软件学报 ISSN 1000-9825, CODEN RUXUEW
[doi: 10.13328/j.cnki.jos.007211] [CSTR: 32375.14.jos.007211]

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

基于半监督和自监督图表示学习的恶意节点检测*

王晨旭[1,2], 王凯月[1], 王梦勤[1]

[1](西安交通大学 软件学院, 陕西 西安 710049)

[2](智能网络与网络安全教育部重点实验室 (西安交通大学), 陕西 西安 710049)

通信作者: 王晨旭, E-mail: cxwang@mail.xjtu.edu.cn

摘　要: 现实场景中, 电子商务、消费点评、社交网络等不同平台用户之间往往存在着丰富的交互关系, 将其构建成图结构, 并基于图神经网络 GNN 进行恶意用户检测已成为相关领域近几年的研究趋势. 然而, 由于恶意用户通常占比较小且存在伪装和标记成本高的情况, 导致了数据不平衡、数据不一致和标签稀缺等问题, 从而使传统 GNN 方法的效果受到了一定的限制. 提出基于半监督图表示学习的恶意节点检测方法, 该方法通过改进的 GNN 方法进行图节点表示学习并对图中节点分类. 具体地, 构造类别感知的恶意节点检测方法 (class-aware malicious node detection, CAMD), 引入类别感知注意力系数、不一致图神经网络编码器、类别感知不平衡损失函数以解决数据不一致与不平衡问题. 接下来, 针对 CAMD 在标签稀缺情况下检测效果受限的问题, 提出基于图对比学习的方法 CAMD$^+$, 引入数据增强、自监督图对比学习及类别感知图对比学习, 使模型可以从未标记的数据中学习更多信息并充分利用稀缺的标签信息. 最后, 在真实数据集上的大量实验结果验证所提方法优于所有基线方法, 且在不同程度的标签稀缺情况下都表现出良好的检测效果.

关键词: 恶意节点检测; 图神经网络; 表示学习

中图法分类号: TP393

中文引用格式: 王晨旭, 王凯月, 王梦勤. 基于半监督和自监督图表示学习的恶意节点检测. 软件学报. http://www.jos.org.cn/1000-9825/7211.htm

英文引用格式: Wang CX, Wang KY, Wang MQ. Malicious Node Detection Based on Semi-supervised and Self-supervised Graph Representation Learning. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7211.htm

Malicious Node Detection Based on Semi-supervised and Self-supervised Graph Representation Learning

WANG Chen-Xu[1,2], WANG Kai-Yue[1], WANG Meng-Qin[1]

[1](School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

[2](Ministry of Education Key Lab for Intelligent Networks and Network Security (Xi'an Jiaotong University), Xi'an 710049, China)

Abstract: In real-world scenarios, rich interaction relationships often exist among users on different platforms such as e-commerce, consumer reviews, and social networks. Constructing these relationships into a graph structure and applying graph neural network (GNN) for malicious user detection has become a research trend in related fields in recent years. However, due to the small proportion of malicious users, as well as their disguises and high labeling costs, traditional GNN methods are limited by problems suchas data imbalance, data inconsistency, and label scarcity. This study proposes a semi-supervised graph representation learning-based method for detecting malicious nodes. The method improves the GNN method for node representation learning and classification. Specifically, a class-aware malicious node detection (CAMD) method is constructed, which introduces a class-aware attention mechanism, inconsistent GNN encoders, and class-aware imbalance loss functions to solve the problems of data inconsistency and imbalance. Furthermore, to address the

* 基金项目: 国家自然科学基金 (62272379, T2341003); 陕西省自然科学基金 (2021JM-018); 中央高校基本科研业务费专项资金 (xzy012023068)
收稿时间: 2023-05-10; 修改时间: 2023-11-03; 采用时间: 2024-04-16; jos 在线出版时间: 2024-06-20

limitation of CAMD in detecting malicious nodes with scarce labels, a graph contrastive learning-based method, CAMD+, is proposed. CAMD+ introduces data augmentation, self-supervised graph contrastive learning, and class-aware graph contrastive learning to enable the model to learn more information from unlabeled data and fully utilize scarce label information. Finally, a large number of experimental results on real-world datasets verify that the proposed methods outperform all baseline methods and demonstrate good detection performance in situations with different degrees of label scarcity.

Key words: malicious node detection; graph neural network (GNN); representation learning

随着互联网的发展, 电子商务、社交网络、消费评价网站等平台近年来发展得越来越完善, 成为人们购物、获取信息、交流、娱乐等方面的重要工具. 然而, 随着这些平台的活跃用户数增多, 一些不法分子从中看到了牟利的机会. 如电商平台或评价网站中, 恶意用户通过虚假交易刷单, 刷好评等操作误导用户消费其产品; 社交网络平台中, 恶意用户可以通过注册大量机器人账号以传播垃圾信息, 如非法广告宣传, 炒作等, 从而牟取暴利; 在金融领域, 如支付宝恶意用户可以通过大量注册账号实现恶意套现[1], 对整个金融系统造成了极大的危害. 因此, 准确识别恶意用户对保障正常用户权益, 维护平台稳定性有着重要作用.

传统基于规则的恶意用户检测方法基于专家知识总结明显的行为信号, 并给予这些信号设计一些规则来进行欺诈预测. 这些方法虽然简单且存在可解释性, 但是其高度依赖于人们的先验知识, 难以处理不断变化和复杂的模式. 事实上, 大部分用户间天然存在着丰富的交互关系. 电商平台中, 用户可以与商家或其他用户进行交易, 不同用户会在同一商品下发布评论; 社交网络中用户间存在好友、共同关注、发布相同的话题等关系; 金融系统中的用户之间存在朋友、同事、亲戚等社交关系, 用户可能与商家或其他用户进行交易, 用户需要登录一些应用程序才能实现金融交易等. 所有这些关系都可能有利于解决恶意用户检测问题, 因为恶意用户可以通过改变攻击策略来使得基于规则或特征的检测方法失效, 但他们之间的关系却没有那么容易改变. 此外, 这些关系也可以为恶意用户的检测提供更多可挖掘的信息. 近年来, 基于图神经网络的研究越发成熟, 其优异的节点表示学习能力可以为节点同时编码丰富的特征信息与图结构信息, 将其与恶意用户检测相结合已经成为当前主流的研究方向.

然而, 当前基于图神经网络的恶意用户检测通常需要面临以下几个问题: (1) 随着黑色产业的发展, 恶意用户存在伪装行为, 如图 1 所示. 图中的虚线表示右侧恶意用户与左侧恶意用户具有相似的邻居, 而右侧恶意用户的特征对于识别左侧恶意用户至关重要, 但是二者之间互不连通, 从而对基于图神经网络 (GNN) 的识别模型的性能造成负面影响. 图 1 中的蓝线表示这两个恶意用户之间互为一阶邻居. 恶意用户的伪装行为主要在于利用一些伪装措施以模仿正常用户, 从而绕过检测系统的探查, 主要有以下两种伪装形式: ① 特征伪装: 恶意用户通过模仿正常用户的特征及行为, 从而绕过基于特征的检测模型; ② 结构伪装: 恶意用户通常会将自己与正常用户关联在一起, 降低被检测模型鉴定为恶意用户的概率. 恶意用户的伪装行为会导致构建的图数据存在数据不一致问题[2], 具体来说其与图神经网络通过聚合邻域信息来挖掘同一类节点共性的工作原理不一致, 在这种情况下使用图神经网络会使得编码得到的节点表示向量质量下降, 从而影响检测准确率; (2) 由于恶意用户通常只占所有用户的很小一部分, 因此恶意用户检测场景的数据常常存在严重的数据不平衡问题; (3) 由于恶意用户过于稀少, 数据中大多是正常用户, 将一部分欺诈行为贴上“异常”的标签成本非常高, 所以恶意检测任务还存在标签稀缺的问题.

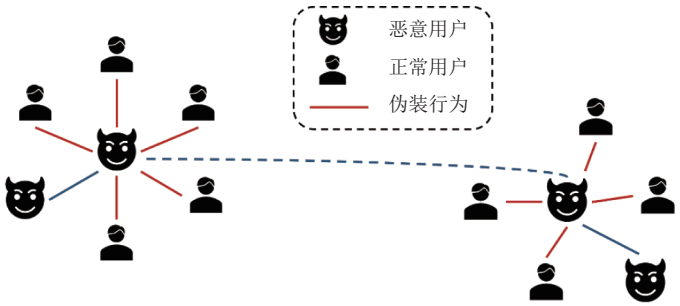


图 1　恶意用户伪装行为

针对以上问题, 本文提出了一种基于类别感知的恶意节点检测方法 CAM (class-aware malicious node detection), 该方法基于图神经网络, 以半监督学习的方式对模型进行训练. 为了解决数据不一致问题, 首先使用节点一致性度量模块来对节点间的一致程度建模, 并基于此构造类别感知注意力系数. 其次使用不一致图神经网络编码器对图中节点进行编码, 该编码器包含自适应邻域信息聚合, 高阶邻域信息获取与中间层组合 3 种机制. 为了解决数据不平衡问题, 在分类阶段引入类别感知不平衡损失函数, 以防止数据不平衡导致模型在少数类上过拟合. 此外, 为解决标签稀缺情况下模型效果受限的问题, 对 CAMD 进行优化, 提出基于图对比学习的 CAMD$^+$模型, 通过引入数据增强、自监督图对比学习、基于类别感知的平衡图对比学习, 使模型可以从未标记数据中学习更多信息, 提高模型的泛化能力与鲁棒性, 同时较为充分地利用稀缺的标签信息, 进一步提高节点表示向量的区分度. 通过在 5 个真实世界数据集上进行多组实验, 证明了本文所提方法在恶意节点检测任务中的有效性.

1 基于图神经网络的恶意检测相关工作

随着图神经网络 (graph neural network, GNN) 的发展, 现代算法倾向于利用 GNN 的强大功能学习有效的节点表示, 以识别嵌入空间中的异常实例. GEM[1]提出恶意用户具有设备聚集、行为聚集的特点. 设备聚集是指一个攻击者或攻击组织往往拥有大量的用户, 这些用户往往会在数量有限的设备上频繁注册或登录, 从而导致在单一设备上常常登录大量用户, 这些用户表现出设备聚集的特征, 对用于检测有组织的恶意用户具有重要意义[1]. GEM 构造一种用户-设备异构图神经网络来学习节点表示以挖掘具有这类表现的恶意用户. Liang 等人[3]提出在运费险诈骗恶意用户检测任务中, 基于设备构建图结构可以更好地学习有区分度的节点表示, 此外结合 GeniePath 方法[4], 使节点在表示学习过程中自适应地选择有价值的邻居信息进行聚合. SemiGNN[5]针对花呗违约用户检测任务, 基于用户社会关系和不同属性构建多视角图多方位的用户关系. 然而, 这些方法在设计面向恶意用户检测任务的 GNN 模型时, 都忽略了数据不一致性问题.

一些方法针对恶意用户的行为模式做出分析, 提出恶意节点检测任务场景中存在不一致问题, 不一致性问题与 GNN 模型的聚合过程有关: 聚合机制基于邻居共享相似特征和标签[6]的假设. 当不满足该假设时, 就不能再聚合邻域信息来学习节点表示. GraphConsis[2]第 1 个对该问题进行分析, 构造多关系图并提出了恶意检测任务中的上下文不一致、特征不一致、关系不一致与对应解决方案. CARE-GNN[7]在 GraphConsis 的基础上做出了改进, 其基于节点表示计算相似度, 并使用强化学习寻找最优邻居过滤阈值, 在聚合过程中去除相似度排名在阈值以外的邻居从而减轻恶意用户伪装行为对节点表示学习造成的干扰. PC-GNN[8]则基于上采样与降采样的思想, 设计了一个标签平衡的图采样器, 为节点选择潜在的邻居并构造子图, 而基于节点表示相似度设计一个距离函数, 为节点删除冗余的邻居, 同时为数据较稀缺的恶意节点增加距离小且标签相同的邻居. 然而, 这些方法通过改变图结构来解决数据不一致问题, 在一些情况下可能会造成图信息的损失, 对检测效果造成负面影响.

除了上述通过修改 GNN 的消息传递过程以减少噪声传播的方法, 一些基于对比学习思想的恶意节点检测方法提出了新的思路. DCI[9]提出了一种解耦合训练的方法, 基于图对比学习方法 DGI[10], 通过最大化节点表示与全局表示的互信息来训练 GNN 捕获全局信息. GCCAD[11]将每个节点与图的全局上下文 (例如, 所有节点的平均值) 进行对比, 设计了上下文感知的图对比损失函数以使正常节点与恶意节点的向量表示更有区分度, 此外通过节点表示向量计算边的可疑度, 并删除可疑边, 基于新的邻接矩阵进行节点表示更新.

