的训练次数下模型的性能会有所下降, 我们推测模型此时产生了过拟合. 训练样本所占比例取值的合理区间应当设置为 [0.4, 0.7].

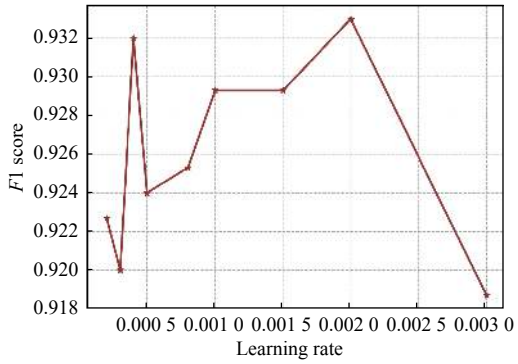

图 11 不同学习率下的模型检测性能

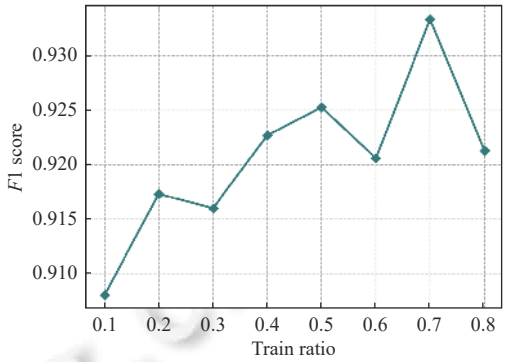

图 12 不同数据集划分比例下的模型检测性能

5.6 理论分析

图对比表征学习的嵌入质量可通过一致性 (alignment) 和均匀性 (uniformity) 度量指标进行衡量[38]. 一致性是指对比学习不同增强视图或编码分支的正样本对间应具有相似的表征, 均匀性是指映射到单位超球面的不同节点表征应均匀地分布在球面上, 即在特征空间中尽可能地保留不同类型节点的独有信息. 首先给出一致性度量指标的计算公式, 其被直接定义为正样本间的距离, 如公式 (13) 所示:

$$\mathcal{L}_{\text{aligned}}(g_1;g_2;\alpha) \triangleq \underset{x\sim p_{\text{data}}}{\mathbb{E}} \left[\| g_1(x)-g_2(x)\|_2^{\alpha}\right],\ \alpha>0 \tag{13}$$

其中, p_{data} 代表数据的样本分布, $g_1(\cdot)$ 和 $g_2(\cdot)$ 代表非对称互异的两个图编码器, α 的取值被设置为 2. 其次, 均匀性度量指标被定义为所有节点对表征间高斯核函数值的对数平均值, 如公式 (14) 所示, 其中 x 和 y 分别代表着数据分布中的不同样本.

$$\mathcal{L}_{\text{uniform}}(g;t) \triangleq \log \underset{x,y\ \text{i.i.d.}\sim p_{\text{data}}}{\mathbb{E}} \left[\mathrm{e}^{-t\|g(x)-g(y)\|_2^2}\right],\ t>0 \tag{14}$$

一致性度量指标期望正样本对间的距离越小越好, 均匀性度量指标期望不同数据样本应保留其彼此间的差异性, 因此公式 (13) 和公式 (14) 的取值均越小越好. 以 mDNS 数据集为例, 在训练 100 个批次的条件下, MD-GCL 及其对比基线模型的均匀性度量值变化曲线图如图 13 所示.

观察图 13 可得, 本文所提出的 MD-GCL 的均匀性分值分布于区间 (−5, −8), 而其他对比基线模型的均匀性分值则分布于区间 (0, −3). 其中, 均匀性分值较低的 HGT 模型的最佳分值也仅为−3, 高于 MD-GCL 最佳均匀性分值约 4.5. 上述实验结果表明 MD-GCL 在训练过程中保留了不同类别节点的更多信息, 因此其均匀性分值更低、节点嵌入质量更好. 此外, 随着训练批次的不断增加, MD-GCL 的均匀性分值不断下降且下降幅度较大, 而其他对比基线模型如 GCN、GAT、GraphSAGE、HAN、DGI 的下降幅度较小且均不超过 2, HGT 的下降幅度虽较大但存在一定波动, BGRL 甚至出现逆上升的变化趋势, 实验结果表明仅 MD-GCL 模型随着训练深度的增加其嵌入质量取得了较大幅度的稳步提升.