上述方法都是基于空间域的 GNN, 近年来一些基于频谱域的恶意节点检测方法也被提出, 这些方法从图信号的角度进行分析, 提出传统 GNN 模型会过滤掉高频信号, 从而导致其无法区分学习到的异常节点与正常节点, 从而不可避免地导致图异常检测问题的性能不佳. BWGNN[12]提出图中恶意节点的存在将导致“右移”现象, 即随着异常程度的增大, 低频能量逐渐向高频转移, 据此提出 Beta 小波图神经网络. AMNet[13]提出了一种自适应多频率 GNN 模型用于图恶意节点检测, 设计了一个由 K 个图滤波器组成的滤波器组, 每个滤波器捕获不同频率的图信号, 通过组合图滤波器捕获不同频率的信息.

H2-FDetector[14]针对多关系图设计了一种新的信息聚合策略, 通过预测节点间边的性质, 对于不一致边两端的

节点在聚合时注意力系数为负, 一致边两端的节点在聚合时注意力系数为正. 同时对欺诈节点和正常节点进行原型提取, 缩短节点与其对应标签类型的原型之间的距离. H2-FDetector 与本文方法的区别在于: 1) H2-FDetector 考虑了数据存在的不平衡问题, 在训练时, 采用下采样的方式进行损失函数的计算, 每次计算时, 将负样本 (正常类) 随机采样到与正样本 (欺诈类) 相同的数量, 再进行计算; 本文所提方法没有进行任何的采样, 不会造成节点信息丢失, 但在设计损失函数时考虑了数据不平衡的影响, 以此解决数据不平衡问题. 2) H2-FDetector 模型聚合机制的目的是为不同类型的边传递不同消息; 本文提出的节点自适应邻域聚合机制能够为类别不同的邻居分配不同的权重. 3) H2-FDetector 在执行聚合操作时, 只聚合了一跳邻居信息; 本文提出的不一致图神经网络编码器可以获取两跳以内的邻居信息, 从而包含了高阶的邻域信息. 4) H2-FDetector 对数据不一致的处理策略是通过预测边类型, 之后将不一致边的聚合注意力系数设为负值, 以此防止节点信息被混淆; 本文提出的不一致图编码器同时考虑了不同类型邻居节点聚合时的注意力系数, 旨在减少类型相似度低的邻域节点信息聚合, 同时加强聚合类型相似度高的邻域节点信息.

2 问题描述与定义

本文的任务旨在发现电商、社交网络等不同领域平台中出现的恶意用户, 具体来说, 将恶意用户检测任务建模为基于图神经网络的恶意节点检测任务, 并采用半监督学习的方式进行节点分类. 首先, 基于用户间的关系构建图结构, 用户作为图中的节点, 正常节点的标签设为 0, 恶意节点的标签设为 1. 接着, 使用基于图神经网络的编码器为节点生成表示向量, 并将其送入分类器中计算每个节点为恶意节点的概率, 从而达到检测恶意用户的目的.

为了形式化定义, 本文用 $\mathcal{G}=\{\mathcal{V},\mathcal{E},\mathbf{X},\mathcal{Y}\}$ 表示包含有恶意节点的图数据, 其中 $\mathcal{V}$ 表示图中节点的集合 $\{v_1,\ldots,v_n\},(u,v)\in\mathcal{E}$ 表示节点 u 和 v 之间的边, 而 $\mathcal{N}(v)=u:(u,v)\in\mathcal{E}$ 表示节点 v 的邻居节点. 每个节点 v 的特征为 $\mathbf{x}_v\in\mathbb{R}^{d_x}$, 其中 d_x 表示节点原始特征的维度, $\mathbf{X}=[\mathbf{x}_1,\ldots,\mathbf{x}_n]\in\mathbb{R}^{n\times d_x}$ 表示所有节点特征构成的特征矩阵. $y_v\in\{0,1\}\in\mathcal{Y}$ 表示所有节点标签的集合.

图神经网络编码器 $g_\phi:\mathbf{x}_v\mapsto\mathbf{h}_v^{(L)}$ 的节点表示学习过程如公式 (1) 所示:

$$\mathbf{h}_v^{(l)}=UPDATE\left(\mathbf{h}_v^{(l-1)},AGG\left(\mathbf{h}_u^{(l-1)}:u\in\mathcal{N}(v)\right)\right) \tag{1}$$

其中, $\mathbf{h}_v^{(l)}$ 是节点 v 在第 l 层的表示向量, 其中 $l\in[1,L],h_v^{(0)}=f(x_v),f(\cdot)$ 为参数化函数, 可以将原始特征 $\mathbf{x}_v$ 映射到隐层空间, $AGG(\cdot)$ 表示邻域信息聚合函数, $UPDATE(\cdot)$ 为节点表示向量更新函数. 节点 v 经过 GNN 编码得到最终层表示向量后, 将其送入下游分类器 $f_\Theta:\mathbf{h}_v^{(L)}\mapsto\mathbf{p}_v$, $\mathbf{p}_v=[p_{v1},p_{v2}]$ 为节点 v 的类别概率分布, p_{v1} 为正常节点的概率, p_{v2} 为恶意节点的概率.

本文以半监督的方式对模型进行训练, 首先, 通过图编码器 g_ϕ 编码得到每个节点 $v\in\mathcal{V}$ 的表示向量. 随后, 通过最小化训练集上的分类损失对模型进行优化, 目标函数如公式 (2) 所示:

$$\begin{aligned}\Theta^*,\phi^*&\vDash\underset{\Theta,\phi}{\operatorname{argmin}}\mathcal{L}_{\text{GNN}}(\mathcal{E},\mathbf{X},\mathcal{Y}_{\text{train}},\Theta,\phi)\\&\vDash\underset{\Theta,\phi}{\operatorname{argmin}}\sum\nolimits_{v_i\in\mathcal{V}_{\text{train}}}J(f_\Theta(g_\phi(\mathbf{x}_{v_i})),y_i)\end{aligned} \tag{2}$$

其中, $J(\cdot)$ 为损失函数, $\mathcal{V}_{\text{train}}=\{v_1,\ldots,v_l\}\subset\mathcal{V},\mathcal{Y}_{\text{train}}=\{y_1,\ldots,y_l\},\Theta$ 为分类器的可训练参数, ϕ 为图编码器的可训练参数. 训练完成后, 预测节点集合中 $\mathcal{V}^U=\mathcal{V}/\mathcal{V}_{\text{train}}$ 中所有节点 u 的类别概率分布, 如公式 (3) 所示:

$$\mathbf{p}_u=f_{\Theta^*}(g_{\phi^*}(\mathbf{x}_u)) \tag{3}$$

其中, Θ^* 训练得到的最优分类器参数, ϕ^* 为最优的图编码器参数.

3 基于类别感知的恶意节点检测方法 CAMD

3.1 数据分析

为了更具体地了解恶意节点检测中存在的问题与伪装行为的比例, 本文对恶意检测任务场景中常见的数据进

行进一步地分析. 表 1 为恶意检测任务场景中常见数据的统计信息, 包括: 1) 数据中正常节点和异常节点的数量及各自的占比; 2) γ^f 为平均特征相似度[2], 用于描述数据的特征伪装比例, 即每个节点与其邻居节点的特征相似度的平均值; 3) γ^l 平均标签相似度[2], 用于描述数据的结构伪装比例, 即每个节点与其邻居节点的标签相似度的平均值; 4) $\hat{h}$ 为同质性度量分数[15], $\hat{h}$ 越大说明图的同质性越强; 5) 特征相似度与标签相似度的比值 $\frac{\gamma^f}{\gamma^l}$.

表 1 数据统计信息

Graph	#Users (normal, abnormal)	Feat Simi (γ^f)	Label Simi (γ^l)	$\hat{h}$	$\frac{\gamma^f}{\gamma^l}$
Amazon	11 944 (93.13%, 7.38%)	0.687	0.072	0.049	9.542
YelpChi	10 893 (85.25%, 14.75%)	0.988	0.157	0.029	6.293
Wiki	8 227 (97.36%, 2.64%)	0.554	0.043	0.010	12.651
Tencent-Weibo	8 405 (89.67%, 10.33%)	0.439	0.760	0.748	0.578
T-finance	10 035 (95.39%, 4.61%)	0.786	0.805	0.799	0.976

观察表 1 中的数据, 发现所有数据集都存在严重的数据不平衡问题, 这是前文提到的恶意检测任务的固有特点. 而数据不平衡会导致模型对少数类过拟合, 使得分类准确度降低. 此外, 大部分数据集都存在平均标签相似度较低, 而平均特征相似度较高的情况. 其中标签相似度较低可能是由恶意节点的结构伪装行为引起的. 特征相似度较高则可能是恶意节点的特征伪装行为导致其特征与正常节点相近. 传统 GNN 通过聚合邻域节点信息来挖掘图中同一类节点的共性信息, 结合对数据的分析, 发现对于本文中的恶意节点检测任务, 大部分的图数据都与 GNN 的工作原理不一致, 本文引入数据不一致的概念以更好地表述这一问题. 具体来说, 本文中的数据不一致体现在: 1) 图中大部分相邻节点类型不同, 数据统计表的 γ^l, $\hat{h}$ 列也验证了这一点, 这将导致节点聚合邻域信息时, 会将较多不同类型的节点信息聚合到自身, 从而使得到的表示向量特征变得混淆; 2) 相邻节点类型不同而特征相近, 这一点可以通过数据统计表的 $\frac{\gamma^f}{\gamma^l}$ 列得以衡量, 这可能是恶意用户的伪装行为造成的, 从而降低节点的区分度, 提高恶意节点识别的难度.

因此在恶意节点检测任务中, 传统的 GNN 模型无法学习到有效的节点表示向量, 从而导致分类效果变差. 综上所述, 本节通过对数据统计信息进行分析, 总结了恶意节点检测任务中存在的问题, 分别是数据不一致与数据不平衡问题.

3.2 CAMD 模型整体框架

本文将恶意节点检测任务建模为基于图神经网络的图节点分类问题, 构造了由节点一致性度量模块 f_s, 不一致图神经网络编码器 g_ϕ 与分类器 f_Θ 组成的恶意节点检测模型, 整体架构如后文图 2 所示.

模型的输入是一个待检测恶意节点的图数据 $\mathcal{G}$. 首先, 使用节点一致性度量模块 f_s 对节点间的一致程度建模, 一致程度描述了两个节点属于同一类的程度, 并基于此构造类别感知注意力系数 α. 其次, 使用不一致图神经网络编码器 g_ϕ 对图中节点 $v \in \mathcal{V}$ 进行节点表示学习, 该编码器包含自适应邻域信息聚合, 高阶邻域信息获取与中间层组合三种机制. 具体来说, 为了学习更有区分度的表示向量, 引入自适应邻域信息聚合机制, 基于上一步得到的类别感知注意力系数 α, 根据节点间的类别相似度自适应地聚合邻域信息. 为了获取更丰富的邻域信息, 引入高阶邻域信息获取机制, 在聚合过程中一次聚合两跳的邻居节点表示. 为了保证节点自身信息在图上的传播过程中不被削弱, 引入中间层组合机制, 拼接所有中间层表示 $\mathbf{h}_v^{(l)}(l \in [0, L])$ 作为节点的最终表示向量 $\mathbf{h}_v^{(\text{final})}$. 之后将不一致图神经网络编码器 g_ϕ 输出的 $\mathbf{h}_v^{(\text{final})}$ 作为分类器 f_Θ 的输入, 开始进行下游分类任务. 通过分类器得到每个节点的类别概率分布 $\mathbf{p}_v$, 并在训练集 $v' \in \mathcal{V}_{\text{train}}$ 上使用类别感知的不平衡损失函数 $\mathcal{L}_{\text{LDAM-RW}}$ 对模型进行优化, 同时这一步也对节点一致性度量模块 f_s 进行优化, 帮助其学习到更好的注意力系数. 最终通过训练好的模型得到所有节点的预测概率.

3.3 类别感知注意力系数

本节的主要目的是得到每个节点的类别感知注意力系数 α, 此注意力系数由节点一致性度量模块 $f_s: \mathbf{x}_v, \mathbf{x}_u \mapsto s_{vu}$

中得到的节点类型相似度 s 确定. 具体来说, 本文利用特征空间的信息, 引入一个与图结构无关的多层感知机(MLP) 从节点的原始特征提取类别信息, 计算每个节点 v 的类别概率分布向量 $\mathbf{s}_v$. 如公式 (4) 所示:

$$\mathbf{s}_v = \sigma(\mathbf{W}_s \mathbf{x}_v + b) \tag{4}$$

其中, $\mathbf{x}_v \in \mathbb{R}^{d_x}$ 为节点的原始特征向量, $\mathbf{W}_s \in \mathbb{R}^{d_s \times d_x}$ 为可训练的参数矩阵, d_s 为隐层维度, σ 为 Softmax 函数. 此方法中的 MLP 可以看作一个非线性的神经网络分类器, 通过最小化其预测类别的损失来对其进行优化, 具体如公式 (5) 所示:

$$\Theta_m^* = \underset{\Theta_m}{\operatorname{argmin}} \mathcal{L}_{\text{mlp}} = \underset{\Theta_m}{\operatorname{argmin}} \frac{1}{|\mathcal{V}_{\text{train}}|} \sum_{v \in \mathcal{V}_{\text{train}}} J(\hat{\mathbf{s}}_v^{\text{mlp}}, y_v) \tag{5}$$

其中, $J(\cdot)$ 为损失函数, Θ_m 为 MLP 中的参数, Θ_m^* 为优化后的参数, $\mathcal{V}_{\text{train}}$ 为训练集的节点, $\hat{\mathbf{s}}_v^{\text{mlp}}$ 则为训练集中的节点 v 经过 MLP 预测得到的类别概率分布向量, y_v 为节点 v 的真实标签.

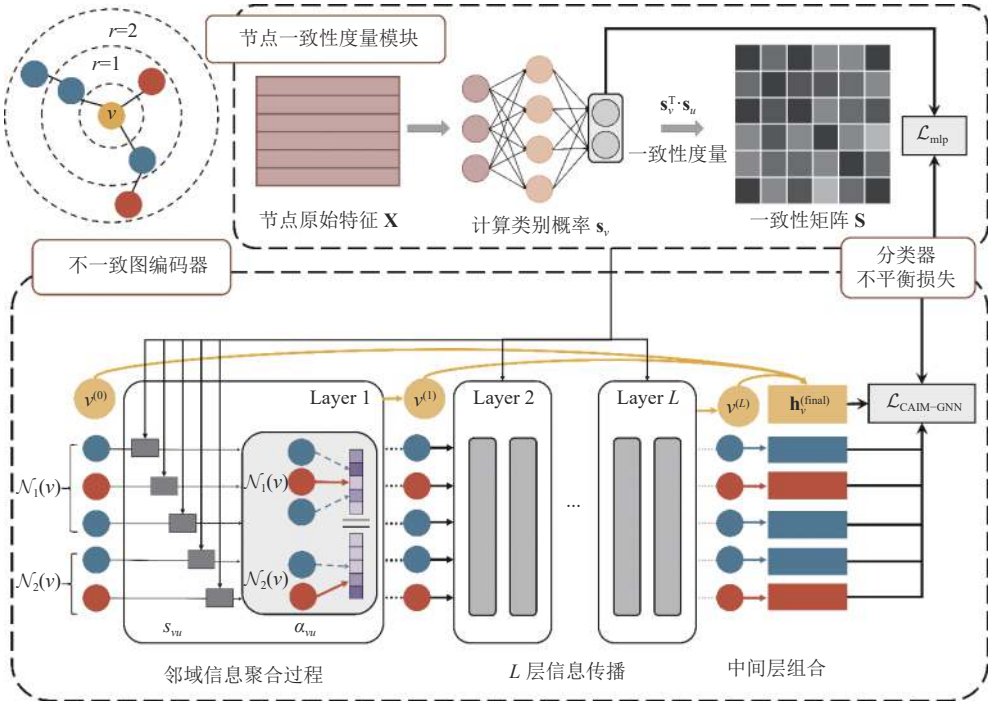


图 2 CAMD 模型图

基于得到的节点类别概率分布向量 $\mathbf{s}_v$, 对两个节点之间的一致程度进行度量, 具体来说就是计算两个节点属于同一类的程度, 即类别相似度 s_{vu} , 并将其作为聚合时的注意力系数 α_{vu} . 具体过程如公式 (6) 所示.

$$\alpha_{vu} = s_{vu} = \mathbf{s}_v^{\top} \cdot \mathbf{s}_u \tag{6}$$

其中, $s_{vu}, \alpha_{vu} \in (0,1)$ 是一个标量, $u \in \mathcal{N}(v), \mathcal{N}(v) = u : (u,v) \in \mathcal{E}$.

由于此相似度的估计仅基于节点原始属性 $\mathbf{x}_v$, 过程中不涉及对邻域信息的聚合操作, 因此不受图数据不一致问题的约束. 此类别感知注意力系数可以自适应地为类别相似度高的邻居节点学习较大权重, 为类别相似度低的邻居节点学习较小权重, 且不受图数据不一致问题的约束. 此外, 这一机制不仅可以在数据不一致性高的图上发挥作用, 在数据不一致性低的图上, 也可以提供较为直接的标签信息从而更好地指导聚合过程.

3.4 不一致图神经网络编码器

本节主要针对上文中提出的数据不一致问题, 对传统 GNN 架构进行改进, 引入自适应邻域信息聚合, 高阶邻域信息获取, 中间层组合三种机制, 在此基础上构建一种更适合本文任务场景的不一致图神经网络编码器. 传统

GNN 模型的表示学习框架如公式 (7) 所示:

$$\mathbf{h}_v^{(l)} = UPDATE\left(\mathbf{h}_v^{(l-1)}, AGG\left(\mathbf{h}_u^{(l-1)} : u \in \mathcal{N}(v)\right)\right) \tag{7}$$

公式 (7) 所示的节点表示学习过程中, 首先通过 $AGG(\cdot)$ 聚合邻域信息 (具体操作如, 求和、取均值、最大池化等), 再通过 $UPDATE(\cdot)$ 将邻域信息与上一层自身表示结合得到新的节点表示 (例如线性组合层). 本节将结合恶意节点检测任务场景中的问题, 对 $AGG(\cdot)$ 和 $UPDATE(\cdot)$ 进行改进.

3.4.1 自适应邻域聚合

为了学习到有区分度的节点表示向量, 本节对 $AGG(\cdot)$ 进行改进. 具体来说使用第 3.3 节中得到的类别感知注意力系数 α_{vu} 来指导信息聚合过程, 计算如公式 (8) 所示:

$$AGG\left(\mathbf{h}_u^{(l-1)}\right) = \sum_{u \in \mathcal{N}(v)} (\lambda_{\text{att}} \alpha_{vu} + \lambda_{\text{adj}}) \mathbf{h}_u^{(l-1)} \tag{8}$$

其中, $\mathcal{N}(v)$ 表示邻居节点的集合, $\mathbf{h}_u^{(l-1)}$ 表示邻居节点在 $l-1$ 层的表示向量. α_{vu} 为类别感知注意力系数, 节点 v 和 u 的类型相似度越高, α_{vu} 越大, 反之越小. 考虑到直接使用 α_{vu} 作为聚合权重可能会损失原本的图结构信息, 为了更灵活地调和注意力系数与原始邻接关系的比例, 此处引入两个超参 λ_{att} 与 λ_{adj}, 二者取值范围为 $[0,1]$, 前者越大则类别感知注意力系数对于聚合过程的贡献越大, 而后者越大则原始图结构信息在聚合过程中所做的贡献越大.

通过这种方式, 在聚合邻域信息时, 可以减少类型相似度低的邻域节点信息聚合, 同时多聚合类型相似度高的邻域节点信息, 从而让每个节点更准确地聚合到自身所需要的信息, 提高学习到的节点表示的区分度.

3.4.2 高阶邻域信息获取

本节的主要目的是使不一致图神经网络编码器在进行表示学习时, 可以为节点编码更丰富的邻域信息. 研究表明[16,17], 传统的图神经网络卷积层常聚焦于一个中心节点的局部性, 这种机制对于节点与邻居类型大多相同的情况很有效, 而对本文中数据不一致的情况反而会带来负面影响. 而每次聚合更高阶的邻居信息会减少这种影响, 对于大部分邻居节点与自身类型不同的情况, 获取全局信息而非仅获取局部信息会提升模型效果. 此外, 也有研究表明[8], 对于恶意节点检测问题, 恶意节点会表现出高度的结构相似性, 但可能彼此相距较远. 较直观的一个解释是, 平台中的恶意用户通常是有组织有预谋的黑色产业, 因此他们会避免直接产生联系, 但每个恶意用户的行为模式都较为相似. 鉴于此, 本节进一步对 $AGG(\cdot)$ 改进, 在进行聚合操作时, 将获取信息的范围从局部拓展到非局部, 以尝试从全局信息中挖掘恶意节点的共性.

具体来说, 每次对两跳的邻居信息分别进行上一节的自适应邻域信息聚合操作, 为了避免混合不同范围的邻域信息, 将两跳邻居信息以拼接的方式进行组合, 作为第 l 层的邻域信息, 具体如公式 (9) 所示:

$$AGG\left(\mathbf{h}_u^{(l-1)}\right) = AGG\left(\mathbf{h}_{u_1}^{(l-1)} : u_1 \in \mathcal{N}_1(v)\right) \| AGG\left(\mathbf{h}_{u_2}^{(l-1)} : u_2 \in \mathcal{N}_2(v)\right) \tag{9}$$

其中, $\mathcal{N}_1(v)$ 为节点 v 的一阶邻居集合, $\mathcal{N}_2(v)$ 为节点 v 的两阶邻居集合, $\|$ 代表拼接操作.

3.4.3 中间层组合

传统 GNN 编码器的表示学习过程如公式 (7) 所示, 这种表示学习方式在数据不一致的情况下, 首先可能会导致节点因为伪装行为或噪声将错误的邻域信息与自身合并, 造成最终学习到的节点表示特征混淆. 此外, 随着模型深度的增加, 学习到的节点表示会趋于相近, 增加过平滑的概率, 同时也会使不必要的噪声参与聚合, 特别是对于本文数据不一致的情况, 仅使用最后一层的表示向量用于分类任务, 很容易因为原本重要的信息被弱化或被抹去, 使得分类效果不佳.

图卷积操作的相关研究表明[17,18], 使用中心节点与邻域分离的策略会使节点表示即使在多层 GNN 卷积层中进行传播也不会出现和邻居节点非常相似的情况. 而图卷积网络每层学习到的信息都具有不同的局域性, 前几层更具有局部性, 而后几层则会捕获到越来越多的全局信息. 显式的组合这些信息可以在数据不一致情况下有效提升模型的效果.

基于以上思想, 本节改进 $UPDATE(\cdot)$, 在每一层更新节点表示时, 仅使用 $AGG(\cdot)$ 得到的邻域信息, 不再融合 $\mathbf{h}_v^{(l-1)}$ 的信息, 而将其以残差连接的方式直接传入最后一层. 具体来说, 将所有节点的中间层表示向量以拼接的形式

组合, 作为最终的节点表示向量. 结合前两节提出的 $AGG(\cdot)$, 改进后的不一致图神经网络表示学习过程如公式 (10) 和公式 (11) 所示:

$$\mathbf{h}_v^{(l)} = \sum_{u_1 \in N_1(v)} (\lambda_{\text{att}} \alpha_{vu_1} + \lambda_{\text{adj}}) \mathbf{h}_{u_1}^{(l-1)} \,\|\, \sum_{u_2 \in N_2(v)} (\lambda_{\text{att}} \alpha_{vu_2} + \lambda_{\text{adj}}) \mathbf{h}_{u_2}^{(l-1)} \tag{10}$$

$$\mathbf{h}_v^{(\text{final})} = \mathbf{h}_v^{(0)} \,\|\, \mathbf{h}_v^{(1)} \,\|\, \ldots \,\|\, \mathbf{h}_v^{(L)} \tag{11}$$

其中, $\|$ 表示拼接操作. 使用 $\mathbf{h}_v^{(0)} = \sigma(W_e \mathbf{x}_v)$ 将节点原始特征映射到隐层空间, $W_e \in \mathbb{R}^{d_h \times d_x}$ 为可训练权重矩阵, d_h 为初始隐层向量维度, σ 为 ReLU 函数, $\mathbf{h}_v^{(\text{final})} \in \mathbb{R}^{(2^{L+1}-1)d_h}$. 这样的 $UPDATE(\cdot)$ 方法, 可以保留每个节点自身原本的信息与表示学习过程中不同局域性的信息, 增加节点表示的表达能力.

3.5 类别感知不平衡损失函数

本节引入一种类别感知不平衡损失函数以解决上文中提出的数据不平衡问题. 对于本文中的恶意节点检测任务, 在下游分类任务中可以将其看作一个二分类问题. 构造一个分类器 $f_\Theta : \mathbf{h}^{(\text{final})} \mapsto \mathbf{p}$, 以第 3.4 节中的不一致图神经网络编码器输出的节点表示向量作为输入, 最终输出得到每个节点的类别概率分布, 其具体形式如公式 (12) 所示:

$$\mathbf{p} = \sigma\left(\mathbf{W}\mathbf{h}^{(\text{final})} + b\right) \tag{12}$$

其中, $\mathbf{W}$ 和 b 为可训练的参数, σ 为 Softmax 函数.

随后引入标签感知的不平衡损失函数对模型进行优化, 该损失函数具体来说分为两部分, 分别是标签分布感知损失函数和自适应类级重加权.