对于图对比学习模型 BGRL 和 MD-GCL, 图 14 给出了其一致性分数变化的曲线图, 可以发现 MD-GCL 的一致性分数要远低于 BGRL: BGRL 的一致性得分始终不低于 120, 而 MD-GCL 的一致性得分不高于 10. 上述实验结果表明 MD-GCL 正样本对表征间的差距较小, 因此较好地捕捉到了正样本对间的相似性. 综合一致性分数和均匀性分数两个指标可知, 本文 MD-GCL 模型相较于其他基线模型具有显著的优越性, 其编码质量更优、检测性能更好.

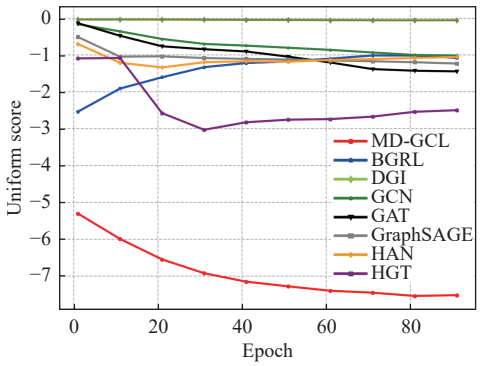

图 13 各模型均匀性分数值变化曲线图

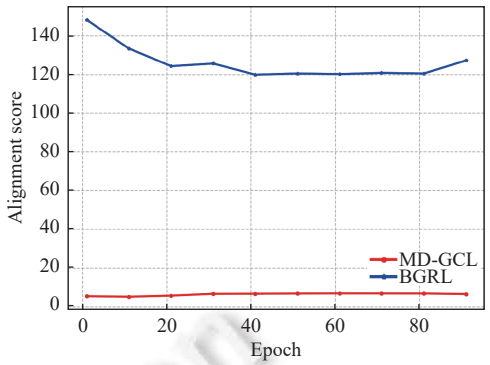

图 14 MD-GCL 和 BGRL 的一致性分数变化曲线图

6 总结和展望

本文首次将自监督机制引入到恶意域名检测领域, 并提出一种基于属性异构图对比学习的恶意域名检测方法 MD-GCL. 与 SimSiam[34]、BYOL[32]等基于孪生网络的对比学习方法不同, MD-GCL 不再采取数据增强策略而是使用两个不同的异构图编码器对原图进行嵌入, 因此避免了数据增强策略对图的语义信息带来的破坏. 此外, MD-GCL 使用 Barlow Twins 策略计算损失函数, 而不是计算不同表征之间的相似度, 通过对不同表征之间的互相关矩阵的各个分量做解耦, 并减少非对角线元素之外的冗余信息, 达到了区分开不同类别节点的目的. 在 mDNS 和 DNS 数据集上的实验结果证明, MD-GCL 要优于已有的同构和异构图神经网络分类算法以及 DGI、BGRL 等图对比学习方法. 未来值得进一步探索的工作主要有以下 3 点: (1) 探索 GraphSAGE 等同构图神经网络模型比异构图神经网络模型具有更好效果的深层机理; (2) 设计合适的对比学习策略, 从异构图中提取更为丰富的信息了, 并有效抑制噪声数据带来的负面影响; (3) 减小图对比学习中上下游任务之间的分歧, 将在预训练过程中学习到的知识更好地转移到下游任务中.