3.5.1 标签分布感知损失函数

本文引入标签分布感知的损失函数 LDAM[19]来对模型进行优化. 此损失函数具有标签分布感知的特点, 以一个节点类型为 y 的节点表示 $\mathbf{h}_v^{(\text{final})}$ 为例, 其损失的具体计算方式如公式 (13) 所示:

$$\mathcal{L}_{\text{LDAM}}((\mathbf{h}_v^{(\text{final})}, y); f_\Theta) = -\log \frac{\mathrm{e}^{p_y - \Delta_y}}{\mathrm{e}^{p_y - \Delta_y} + \sum_{j \neq y} \mathrm{e}^{z_j}} \tag{13}$$

其中, f_Θ 代表分类器, $p_y = f_\Theta(\mathbf{h}_v^{(\text{final})})_y$ 表示分类器的输出 $\mathbf{p}$ 中第 y 个元素的值, 即节点类型为 y 的概率, $\Delta_j = \dfrac{C}{n_j^{1/4}}$, C 为超参, n_j 为类型为 j 的节点个数, 因本文为二分类问题, 因此 $j \in \{0,1\}$.

此损失函数根据每一类的节点个数来计算此类节点的间隔距离 Δ_j, 因此少数类 (恶意节点) 的 Δ_j 会比正常类大, 从上式不难看出, 此损失函数较为直观的作用是使少数类的表示距离分类器的决策超平面尽可能地远离, 从而防止模型向少数类过拟合.

3.5.2 自适应类级重加权

类级重加权 (class level re-weighting)[20]可以自适应地根据不同类型节点的数量调整训练期间不同类别的损失值, 以此来重新平衡类别. 具体来说, 其可以给恶意节点以较大的分类损失来有效地使不平衡的训练分布更接近均匀的测试分布. 此方法提出, 传统逆类频率重加权方法在数据极度不平衡的情况下会使模型优化变得困难, 引入有效样本数量的概念, 通过逆有效样本数量对损失重新加权, 具体如公式 (14) 和公式 (15) 所示:

$$weight_{[y]} = \frac{\dfrac{1-\beta}{1-\beta^{n_y}}}{\dfrac{1}{2}\sum_{j=0,1} \dfrac{1-\beta}{1-\beta^{n_j}}} \tag{14}$$

$$\mathcal{L}_{\text{RW}}(\mathbf{p}, y) = weight_{[y]} \mathcal{L}(\mathbf{p}, y) \tag{15}$$

其中, $\beta \in [0,1)$ 为超参, n_y 表示类型为 y 的节点的数量, $\dfrac{1-\beta}{1-\beta^{n_y}}$ 表示数据中类型为 y 的节点的有效样本数量的倒数, 本文对该值进行归一化使得两类节点的权重之和为 1. $\mathcal{L}(\mathbf{p}, y)$ 为预测概率分布为 $\mathbf{p}$, 真实标签为 y 的损失.

3.5.3 CAMD 全局损失函数

以上两种方法中, 重加权引入的权重只依赖于节点的类别信息, 而 LDAM 还依赖于模型的输出. 研究表明[19],

重加权与 LDAM 是互补的, 两者结合使用可能会起到更好的效果. 因此, 本文中将两者结合使用来优化模型参数, 以缓解数据不平衡对模型带来的影响, 具体如公式 (16) 和公式 (17) 所示:

$$\mathcal{L}_{\text{CAMD}} = \mathcal{L}_{\text{LDAM-RW}}\left(\left(\mathbf{h}_v^{(\text{final})}, y\right); f_\Theta\right) = weight_{[y]} \cdot \mathcal{L}_{\text{LDAM}}\left(\left(\mathbf{h}_v^{(\text{final})}, y\right); f_\Theta\right) \tag{16}$$

$$\mathcal{L}_{\text{CAMD}} = \frac{\dfrac{1-\beta}{1-\beta^{n_y}}}{\dfrac{1}{2}\displaystyle\sum_{j=0,1}\frac{1-\beta}{1-\beta^{n_j}}} \cdot \left(-\log\frac{\mathrm{e}^{p_y-\Delta_y}}{\mathrm{e}^{p_y-\Delta_y}+\displaystyle\sum_{j\neq y}\mathrm{e}^{z_j}}\right) \tag{17}$$

其中, y 为节点 v 的真实标签, f_Θ 代表分类器, $\beta \in [0,1)$ 为超参, n_y 表示类型为 y 的节点的数量, $\dfrac{1-\beta}{1-\beta^{n_y}}$ 表示数据中类型为 y 的节点的有效样本数量的倒数, $p_y = f_\Theta(\mathbf{h}_v^{(\text{final})})_y$ 表示分类器的输出 $\mathbf{p}$ 中第 y 个元素的值, 即节点类型为 y 的概率, $\Delta_j = \dfrac{C}{n_j^{1/4}}$, C 为超参, n_j 为类型为 j 的节点个数. CAMD 的损失函数为标签感知的不平衡损失函数, 该损失函数分为两部分, 分别是标签分布感知损失函数和自适应类级重加权. 标签分布感知损失函数旨在提高少数类的可区分性, 确保模型不会由于类别内样本数量的不均衡而偏向于某些类别, 有助于更准确地识别边界上的少数类别, 避免模型向某些类别过度拟合. 自适应类级重加权通过动态平衡机制, 确保模型在训练过程中持续关注难以分类的类别. 因此, 标签感知的不平衡损失函数能够在全局下有效处理不平衡数据集, 有助于提升模型整体性能.

4 标签稀缺情况下的恶意节点检测方法 CAMD$^+$

现实世界的恶意用户检测任务, 要检测的数据中大多是正常用户, 将很小一部分欺诈用户贴上“异常”的标签, 成本非常高. 对于恶意节点检测任务来说, 很少有数据被标记, 通常有大量的未标记数据. 标记数据稀缺将导致两个问题: (1) 模型在训练过程中获取的信息不足, 无法捕捉到恶意节点的真实分布和规律, 分类器会在少量的标记数据上过拟合, 编码器也无法学习到有区分度的节点表示, 从而使得整体模型泛化能力不足, 影响检测效果. (2) 类别感知注意力系数机制会因标记数据稀少而影响性能, 此机制基于多层感知机 MLP, 通过节点特征信息预测每个节点的类别, 在此基础上计算节点间的类型相似度并作为聚合时的注意力系数. 而该模块的训练需要用到充分的节点标签信息, 标记数据过少会导致 MLP 预测准确率大大降低, 进一步影响注意力系数的效果, 对于图上的信息聚合过程可能会起到负面影响. 因此, 结合恶意检测任务标记成本较高、标记数据稀缺的任务场景, 有必要构造一种方法, 使得在标签稀缺情况下也能保证检测效果. 基于此, 本文提出了 CAMD 在标签稀缺情况下的优化算法 CAMD$^+$.

该方法对于标签稀缺情况下, 真实标签信息对于指导模型训练远远不足的问题, 本文提出了如下解决方案.

如果直接从原始图 ϕ_1 学习节点表达, 则具有局部性. 使用图增强的方法可以为原始图 ϕ_1 生成一个全局视图 ϕ_2, 则同时得到原始图 ϕ_1 和一个全局视图 ϕ_2, 从而扩大了训练集, 提高模型的泛化能力和性能. 此外, 由于 CAMD 中的类别感知注意力系数在标签稀缺情况下无法很好地发挥作用, 在本章方法中将此机制去除, 因此本节中不一致图神经网络编码器的节点表示学习过程如公式 (18) 和公式 (19) 所示:

$$\mathbf{h}_v^{(l)} = \sum_{u_1 \in \mathcal{N}_1(v)} \mathbf{h}_{u_1}^{(l-1)} \,\|\, \sum_{u_2 \in \mathcal{N}_2(v)} \mathbf{h}_{u_2}^{(l-1)} \tag{18}$$

$$\mathbf{h}_v^{(\text{final})} = \mathbf{h}_v^{(0)} \,\|\, \mathbf{h}_v^{(1)} \,\|\, \ldots \,\|\, \mathbf{h}_v^{(L)} \tag{19}$$

其中, $\|$ 表示拼接操作. 使用 $\mathbf{h}_v^{(0)} = \sigma(W_e \mathbf{x}_v)$ 将节点原始特征映射到隐层空间, $W_e \in \mathbb{R}^{d_h \times d_x}$ 为可训练权重矩阵, d_h 为初始隐层向量维度, σ 为 ReLU 函数, $\mathbf{h}_v^{(\text{final})} \in \mathbb{R}^{(2^{L+1}-1)d_h}$ 为编码器输出的最终节点表示. 为原始图 ϕ_1 和一个全局视图 ϕ_2 分别构造一个不一致图神经网络编码器 g_{ϕ_1}, g_{ϕ_2}, 然后使用图对比学习的方式来对两个编码器进行训练. 采用自监督学习方式从未标记数据中挖掘信息, 从而弥补标签信息不足带来的问题, 同时引入类别感知图对比学习, 充分利用稀缺的标签信息, 从而增加节点的表示向量的区分度. 最终将两个编码器输出的表示向量 $\mathbf{h}^{(\text{final}),\phi_1}, \mathbf{h}^{(\text{final}),\phi_2}$ 进行求和, 如下式所示:

$$\mathbf{h}_v^{(\text{final})} = \lambda_1 \mathbf{h}_v^{(\text{final}),\phi_1} + \lambda_2 \mathbf{h}_v^{(\text{final}),\phi_2} \tag{20}$$

其中, λ_1 与 $\lambda_2 \in [0,1]$ 为超参, 用于调节局部视图信息与全局视图信息的比例. 将 $\mathbf{h}^{(\text{final})}$ 作为分类器 $f_\Theta : \mathbf{h}^{(\text{final})} \mapsto \mathbf{p}$ 的输入进行下游分类任务, 具体过程如公式 (21) 所示:

$$\mathbf{p} = \sigma\left(\mathbf{W}\mathbf{h}^{(\text{final})} + b\right) \tag{21}$$

其中, $\mathbf{W}$ 和 b 为可训练的参数, σ 为 Softmax 函数.

4.1 基于图扩散的数据增强

常用的图数据增强方法大致可以分为以下两种: (1) 节点特征变换: 通过噪声注入、特征掩码、特征置换等方式对初始节点特征进行特征空间增强; (2) 图拓扑变换: 通过添加或删除边、添加或删除节点、重采样边权等方式改变图的拓扑结构, 或使用最短距离或扩散矩阵生成全局视图. 节点特征变换可能会改变图的语义信息, 导致原本的标签不适用于新生成的样本, 并且其可能会引入一些噪声或者模糊, 降低模型的性能. 而图拓扑变换中的对边或节点的增加删除等操作可能会导致图数据原本的信息被破坏. 研究表明 [21], 大多数情况下, 通过图扩散卷积将邻接矩阵转换为扩散矩阵, 并将两个矩阵视为同一图结构的两个一致视图可以取得最佳结果. 基于这个思路, 本节中采用基于个性化 PageRank 的图扩散卷积模型, 根据原始图的邻接矩阵得到扩散矩阵, 具体计算方式如公式 (22) 所示:

$$\mathbf{S}^{\text{PPR}} = \alpha\left(\mathbf{I}_n - (1-\alpha)\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\right)^{-1} \tag{22}$$

其中, $\mathbf{A} \in \mathbb{R}^{n\times n}$ 表示原始图的邻接矩阵, $\alpha \in (0,1)$ 表示个性化 PageRank 中的传送概率, α 越小则表示越倾向于利用大邻域的信息, 而 α 越大则越倾向于保持局域性, $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ 表示单位矩阵, $\mathbf{D} \in \mathbb{R}^{n\times n}$ 为度矩阵, 对角线上的值为每个节点的度数.

与基于原始邻接矩阵的表示学习不同, 基于上述扩散矩阵进行图节点表示学习, 聚合过程中节点会从更大的邻域范围内获取信息, 因而学习到的节点表示可以反映图的全局结构和特征. 将扩散矩阵 $\mathbf{S}^{\text{PPR}}$ 作为原始图 ϕ_1 的全局视图 ϕ_2, 并在这两个视图上进行表示学习, 以同时挖掘图中的局部信息与全局信息.

4.2 自监督图对比学习

本文中原始图 ϕ_1 上学习到的节点表示具有局部性, 全局视图 ϕ_2 上学习的节点表示则反映图的全局结构和特征. 在这种情况下, 不同视图生成的表示可以相互补充, 从而丰富最终表示结果. 因而本节在原始图与全局视图上构建代理任务以训练不一致图神经网络编码器 g_{ϕ_1}, g_{ϕ_2}. 具体来说, 本文将不同视图上的同一节点视为一对正样本对, 例如 (v^{ϕ_1}, v^{ϕ_2}), 而该节点与另一视图上的其他节点则为负样本对例如 $(v^{\phi_1}, u^{\phi_2}), (u^{\phi_1}, v^{\phi_2})$. 如图 3 所示.

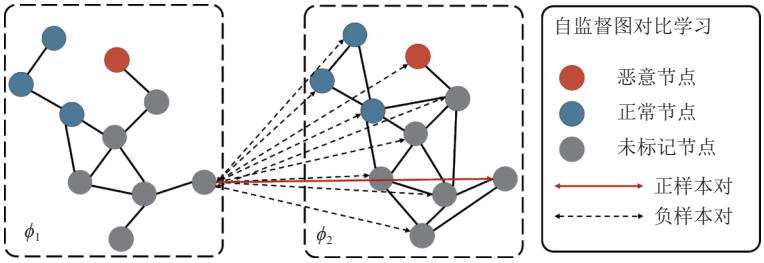


图 3 自监督图对比学习

通过最大化正样本对之间的互信息, 可以使编码器同时挖掘图中的局部信息与全局信息. 具体来说, 引入对比损失来进行训练, 具体方式如公式 (23) 所示:

$$\mathcal{L}_{\text{self}}(\mathbf{x}_i) = \mathcal{L}_{\text{self}}^{\phi_1}(\mathbf{x}_i) + \mathcal{L}_{\text{self}}^{\phi_2}(\mathbf{x}_i) \tag{23}$$

其中, n 为图中的节点数, $\mathcal{L}_{\text{self}}^{\phi_1}(\mathbf{x}_i)$ 和 $\mathcal{L}_{\text{self}}^{\phi_2}(\mathbf{x}_i)$ 为节点 i 在原始图和全局视图中的自监督图对比损失. 由于互信息很难被直接计算 [22], 通常会采用参数化互信息下界, 并抬高下界的方式以实现互信息最大化, 而本文基于 InfoNCE 互信息下界估计器 [22]来构造此对比损失, 则 $\mathcal{L}_{\text{self}}^{\phi_1}(\mathbf{x}_i)$ 具体如公式 (24) 所示:

$$\mathcal{L}_{\text{self}}^{\phi_1}(\mathbf{x}_i) = -\log \frac{\exp\left(\left\langle \mathbf{h}_i^{(\text{final}),\phi_1}, \mathbf{h}_i^{(\text{final}),\phi_2} \right\rangle\right)}{\sum_{j=1}^{n} \exp\left(\left\langle \mathbf{h}_i^{(\text{final}),\phi_1}, \mathbf{h}_j^{(\text{final}),\phi_2} \right\rangle\right)} \tag{24}$$

其中, $\mathbf{h}_i^{(\text{final}),\phi_1} = g_{\phi_1}(\mathbf{x}_i)$ 为节点 i 在原始图 ϕ_1 中编码得到的表示向量, $\mathbf{h}_i^{(\text{final}),\phi_2} = g_{\phi_2}(\mathbf{x}_i)$ 为节点 i 在全局视图 ϕ_2 中编码得到的表示向量, $\langle\cdot\rangle$ 表示向量内积, 用于衡量两个表示向量的相似度.

通过公式 (24), 正样本对 $(\mathbf{x}_i^{\phi_1}, \mathbf{x}_i^{\phi_2})$ 和负样本对 $(\mathbf{x}_i^{\phi_1}, \mathbf{x}_j^{\phi_2})$ 的表示向量相似度可以形成对比, 该式的目的是最大化正样本对相似度的同时最小化负样本对的相似度. 相应的, $\mathcal{L}_{\text{self}}^{\phi_2}(\mathbf{x}_i)$ 也是类似的计算方式, 具体如公式 (25) 所示:

$$\mathcal{L}_{\text{self}}^{\phi_2}(\mathbf{x}_i) = -\log \frac{\exp\left(\left\langle \mathbf{h}_i^{(\text{final}),\phi_2}, \mathbf{h}_i^{(\text{final}),\phi_1} \right\rangle\right)}{\sum_{j=1}^{n} \exp\left(\left\langle \mathbf{h}_i^{(\text{final}),\phi_2}, \mathbf{h}_j^{(\text{final}),\phi_1} \right\rangle\right)} \tag{25}$$

公式 (24) 和 (25) 分别为节点 i 在原始图和全局视图中的自监督图对比损失.

4.3 基于类别感知的平衡图对比学习

本节使用标签信息指导对比学习的训练过程, 使编码器更直接地捕获标签语义. 由于 CAMD 中的类别感知注意力系数在标签稀缺的任务场景中不再适用, 因而考虑使用此种方式来更直接地利用标签信息学习节点表示, 进一步增加节点的表示向量的区分度, 同时充分利用稀缺的标签信息. 具体来说, 在原始图 ϕ_1 与全局视图 ϕ_2 上基于标签语义构造正负样本对, 将两个视图中两个标签相同的节点视作一对正样本对, 例如 $(v^{\phi_1}, k^{\phi_2}), (k^{\phi_1}, v^{\phi_2})$, 其中 $y_v = y_k$, 标签不同的节点视作负样本对, 例如 $(v^{\phi_1}, j^{\phi_2}), (j^{\phi_1}, v^{\phi_2})$, 其中 $y_v \neq y_j$. 如图 4 所示.

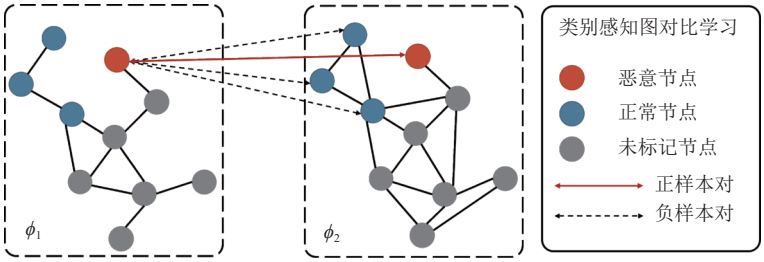


图 4 类别感知图对比学习

通过最大化正样本对的节点表示的相似度, 可以使编码器挖掘更多标签语义信息, 拉进相同类型节点表示间的距离, 拉大不同类型节点表示间的距离, 使表示向量更有区分度. 引入类别感知对比损失以进行训练, 本文中的具体方式如公式 (26) 所示:

$$\mathcal{L}_{\text{sup}}(\mathbf{x}_i) = \mathcal{L}_{\text{sup}}^{\phi_1}(\mathbf{x}_i) + \mathcal{L}_{\text{sup}}^{\phi_2}(\mathbf{x}_i) \tag{26}$$

其中, $\mathcal{L}_{\text{sup}}^{\phi_1}(\mathbf{x}_i)$ 和 $\mathcal{L}_{\text{sup}}^{\phi_2}(\mathbf{x}_i)$ 为节点在原始图和全局视图中的有监督对比损失, 同样基于 InfoNCE 互信息下界估计器来构造此对比损失. 由于类别感知对比学习也使用了标签信息, 研究表明数据不平衡也会影响对比学习的效果[23], 具体来说, 作为多数类的正常节点所贡献的梯度将比作为少数类的恶意节点大得多, 这不可避免地导致损失函数更多地专注于优化正常节点表示, 从而影响整体性能. 本文引入基于类平均的不平衡损失, 其对损失函数分母中表示向量的相似度之和按照类内节点个数求平均, 具体来说如公式 (27) 和公式 (28) 所示:

$$\mathcal{L}_{\text{sup}}^{\phi_1}(\mathbf{x}_i) = -\frac{1}{|B_{y_i}| - 1} \sum_{k=1}^{l} \log \frac{1_{[y_i = y_k]} \exp\left(\left|\mathbf{h}_i^{(\text{final}),\phi_1}, \mathbf{h}_k^{(\text{final}),\phi_2}\right|\right)}{\sum_{y \in \{0,1\}} \frac{1}{|B_y|} \sum_{j \in B_y} \exp\left(\left\langle \mathbf{h}_i^{(\text{final}),\phi_1}, \mathbf{h}_j^{(\text{final}),\phi_2} \right\rangle\right)} \tag{27}$$

$$\mathcal{L}_{\text{sup}}^{\phi_2}(\mathbf{x}_i) = -\frac{1}{|B_{y_i}| - 1} \sum_{k=1}^{l} \log \frac{1_{[y_i = y_k]} \exp\left(\left|\mathbf{h}_i^{(\text{final}),\phi_2}, \mathbf{h}_k^{(\text{final}),\phi_1}\right|\right)}{\sum_{y \in \{0,1\}} \frac{1}{|B_y|} \sum_{j \in B_y} \exp\left(\left\langle \mathbf{h}_i^{(\text{final}),\phi_2}, \mathbf{h}_j^{(\text{final}),\phi_1} \right\rangle\right)} \tag{28}$$

其中, B_{y_i} 表示训练集中所有标签为 y_i 的节点的集合, $|B_{y_i}|$ 则表示这些节点的个数, 节点 k 为标签与节点 i 相同的节

点, 分母中的 $B_y, y \in \{0,1\}$ 在本文中分别表示为标签为 0 和 1 的节点集合. 此机制平衡了不同类型节点贡献的梯度, 这样正常节点和恶意节点对模型的优化都有近似的贡献, 直观地说, 它减少了作为多数类的正常节点在分母中的比例, 并强调了作为少数类的恶意节点的重要性. 公式 (27) 和公式 (28) 分别为节点 i 在原始图和全局视图中的有监督图对比损失.

4.4 CAMD$^+$全局损失函数

CAMD$^+$采用联合训练的形式, 具体损失函数如下:

$$\mathcal{L}_{\text{CAMD}^+} = \mathcal{L}_{\text{LSA-cls}} + \lambda_{\text{sl}}\mathcal{L}_{\text{LSA-self}} + \lambda_{\text{sp}}\mathcal{L}_{\text{LSA-sup}} \tag{29}$$

其中, 分类损失 $\mathcal{L}_{\text{LSA-cls}}$ 沿用 CAMD 的类别感知不平衡损失函数, λ_{sl}, λ_{sp} 为超参, 用于调节自监督对比损失和类别感知对比损失的比重. 分类损失有助于处理不平衡的类别分布, 尤其是能够提高对少数类别的分类准确性. 自监督对比损失 $\mathcal{L}_{\text{LSA-self}}$ 可以让编码器从未标记数据中学习更多信息, 提高模型的泛化能力与鲁棒性. 有监督对比损失 $\mathcal{L}_{\text{LSA-sup}}$ 则利用标签信息指导对比学习的训练过程, 使编码器能更直接地捕获标签语义. 将 $\mathcal{L}_{\text{LSA-sup}}$ 和 $\mathcal{L}_{\text{LSA-self}}$ 相结合, 模型能够同时考虑标签信息和数据的内在结构信息, 从而更全面地捕获数据的语义信息. 此外, 分类损失中的类别感知不平衡损失函数和自适应级重加权机制有助于处理不平衡的类别分布, 特别是能够提升对少数类别分类的能力.

5 实验与分析

5.1 数据集

为验证本文所提出的 CAMD 与 CAMD$^+$算法的性能, 采用以下恶意节点检测领域的数据集, 数据集具体统计信息见表 1.

1) Amazon[7]: 包括 Amazon 电商平台乐器分类下进行产品评论的用户, 将评论有用投票 80% 以上的用户标记为良性用户, 有用投票少于 20% 的用户标记为恶意用户.

2) YelpChi[7]: 包括 Yelp 平台中对芝加哥地区的酒店和餐厅的评价, 将被平台推荐的评价标记为良性评价, 被平台过滤掉的评价标记为恶意评价. YelpChi 原本有 49 315 个节点, 由于该数据集规模较大, 训练时会存在显存不足的问题, 因此本文中使用 METIS 图划分算法对原本图数据进行划分, 划分后的节点数目为 10 893.

3) Wikipedia edits[6]: 维基百科为防止恶意修改会通过各种措施找到并封禁一些账号, 该数据集包含了一个月内用户编辑页面的信息, 用户包括被封禁用户和良性用户. 若两个用户编辑过同一个页面, 则在用户间构建连接关系.

4) Tencent-Weibo[24]: 包括腾讯微博平台上发布话题的用户, 此数据集假设一个用户在较短时间 (数秒) 内发布两条话题就是可疑事件, 如果一个用户至少发生了 5 起可疑事件则将被标记为可疑用户.

5) T-finance[25]: 包括金融交易平台中的交易账户, 包含手工标注的恶意账户以及账户之间的交易关系. 该数据集包含新出现的节点, 可以用于说明本文方法在动态恶意节点监测方面的效果.

5.2 评价指标与参数设置

由于本文的恶意节点检测数据存在不平衡的问题, 因此采用 ROC 的曲线下面积 ROC-AUC 作为评估指标, 将恶意节点作为正样本, 该指标的计算方式如公式 (30) 所示:

$$AUC = \frac{\sum_{u \in \mathcal{U}^+} rank_u - \dfrac{|\mathcal{U}^+| \times (|\mathcal{U}^+| + 1)}{2}}{|\mathcal{U}^+| \times |\mathcal{U}^-|} \tag{30}$$

其中, $\mathcal{U}^+$ 和 $\mathcal{U}^-$ 分别表示恶意节点与正常节点, $rank_u$ 表示节点 u 在预测得分中的排名. 因为 AUC 是基于所有实例的预测概率的相对排序来计算的, 对样本的正负样本比例情况不敏感, 即使正例与负例的比例发生了很大变化, 其值也不会产生大的变化, 这可以消除数据不平衡带来的影响.