References:

[1] Liu WF, Zhang Y, Zhang HL, Fang BX. Survey on domain name system measurement research. Ruan Jian Xue Bao/Journal of Software, 2022, 33(1): 211–232 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6218.htm [doi: 10.13328/j.cnki.jos.006218]

[2] Fan ZS, Wang Q, Liu JR, Cui ZL, Liu YL, Liu S. Survey on domain name abuse detection technology. Journal of Computer Research and Development, 2022, 59(11): 2581–2605 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.20210121]

[3] Plohmann D, Yakdan K, Klatt M, Bader J, Gerhards-Padilla E. A comprehensive measurement study of domain generating malware. In: Proc. of the 25th USENIX Int'l Symp. on Security. Austin: USENIX Association, 2016. 263–278.

[4] Yadav S, Reddy AKK, Reddy ALN, Ranjan S. Detecting algorithmically generated domain-flux attacks with DNS traffic analysis. IEEE/ACM Trans. on Networking, 2012, 20(5): 1663–1677. [doi: 10.1109/TNET.2012.2184552]

[5] Holz T, Gorecki C, Rieck K, Freiling FC. Measuring and detecting fast-flux service networks. In: Proc. of the 2008 NDSS Int'l Symp. on Network and Distributed System Security. San Diego: ISOC, 2008. 1–12.

[6] Zhauniarovich Y, Khalil I, Yu T, Dacier M. A survey on malicious domains detection through DNS data analysis. ACM Computing Surveys, 2019, 51(4): 67. [doi: 10.1145/3191329]

[7] Han CY, Zhang YZ, Zhang Y. Fast-flucos: Malicious domain name detection method for Fast-flux based on DNS traffic. Journal on Communications, 2020, 41(5): 37–47 (in Chinese with English abstract). [doi: 10.11959/j.issn.1000-436x.2020094]

[8] Zhang WW, Gong J, Liu Q, Liu SD, Hu XY. Lightweight domain name detection algorithm based on morpheme features. Ruan Jian Xue Bao/Journal of Software, 2016, 27(9): 2348−2364 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4913.htm [doi: 10.13328/j.cnki.jos.004913]

[9] Zhang B, Liao RJ. Malicious domain name detection method based on associated information extraction. Journal on Communications, 2021, 42(10): 162–172 (in Chinese with English abstract). [doi: 10.11959/j.issn.1000-436x.2021181]

[10] Sun XQ, Tong MK, Yang JH, Liu XR, Liu H. HinDom: A robust malicious domain detection system based on heterogeneous information

network with transductive classification. In: Proc. of the 22nd Int'l Symp. on Research in Attacks, Intrusions and Defenses. Beijing: USENIX Association, 2019. 399–412.

[11] Sun XQ, Wang ZL, Yang JH, Liu XR. Deepdom: Malicious domain detection with scalable and heterogeneous graph convolutional networks. Computers & Security, 2020, 99: 102057. [doi: 10.1016/j.cose.2020.102057]

[12] Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. In: Proc. of the 5th Int'l Conf. on Learning Representations. Toulon: OpenReview.net, 2017. 1–14.

[13] Kumarasinghe U, Deniz F, Nabeel M. PDNS-Net: A large heterogeneous graph benchmark dataset of network resolutions for graph learning. arXiv:2203.07969, 2022.

[14] Antonakakis M, Perdisci R, Dagon D, Lee W, Feamster N. Building a dynamic reputation system for DNS. In: Proc. of the 19th USENIX Conf. on Security. Washington: USENIX Association, 2010. 18.