模型中节点一致性度量模块中的多层感知机隐藏层数设置为 2, 每层 64 个单元, Dropout 率设置为 0.5, 使用 Adam 优化器来训练模型的参数, 学习率设置为 0.01, Weight Decay 设置为 0.001. 图神经网络编码器的节点中检测表示向量维度设置为 32, 卷积层数设置为 2. 验证 CAMD 方法有效性时, 类别感知注意力系数和原始图结构信

息的系数 λ_{att} 和 λ_{adj} 分别设置为 0.5 和 1. 使用 Python 的 Scikit-learn 库中的数据集划分方法, 在保证标签分布一致的情况下, 对 5 个数据集分别进行 10 次随机的划分, 训练集、验证集、测试集的比例分别为 40%、20%、40%, 验证 CAMD$^+$方法有效性时, 为了模拟标签稀缺场景, 比例分别为 10%、30%、60%, 原始图节点表示向量 λ_1 和全局视图节点表示向量 λ_2 系数分别设置为 1 和 0.7.

5.3 对比方法

(1) 仅使用节点特征信息: MLP.

(2) 传统 GNN 模型: GAT[26]、APPNP[27].

(3) 非同质 GNN 模型: MixHop[16]、H2GCN[17]、GPRGNN[28].

(4) 图对比学习模型: MVGRL[21].

(5) 基于图的恶意节点检测模型: CARE-GNN[7]、PC-GNN[8]、H2-FDetector[14]、GHRN[25].

5.4 恶意节点检测性能比较

5.4.1 CAMD 对比实验

对比模型和 CAMD 方法在恶意节点检测任务上的对比结果如表 2 所示. 本文以粗体突出显示每个数据集上的最佳分类效果, 从表中可以看出本文提出的方法 CAMD 表现要优于其他方法. 观察可以发现, Amazon、YelpChi、Wiki 等数据集仅使用 MLP 就可以接近甚至优于传统 GNN 方法的分类效果, 这说明由于数据不一致性的存在, 使用 GNN 模型时如果不采取相应的处理策略, 那么在 GNN 的消息传递过程中, 图中节点很可能聚合到错误的邻域信息, 从而导致学习到的节点表示区分性下降, 影响下游分类任务效果. 观察非同质 GNN 模型在几个数据集上的表现, 可以发现这些方法在大部分情况下优于传统 GNN, 这说明对于恶意节点检测中存在的数据不一致问题, 使用非同质图方法是有效的, 其常用的策略如获取高阶邻域信息, 以及使用中间层信息等可以有效地在节点表示学习过程中获取更多丰富的信息, 增强节点的表达能力.

表 2 CAMD 对比实验结果 (%)

Class	Method	Amazon	YelpChi	Wiki	Tencent-Weibo	T-finance
Features only	MLP	96.76	82.37	61.74	77.50	86.09
General GNNs	GAT	82.12	58.34	63.63	97.06	93.88
	APPNP	94.08	68.60	62.99	96.82	96.36
Heterophilious GNNs	GPRGNN	93.39	73.24	60.39	98.99	96.42
	MixHop	97.25	82.43	61.96	97.19	95.69
	H2GCN	96.79	84.06	62.73	97.78	95.53
Graph fraud detection	CARE-GNN	94.82	77.48	56.76	91.05	93.26
	PC-GNN	96.42	81.04	56.97	95.77	96.62
	H2-FDetector	96.94	84.60	57.25	94.18	97.15
	GHRN	97.02	84.44	63.03	94.93	97.23
	CAMD (Ours)	**98.21**	**85.31**	**65.89**	**99.22**	**98.04**

观察基于图的恶意节点检测方法在几个数据集上的表现, 首先发现这几个模型在 Amazon 和 YelpChi 数据集上的效果普遍优于传统 GNN 模型, 这说明相应的恶意检测策略是有效的. 其中 CARE-GNN 和 PC-GNN 基于邻居选择的思想解决数据不一致问题, 聚合过程中, 去除与自身不相似的邻居节点, 以防止节点特征由于恶意节点的伪装行为被混淆. 而 H2-FDetector 对于可能为不同类型的邻居节点, 聚合时注意力系数可以为负, 以此种方式防止节点信息被混淆, 同时引入了节点原型以拉近相同类型节点表示, 其效果要优于前两者. 而 GHRN 则是基于谱域视角, 将图中存在不一致连接的节点视为高频信号, 构造基于 Laplacian 图的高通滤波器以筛选出异常节点及与之相连的边, 并进行异常边删除, 其效果是这 4 个对比方法中最好的. 而本文方法 CAMD 同时考虑了不同类型邻居节点聚合时的注意力系数, 不一致图数据的表示学习, 以及数据不平衡问题, 检测效果优于其他模型.

在金融领域 T-finance 数据集上使用 MLP 进行预测时的效果比传统 GNN 方法差很多, 这说明在此数据集上节点间的交互关系包含着丰富的信息, 仅使用节点特征无法充分挖掘这些信息, 导致模型分类效果较差. 在

T-finance 数据集上, CARE-GNN, PC-GNN, H2-FDetector 以及 GHRN 等模型中的数据不一致处理策略反而会破坏图中原有的信息, 从而降低模型分类的效果, 而 CAMD 在为节点学习表达能力强的表示向量的同时, 并不会破坏图本身的信息. 因此, CAMD 对新领域的恶意检测数据具有良好的自适应性.

5.4.2 CAMD$^+$对比实验

对比模型和 CAMD$^+$方法在恶意节点检测任务上的对比结果如表 3 所示. 本文以粗体突出显示每个类别中的最佳结果, 从表中可以看出本节提出的方法在大部分数据集上表现要优于其他方法, 在 Tencent-Weibo 数据集上略低于本文第 3 节提出的方法 CAMD. 首先发现与第 5.4.1 节的实验结果相比, 这些方法在标签稀缺的情况下性能都会有所下降, 这是因为标记数据稀缺导致模型在训练过程中获取的信息不足, 无法捕捉到恶意节点的真实分布和规律.

表 3 CAMD$^+$对比实验结果 (%)

Class	Method	Amazon	YelpChi	Wiki	Tencent-Weibo	T-finance
General GNN	APPNP	93.70	64.88	56.03	96.78	95.99
Contrastive GNN	MVGRL	82.29	75.41	60.02	89.58	93.11
Heterophilious GNNs	GPRGNN	90.40	60.77	59.79	98.83	95.97
	MixHop	92.48	77.62	58.08	94.05	95.12
	H2GCN	94.12	78.01	58.81	97.23	95.24
Graph fraud detection	CARE-GNN	91.73	72.59	51.26	85.28	93.01
	PC-GNN	90.82	75.89	53.66	90.82	95.65
	H2-FDetector	93.78	79.39	52.58	93.34	95.51
	GHRN	93.92	76.80	58.51	91.22	96.22
	CAMD (Ours)	96.54	78.83	59.82	**99.20**	97.11
	CAMD$^+$ (Ours)	**96.59**	**80.36**	**61.04**	99.16	**97.40**

观察图对比学习方法 MVGRL, 此方法首先基于图扩散生成全局视图, 然后在两个视图上进行自监督图对比学习. 此方法在 Amazon、Tencent-Weibo 和 T-finance 上表现较差, 而在另外两个数据集上表现相对较好, 这可能是因为对于 YelpChi 和 Wiki 这种不一致程度较高的数据集, 使用原始视图和全局视图对比学习可以更有效的挖掘全局的隐含信息. 由于自监督图对比学习可能也会带来噪声, 而对于 Amazon 和 Tencent-Weibo 数据集, 这种噪声带来的负面影响可能比自监督挖掘到的有效信息带来的提升要大. 而该模型相对于 CAMD$^+$表现较差, 判断可能是最大化节点和另一视图的图表示这种 local-global 的对比学习方式, 对于恶意检测任务而言并不合适, 会带来更多的噪声.

观察非同质图方法, 发现大部分方法要优于或接近基于图的恶意节点检测方法, 这说明对于恶意节点检测中存在的数据不一致问题, 使用非同质图方法在标签稀缺的情况下也是有效的, 而这些方法在 Wiki 数据集上效果都差于 MVGRL, 进一步说明对于 Wiki 数据集, 使用对比学习挖掘图内信息更能提升模型效果.

观察除本文 CAMD 方法以外的基于图的恶意节点检测方法, 可以看出这些方法效果下降较为明显, 这可能是因为这些方法都不同程度地基于标签信息对图原始结构做出了改变, 而在标签稀缺时, 这种方式会导致对图原有信息的破坏, 从而导致模型效果下降. 可以看出 CAMD$^+$方法在 Tencent-Weibo 数据集上略差于 CAMD 方法, 这两种方法都使用不一致图神经网络编码器, 而在此数据集上仅使用较少的标签就可以训练得到较好的注意力系数, 并对聚合过程起到有效的指引作用, 从而在标签稀缺的情况下也可以有较好的分类效果. CAMD$^+$方法在金融领域 T-finance 数据集上的表现优于其他方法. 在标签稀缺情况下, 相比于表 2 中的实验结果, 大多数对比模型在 T-finance 数据集上性能有所下降. 这是因为标记数据的稀缺导致模型在训练过程中获取的信息不足, 无法充分捕捉到节点的分布信息. 相比其他方法, CAMD$^+$在标签稀缺时表现依旧较为优越. 同时, 本文提出的 CAMD 方法在 T-finance 数据集标签稀缺的情况下, 实验结果也优于其他对比模型.

5.5 消融实验

本节进行消融实验以验证本文提出方法各机制的有效性, 分别对 CAMD 方法与 CAMD$^+$方法进行消融实验.

5.5.1 CAMD 消融实验

各消融项如下: CAMD\$_{\text{Att}}$ 去除类别感知注意力系数; CAMD\$_{\text{Comb}}$ 去除中间层组合机制; CAMD\$_{\text{Att\&Comb}}$ 同时去除类别感知注意力系数与中间层组合机制; CAMD\$_{\text{RW}}$ 去除不平衡损失机制中的重加权机制; CAMD\$_{\text{LDAM-RW}}$ 去除类别感知不平衡损失机制. 消融实验结果如表 4 所示.

表 4 消融实验结果 (%)

Method	Amazon	YelpChi	Wiki	Tencent-Weibo	T-finance
CAMD	**98.21**	**85.31**	**65.89**	**99.22**	**98.04**
CAMD\$_{\text{Comb}}$	96.58	75.03	64.56	99.12	97.63
CAMD\$_{\text{Att}}$	97.94	83.78	65.37	98.95	97.41
CAMD\$_{\text{Att\&Comb}}$	83.20	64.62	62.55	99.02	97.17
CAMD\$_{\text{RW}}$	97.79	84.47	65.18	97.52	97.77
CAMD\$_{\text{LDAM-RW}}$	97.88	85.01	65.06	97.21	96.80

首先通过观察 CAMD\$_{\text{LDAM-RW}}$ 发现, 去除不平衡损失函数后各数据集上的分类 *AUC* 分数都有所下降, 这意味着不平衡损失函数对于提升恶意节点检测任务的效果有所帮助. 此外在不一致程度较高的 Amazon 与 YelpChi 数据集上, 去除中间层组合机制与类别感知注意力系数后 *AUC* 分数分别都有少量的下降, 而同时去除这两种机制则下降较多, 这首先说明本文提出的不一致图编码器在此类数据集上体现了较大的作用, 且中间层组合机制与类别感知注意力系数这两种机制可以互相弥补另一方缺失带来的不足. 而在 Wiki、Tencent-Weibo、T-finance 数据集上, 分别观察 CAMD\$_{\text{Att}}$, CAMD\$_{\text{Comb}}$ 与 CAMD\$_{\text{Att\&Comb}}$ 发现, 不一致图编码器起到了一定的作用, 但提升与 Amazon、YelpChi 相比没有那么显著, 这可能是因为 Wiki 数据集相邻节点间的特征相似性和标签相似性都较低, 导致即使使用中间层组合机制也无法获取充足的信息, 同时依靠特征信息对节点类型进行判断也较为不充分, 使用类别感知注意力系数带来的提升有限. 而 Weibo 和 T-finance 数据集本身的不一致程度并不高, 使用传统 GNN 即可较为有效地进行恶意节点检测.

5.5.2 CAMD^{+}消融实验

各消融项如下所示: CAMD^{+}\$_{\text{Sup}}$ 去除类别感知对比学习模块; CAMD^{+}\$_{\text{Self}}$ 去除自监督对比学习模块; CAMD^{+}\$_{\text{Bal-CL}}$ 去除基于类平均的不平衡对比损失; CAMD^{+}\$_{\text{Bal}}$ 去除不平衡对比损失以及分类不平衡损失 $\mathcal{L}_{\text{LDAM-RW}}$. 消融实验结果如表 5 所示.

表 5 消融实验结果 (%)

Method	Amazon	YelpChi	Wiki	Tencent-Weibo	T-finance
CAMD^{+}	96.59	**80.36**	**61.04**	99.16	**97.40**
CAMD^{+}\$_{\text{Sup}}$	96.42	80.02	58.93	98.72	85.49
CAMD^{+}\$_{\text{Self}}$	**96.78**	79.89	58.12	**99.23**	97.16
CAMD^{+}\$_{\text{Bal-CL}}$	95.63	79.05	59.01	99.09	97.24
CAMD^{+}\$_{\text{Bal}}$	95.68	79.57	58.20	97.48	96.30

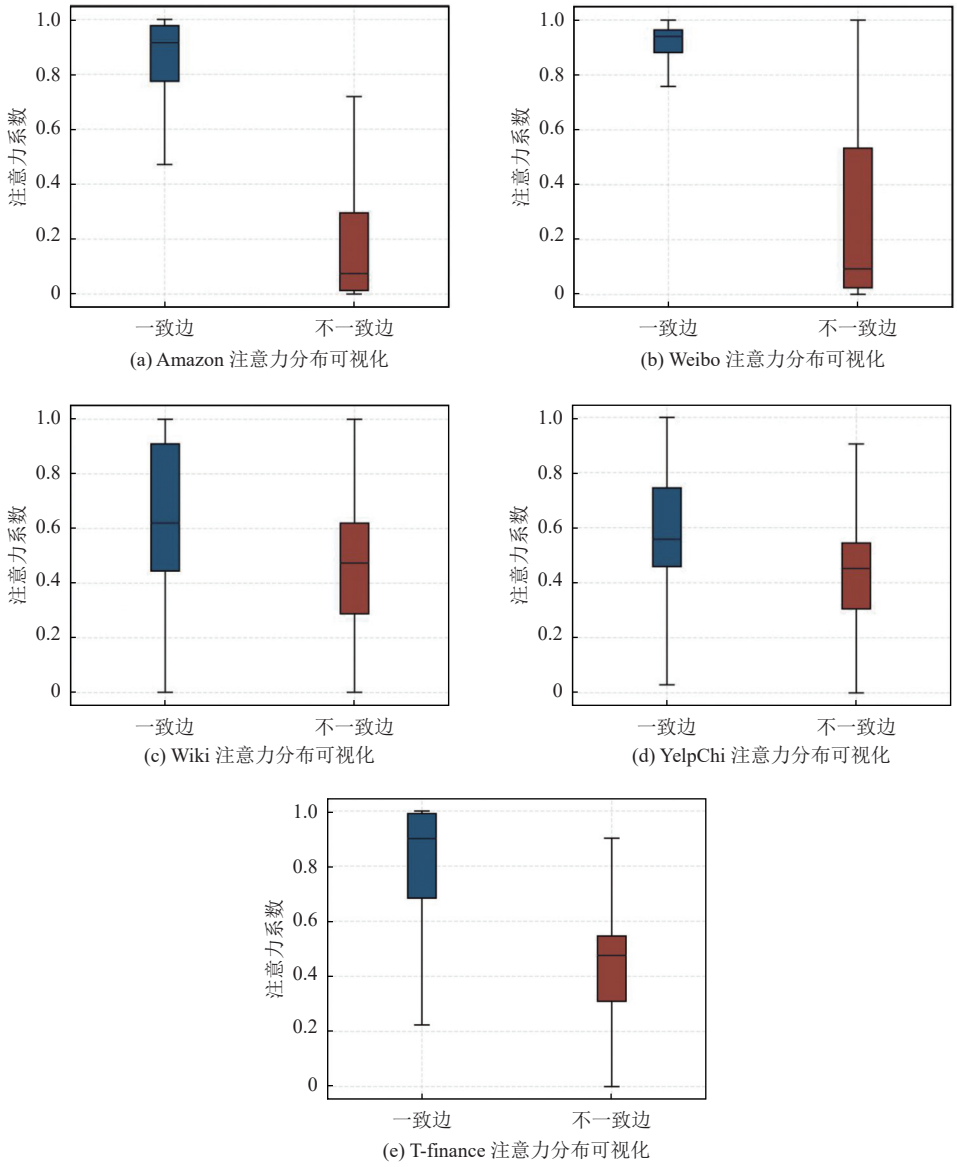
首先观察 CAMD^{+}\$_{\text{Self}}$ 发现, Amazon 和 Tencent-Weibo 数据集上的分类 *AUC* 分数都比原始模型有了些许提高, 可能的原因是自监督图对比学习的过程中可能会捕获冗余的图信息, 给节点表示向量带来噪声, 而这两个数据集中的节点表示可能受噪声信息的影响较大, 因而使用自监督会使模型效果下降. 此外, 观察 CAMD^{+}\$_{\text{Sup}}$ 发现使用标签感知对比学习对提升模型效果是有用的, 说明在这两个数据集上使用标签信息指导训练可以学习更有区分度的节点表示, 从而提升模型效果. 而在 YelpChi 与 Wiki 数据集中, 去除自监督对比学习带来的影响要稍大于去除类别感知对比学习的影响, 这可能是因为在这两个数据集上通过标签信息拉大不同节点表示向量的区分度带来的提升有限, 相比之下通过自监督图对比学习可以更有效地挖掘图中包含的信息, 使得编码器可以编码得到有益于下游节点分类任务的节点表示向量. 在 T-finance 数据集中, 观察 CAMD^{+}\$_{\text{Sup}}$ 可以发现, 去除类别感知对比学习模块模型性能受影响较大, 说明在此数据集上使用标签信息可以获得较大提升. 这可能是因为在此数据集上特征

相似度与标签相似度都较高, 所以更需要标签信息来用于区分不同类型的节点.

观察 CAMD^{+}\$_{Bal}$ 发现, 去除不平衡损失函数后各数据集的 *AUC* 分数都有所下降, 这意味着使用不平衡损失函数对于恶意检测任务有所帮助, 相对来说 Tencent-Weibo 受不平衡问题影响较大. 此外, 观察 CAMD^{+}\$_{Bal-CL}$ 发现, 在 Amazon 和 YelpChi 数据集上的 *AUC* 分数下降相对 CAMD^{+}\$_{Bal}$ 较多, 这说明基于类平均的不平衡对比损失在这两个数据集上相对更有效, 而在 Wiki、Tencent-Weibo、T-finance 数据集上, 两种损失函数配合的情况下效果最优.

5.6 类别感知注意力系数分析

本文的 CAMD 方法中提出为图中的节点对进行一致性度量, 并以此得到类别感知的注意力系数来指导邻域信息的聚合过程. 本节将对此机制的有效性进行进一步分析, 将每个数据集中的注意力系数在一致边 (两端是标签相同的节点) 和不一致边 (两端是标签不同的节点) 上的分布可视化, 如图 5 所示.



(a) Amazon 注意力分布可视化　(b) Weibo 注意力分布可视化

(c) Wiki 注意力分布可视化　(d) YelpChi 注意力分布可视化

(e) T-finance 注意力分布可视化

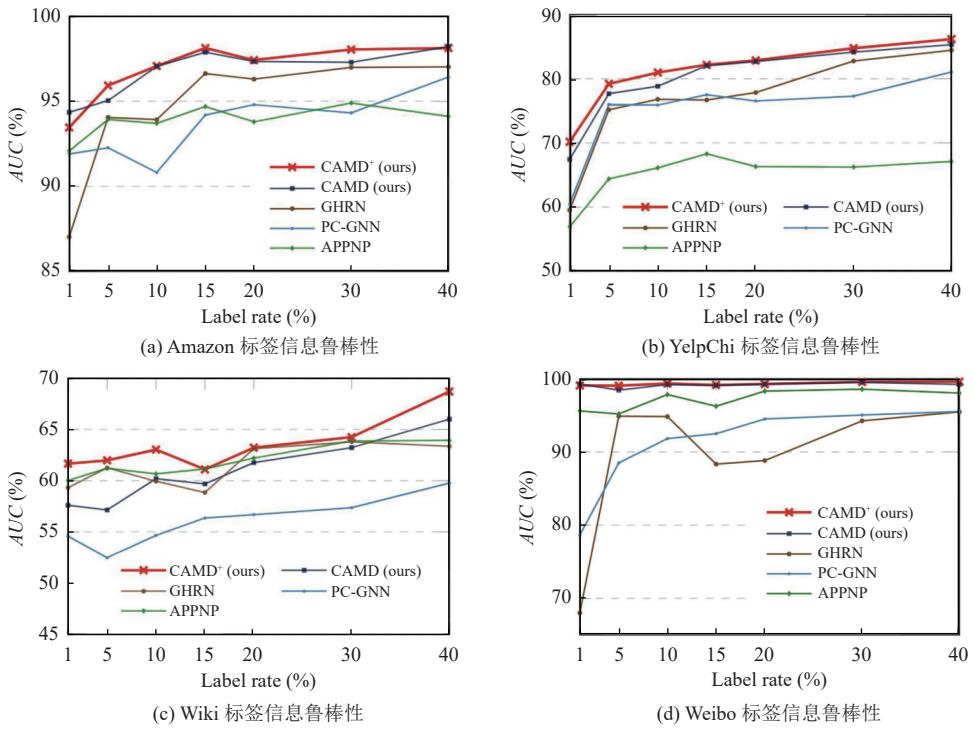
图 5　注意力系数分布可视化

首先发现在每个数据集上, 注意力系数都可以自适应的在一致边上学习到更大的权重, 而在不一致边上学习较低的权重. 因此在聚合邻域信息的过程中, 可以聚合到更多同类节点的信息, 而减少不同类型节点信息的聚合. 不同于部分恶意节点检测方法, 通过对图中的边进行裁剪或添加来解决数据不一致问题, 这种机制可能会破坏图原本的信息, 从而导致在不一致性程度低数据集上起到负作用. 本文提出的注意力系数不改变图结构, 在数据不一致程度高的数据集 (Amazon) 和数据不一致性程度低的数据集 (Tencent-Weibo) 都可以有效提升模型效果.

此外, 可以发现注意力系数在 Amazon、Tencent-Weibo 和 T-finance 上对不同类型的边的权重分布差别较明显, 而在 YelpChi 和 Wiki 数据集上并没有拉开太大的差距, 通过对两个数据集的信息进行分析, 判断这可能是由于在这两个数据集上, 仅依靠原始特征信息来判断节点的类型属性是相对不充分的, 与此同时由于不一致程度过高, 在图神经网络上的学习过程也无法给予注意力系数较为充分的类别信息. 通过观察第 5.4.1 节中 MLP 模型在各数据集上的分类 *AUC* 分数也可看出, 虽然 YelpChi、Wiki、Tencent-Weibo 在仅使用特征信息进行分类时效果都较为一般, 但由于 Tencent-Weibo 的图数据中包含着丰富的信息且数据不一致性较低, 也可以在训练时给注意力系数相应的信息. 因此, 在 YelpChi 这类数据集上, 使 λ_{adj} 较大, λ_{att} 较小, 使中间层组合机制发挥主要作用, 注意力系数发挥辅助作用可能会更有效地提升模型效果.

5.7 标签信息鲁棒性实验

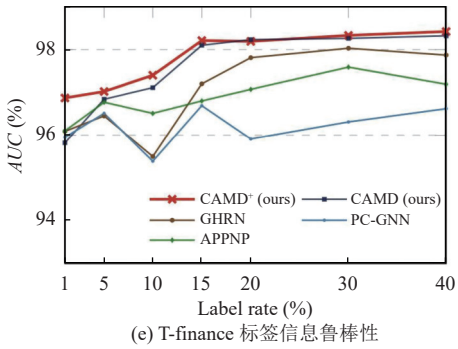
为了验证本文中 CAMD$^+$方法对于不同程度标签稀缺恶意节点检测任务场景的自适应效果, 引入本节标签信息鲁棒性实验. 其中为了更细致地观察本方法在低标签比例情况下的表现与鲁棒性, 对于 1%–20% 之间每 5% 设置一个观测点, 对于 20%–40% 之间每 10% 设置一个观测点. 综上所述, 在训练集占比分别为 1%, 5%, 10%, 15%, 20%, 30%, 40% 的数据集上进行实验, 观察不同模型在 5 个数据集上的表现, 本节中的对比模型为本文第 3 节模型 CAMD, 两个表现较好的基于图神经网络的恶意节点检测模型 GHRN 和 PC-GNN, 以及与 CAMD$^+$全局视图思想相似的 APPNP 模型, 实验中验证集和测试集在剩余数据中占比为 1:2, 具体实验结果如图 6 所示.