[15] Bilge L, Sen S, Balzarotti D, Kirda E, Kruegel C. Exposure: A passive DNS analysis service to detect and report malicious domains. ACM Trans. on Information and System Security, 2014, 16(4): 14. [doi: 10.1145/2584679]

[16] Peng CW, Yun XC, Zhang YZ, Li SH. Detecting malicious domains using co-occurrence relation between DNS query. Journal of Computer Research and Development, 2019, 56(6): 1263–1274 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2019.20180481]

[17] Zhang S, Zhou Z, Li D, Zhong YB, Liu QY, Yang W, Li S. Attributed heterogeneous graph neural network for malicious domain detection. In: Proc. of the 24th Int'l Conf. on Computer Supported Cooperative Work in Design. Dalian: IEEE, 2021. 397–403. [doi: 10.1109/CSCWD49262.2021.9437852]

[18] Wang X, Bo DY, Shi C, Fan SH, Ye YF, Yu PS. A survey on heterogeneous graph embedding: Methods, techniques, applications and sources. IEEE Trans. on Big Data, 2023, 9(2): 415–436. [doi: 10.1109/TBDATA.2022.3177455]

[19] Shi C, Li YT, Zhang JW, Sun YZ, Yu PS. A survey of heterogeneous information network analysis. IEEE Trans. on Knowledge and Data Engineering, 2017, 29(1): 17–37. [doi: 10.1109/TKDE.2016.2598561]

[20] Li JD, Dani H, Hu X, Tang JL, Chang Y, Liu H. Attributed network embedding for learning in a dynamic environment. In: Proc. of the 2017 ACM Conf. on Information and Knowledge Management. Singapore: Association for Computing Machinery, 2017. 387–396. [doi: 10.1145/3132847.3132919]

[21] Fu TY, Lee WC, Lei Z. HIN2Vec: Explore meta-paths in heterogeneous information networks for representation learning. In: Proc. of the 2017 ACM Conf. on Information and Knowledge Management. Singapore: Association for Computing Machinery, 2017. 1797–1806. [doi: 10.1145/3132847.3132953]

[22] Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. In: Proc. of the 6th Int'l Conf. on Learning Representations. Vancouver: OpenReview.net, 2018. 1–12.

[23] Schlichtkrull M, Kipf TN, Bloem P, van den Berg R, Titov I, Welling M. Modeling relational data with graph convolutional networks. In: Proc. of the 15th European Semantic Web Conf. Heraklion: Springer, 2018. 593–607. [doi: 10.1007/978-3-319-93417-4_38]

[24] Wang X, Ji HY, Shi C, Wang B, Ye YF, Cui P, Yu PS. Heterogeneous graph attention network. In: Proc. of the 2019 World Wide Web Conf. San Francisco: Association for Computing Machinery, 2019. 2022–2032. [doi: 10.1145/3308558.3313562]

[25] Zhang CX, Song DJ, Huang C, Swami A, Chawla NV. Heterogeneous graph neural network. In: Proc. of the 25th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining. Anchorage: Association for Computing Machinery, 2019. 793–803. [doi: 10.1145/3292500.3330961]

[26] Hu ZN, Dong YX, Wang KS, Sun YZ. Heterogeneous graph transformer. In: Proc. of the 2020 Web Conf. Taipei: Association for Computing Machinery, 2020. 2704–2710. [doi: 10.1145/3366423.3380027]

[27] Velickovic P, Fedus W, Hamilton WL, Liò P, Bengio Y, Hjelm RD. Deep graph infomax. In: Proc. of the 7th Int'l Conf. on Learning Representations. New Orleans: OpenReview.net, 2019. 1–17.

[28] Zhu YQ, Xu YC, Yu F, Liu Q, Wu S, Wang L. Deep graph contrastive representation learning. arXiv:2006.04131, 2020.

[29] You YN, Chen TL, Sui YD, Chen T, Wang ZY, Shen Y. Graph contrastive learning with augmentations. In: Proc. of the 34th Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 4883.

[30] Zhu YQ, Xu YC, Yu F, Liu Q, Wu S, Wang L. Graph contrastive learning with adaptive augmentation. In: Proc. of the 2021 Web Conf. Ljubljana: Association for Computing Machinery, 2021. 2069–2080. [doi: 10.1145/3442381.3449802]

[31] Thakoor S, Tallec C, Azar MG, Munos R, Veličković P, Valko M. Bootstrapped representation learning on graphs. In: Proc. of the 2021 ICLR Workshop. on Geometrical and Topological Representation Learning. Vienna: OpenReview.net, 2021. 1–14.