(a) Amazon 标签信息鲁棒性　(b) YelpChi 标签信息鲁棒性

(c) Wiki 标签信息鲁棒性　(d) Weibo 标签信息鲁棒性

图 6　标签信息鲁棒性可视化



(e) T-finance 标签信息鲁棒性

图 6 标签信息鲁棒性可视化 (续)

通过观察以上结果可以看出, 本文中的 CAMD$^+$方法在不同程度的标签稀缺任务场景下都可以取得较好的结果. PC-GNN 会根据标签信息训练计算节点表示相似度并删除一些边, 而在标签稀缺情况下节点表示的学习与相似度计算很可能都是不准确的, 因而会导致边的误删, 因此 PC-GNN 的鲁棒性较差, 其分类 *AUC* 分数随标签比例变化波动较大. GHRN 也是类似的机制, 所以其 *AUC* 分数随标签比例变化, 存在较大的波动. 而 APPNP 在几个数据集上波动相较前两者而言较为平稳, 这说明在标签稀缺的情况下, 获取更远邻域的信息可以有效提升模型鲁棒性. 然而由于数据不一致性和数据不平衡问题的存在, APPNP 的整体分类效果在 Amazon 和 YelpChi 数据集低于前两个恶意节点检测模型.

5.8 训练效率分析

为了全面验证本文中提出的解决恶意节点检测任务的两种不同方法 CAMD 与 CAMD$^+$各自的优劣, 本文不仅在分类的 *AUC* 分数, 标签鲁棒性上进行对比, 还在不同数据集上的模型平均训练时间上进行对比. 具体结果如表 6 所示, 表中数据为训练中每迭代一次所花费的时间, 单位为毫秒 (ms).

表 6 每 epoch 平均训练时间 (ms)

Method	Amazon	YelpChi	Wiki	Tencent-Weibo	T-finance
CAMD	**120.370**	**92.134**	**79.561**	**63.529**	**83.280**
CAMD$^+$	684.591	418.946	447.592	298.955	440.168
CARE-GNN	894.965	4 897.303	151.425	657.535	4 322.319
PC-GNN	1 661.053	7 849.918	99.032	1 031.240	15 683.106
H2-FDetector	79 229.423	77 033.152	806.236	42 246.556	137 156.230
GHRN	908.117	336.880	102.575	118.705	190.355

结合表 2、表 3 和表 6 可知, CAMD 在实现了次优结果的同时, 平均训练时间最短, 而 CAMD$^+$在实现了最优效果的同时, 其训练效率在大多数数据集上仅次于 CAMD 和 GHRN. H2-FDetector 模型训练耗时最长, 主要原因在于其需要先预测边类型, 之后使用注意力机制聚合信息并计算不同类别的原型, 整体耗费时间较长. 而 CARE-GNN 与 PC-GNN 都包含了节点的选择机制, 模型在此步骤花费了较长时间. 相比之下, GHRN 采用了基于谱域视角构造高通滤波器, 以筛选出异常边并进行删除, 模型训练效率最高. 基于对比学习的 CAMD$^+$训练时间远高于基于类别感知的 CAMD 模型, 这是因为 CAMD$^+$方法会在原始图 ϕ_1 与全局视图 ϕ_2 两个图上分别使用两个图神经网络编码器 g_{ϕ_1} 和 g_{ϕ_2} 独立地进行节点表示学习, 此外为了充分挖掘图中信息, CAMD$^+$同时使用两种对比学习方式以训练编码器, 这一过程会带来更多的训练开销. 通过不同标签稀缺程度下的分类 *AUC* 分数的对比以及平均训练时间的对比, 可以看出 CAMD 模型与 CAMD$^+$模型的在不同维度的考量上各有其优势.

6 总结与展望

本文提出了一种基于自监督图对比表示学习的恶意节点检测方法, 该方法将恶意用户检测任务建模为基于图

神经网络的节点分类任务. 首先, 针对数据不一致问题以及数据不平衡问题, 提出基于类别感知的恶意节点检测方法 CAMD. CAMD 通过引入类别感知注意力系数、不一致图神经网络编码器以及类别感知不平衡损失函数, 增加了不同类型节点表示的区分度, 保留了节点原本的信息以及表示学习过程中不同局域性的信息, 增加节点表示的表达能力. 接下来, 针对 CAMD 在标签稀缺情况下效果受限的问题, 提出了基于图对比学习的方法 CAMD$^+$, 引入数据增强、自监督图对比学习以及基于类别感知的平衡图对比学习, 使模型在标签稀缺情况下取得良好的效果. 在 5 个真实世界数据集上的大量实验表明, 本文提出的 CAMD 与 CAMD$^+$方法的恶意节点检测性能由于其他基线方法. 未来的工作包括进一步探索数据不一致场景下, 基于图的标签传播方法对模型的效果提升, 以及考虑采用类似 GraphSAGE 的归纳式学习方法以减少计算内存占用, 从而使模型适用于更大规模的图数据.

References:

[1] Liu ZQ, Chen CC, Yang XX, Zhou J, Li XL, Song L. Heterogeneous graph neural networks for malicious account detection. In: Proc. of the 27th ACM Int'l Conf. on Information and Knowledge Management. Torino: ACM, 2018. 2077–2085. [doi: 10.1145/3269206.3272010]

[2] Liu ZW, Dou YT, Yu PS, Deng YT, Peng H. Alleviating the inconsistency problem of applying graph neural network to fraud detection. In: Proc. of the 43rd Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. ACM, 2020. 1569–1572. [doi: 10.1145/3397271.3401253]

[3] Liang C, Liu ZQ, Liu B, Zhou J, Li XL, Yang S, Qi Y. Uncovering insurance fraud conspiracy with network learning. In: Proc. of the 42nd Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. Paris: ACM, 2019. 1181–1184. [doi: 10.1145/3331184.3331372]

[4] Liu ZQ, Chen CC, Li LF, Zhou J, Li XL, Song L, Qi Y. GeniePath: Graph neural networks with adaptive receptive paths. In: Proc. of the 33rd AAAI Conf. on Artificial Intelligence. Honolulu: AAAI, 2019. 4424–4431. [doi: 10.1609/aaai.v33i01.33014424]

[5] Wang DX, Lin JB, Cui P, Jia QH, Wang Z, Fang YM, Yu Q, Zhou J, Yang S, Qi Y. A semi-supervised graph attentive network for financial fraud detection. In: Proc. of the 2019 IEEE Int'l Conf. on Data Mining (ICDM). Beijing: IEEE, 2019. 598–607. [doi: 10.1109/ICDM.2019.00070]

[6] Kumar S, Zhang XK, Leskovec J. Predicting dynamic embedding trajectory in temporal interaction networks. In: Proc. of the 25th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining. Anchorage: ACM, 2019. 1269–1278. [doi: 10.1145/3292500.3330895]

[7] Dou YT, Liu ZW, Sun L, Deng YT, Peng H, Yu PS. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In: Proc. of the 29th ACM Int'l Conf. on Information & Knowledge Management. ACM, 2020. 315–324. [doi: 10.1145/3340531.3411903]

[8] Liu Y, Ao X, Qin ZD, Chi JF, Feng JH, Yang H, He Q. Pick and choose: A GNN-based imbalanced learning approach for fraud detection. In: Proc. of the 2021 Web Conf. Ljubljana: ACM, 2021. 3168–3177. [doi: 10.1145/3442381.3449989]

[9] Wang YL, Zhang J, Guo SS, Yin HZ, Li CP, Chen H. Decoupling representation learning and classification for GNN-based anomaly detection. In: Proc. of the 44th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. ACM, 2021. 1239–1248. [doi: 10.1145/3404835.3462944]

[10] Veličković P, Fedus W, Hamilton WL, Liò P, Bengio Y, Hjelm RD. Deep graph infomax. In: Proc. of the 2019 Int'l Conf. on Learning Representations. New Orleans: OpenReview.net, 2019.

[11] Chen B, Zhang J, Zhang XK, Dong YX, Song J, Zhang P, Xu KB, Kharlamov E, Tang J. GCCAD: Graph contrastive coding for anomaly detection. IEEE Trans. on Knowledge and Data Engineering, 2023, 35(8): 8037–8051. [doi: 10.1109/TKDE.2022.3200459]

[12] Tang JH, Li JJ, Gao ZQ, Li J. Rethinking graph neural networks for anomaly detection. In: Proc. of the 39th Int'l Conf. on Machine Learning. Baltimore: PMLR, 2022. 21076–21089.

[13] Chai ZW, You SQ, Yang Y, Pu SL, Xu JR, Cai HY, Jiang WH. Can abnormality be detected by graph neural networks? In: Proc. of the 31st Int'l Joint Conf. on Artificial Intelligence. Vienna, 2022. 1945–1951. [doi: 10.24963/ijcai.2022/267]

[14] Shi FZ, Cao YN, Shang YM, Zhou YC, Zhou C, Wu J. H2-FDetector: A GNN-based fraud detector with homophilic and heterophilic connections. In: Proc. of the 2022 ACM Web Conf. ACM, 2022. 1486–1494. [doi: 10.1145/3485447.3512195]

[15] Lim D, Hohne F, Li XY, Huang SL, Gupta V, Bhalerao O, Lim SN. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. In: Proc. of the 35th Conf. on Neural Information Processing Systems (NeurIPS 2021). Montreal, 2021. 1–16.

[16] Abu-El-Haija S, Perozzi B, Kapoor A, Alipourfard N, Lerman K, Harutyunyan H, Steeg GV, Galstyan A. MixHop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In: Proc. of the 36th Int'l Conf. on Machine Learning. Long Beach:

PMLR, 2019. 21–29.

[17] Zhu J, Yan YJ, Zhao LX, Heimann M, Akoglu L, Koutra D. Beyond homophily in graph neural networks: Current limitations and effective designs. In: Proc. of the 34th Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 7793–7804.

[18] Xu K, Li CT, Tian YL, Sonobe T, Kawarabayashi K, Jegelka S. Representation learning on graphs with jumping knowledge networks. In: Proc. of the 35th Int'l Conf. on Machine Learning. Stockholm: PMLR, 2018. 5453–5462.

[19] Cao KD, Wei C, Gaidon A, Arechiga N, Ma TY. Learning imbalanced datasets with label-distribution-aware margin loss. In: Proc. of the 33rd Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2019. 1567–1578.

[20] Cui Y, Jia ML, Lin TY, Song Y, Belongie S. Class-balanced loss based on effective number of samples. In: Proc. of the 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 9260–9269. [doi: 10.1109/CVPR.2019.00949]

[21] Hassani K, Khasahmadi AH. Contrastive multi-view representation learning on graphs. In: Proc. of the 37th Int'l Conf. on Machine Learning. Online: JMLR.org, 2020. 4116–4126.

[22] Zhang CS, Chen J, Li QL, Deng BQ, Wang J, Chen CG. Deep contrastive learning: A survey. Acta Automatica Sinica, 2023, 49(1): 15–39 (in Chinese with English abstract). [doi: 10.16383/j.aas.c220421]

[23] Zhu JG, Wang Z, Chen JJ, Chen YPP, Jiang YG. Balanced contrastive learning for long-tailed visual recognition. In: Proc. of the 2022 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. New Orleans: IEEE, 2022. 6908–6917. [doi: 10.1109/CVPR52688.2022.00678]

[24] Jiang M. Catching social media advertisers with strategy analysis. In: Proc. of the 1st Int'l Workshop on Computational Methods for CyberSafety. Indianapolis: ACM, 2016. 5–10. [doi: 10.1145/3002137.3002143]

[25] Gao Y, Wang X, He XN, Liu ZG, Feng HM, Zhang YD. Addressing heterophily in graph anomaly detection: A perspective of graph spectrum. In: Proc. of the 2023 Web Conf. Austin: ACM, 2023. 1528–1538. [doi: 10.1145/3543507.3583268]

[26] 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.

[27] Gasteiger J, Bojchevski A, Günnemann S. Predict then propagate: Graph neural networks meet personalized PageRank. In: Proc. of the 2019 Int'l Conf. on Learning Representations. New Orleans: OpenReview.net, 2019.

[28] Chien E, Peng JH, Li P, Milenkovic O. Adaptive universal generalized PageRank graph neural network. In: Proc. of the 2021 Int'l Conf. on Learning Representations. Vienna: OpenReview.net, 2021.

附中文参考文献:

[22] 张重生, 陈杰, 李岐龙, 邓斌权, 王杰, 陈承功. 深度对比学习综述. 自动化学报, 2023, 49(1): 15–39. [doi: 10.16383/j.aas.c220421]

王晨旭(1986－), 男, 博士, 副教授, CCF 高级会员, 主要研究领域为网络数据挖掘与网络安全, 数据安全, 区块链.

王梦勤(2000－), 女, 硕士生, 主要研究领域为财务欺诈检测.

王凯月(1995－), 女, 硕士, 主要研究领域为图神经网络, 恶意节点检测.