[32] Grill JB, Strub F, Altché F, Tallec C, Richemond PH, Buchatskaya E, Doersch C, Pires BA, Guo ZD, Azar MG, Piot B, Kavukcuoglu K, Munos R, Valko M. Bootstrap your own latent a new approach to self-supervised learning. In: Proc. of the 34th Int'l Conf. on Neural

Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 1786.
[33] Bielak P, Kajdanowicz T, Chawla NV. Graph Barlow Twins: A self-supervised representation learning framework for graphs. Knowledge-based Systems, 2022, 256: 109631. [doi: 10.1016/j.knosys.2022.109631]
[34] Chen XL, He KM. Exploring simple Siamese representation learning. In: Proc. of the 2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Nashville: IEEE, 2021. 15750–15758. [doi: 10.1109/CVPR46437.2021.01549]
[35] Zbontar J, Jing L, Misra I, LeCun Y, Deny S. Barlow Twins: Self-supervised learning via redundancy reduction. In: Proc. of the 38th Int'l Conf. on Machine Learning. PMLR, 2021. 12310–12320.
[36] Lv QS, Ding M, Liu Q, Chen YX, Feng WZ, He SM, Zhou C, Jiang JG, Dong YX, Tang J. Are we really making much progress? Revisiting, benchmarking and refining heterogeneous graph neural networks. In: Proc. of the 27th ACM SIGKDD Conf. on Knowledge Discovery & Data Mining. Singapore: Association for Computing Machinery, 2021. 1150–1160. [doi: 10.1145/3447548.3467350]
[37] Hamilton WL, Ying R, Leskovec J. Inductive representation learning on large graphs. In: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 1025–1035.
[38] Wang TZ, Isola P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In: Proc. of the 37th Int'l Conf. on Machine Learning. PMLR, 2020. 9929–9939.

附中文参考文献:

[1] 刘文峰, 张宇, 张宏莉, 方滨兴. 域名系统测量研究综述. 软件学报, 2022, 33(1): 211–232. http://www.jos.org.cn/1000-9825/6218.htm [doi: 10.13328/j.cnki.jos.006218]
[2] 樊昭杉, 王青, 刘俊荣, 崔泽林, 刘玉岭, 刘松. 域名滥用行为检测技术综述. 计算机研究与发展, 2022, 59(11): 2581–2605. [doi: 10.7544/issn1000-1239.20210121]
[7] 韩春雨, 张永铮, 张玉. Fast-flucos: 基于DNS流量的Fast-flux恶意域名检测方法. 通信学报, 2020, 41(5): 37–47. [doi: 10.11959/j.issn.1000-436x.2020094]
[8] 张维维, 龚俭, 刘茜, 刘尚东, 胡晓艳. 基于词素特征的轻量级域名检测算法. 软件学报, 2016, 27(9): 2348–2364. http://www.jos.org.cn/1000-9825/4913.htm [doi: 10.13328/j.cnki.jos.004913]
[9] 张斌, 廖仁杰. 基于关联信息提取的恶意域名检测方法. 通信学报, 2021, 42(10): 162–172. [doi: 10.11959/j.issn.1000-436x.2021181]
[16] 彭成维, 云晓春, 张永铮, 李书豪. 一种基于域名请求伴随关系的恶意域名检测方法. 计算机研究与发展, 2019, 56(6): 1263–1274. [doi: 10.7544/issn1000-1239.2019.20180481]

张震(1998－), 男, 硕士生, 主要研究领域为图神经网络, 图自监督学习, 恶意域名检测.

杨望(1979－), 男, 博士, 讲师, CCF 专业会员, 主要研究领域为威胁情报分析, 恶意软件检测, 网络安全应急响应.

张三峰(1979－), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为威胁情报分析, 智能安全, 对抗样本.