·网络空间安全·　　文章编号：1000-3428(2022)10-0176-08　　文献标志码：A　　中图分类号：TP18

基于改进投影梯度下降算法的图卷积网络投毒攻击

金柯君[1]，于洪涛[1]，吴翼腾[1]，李邵梅[1]，操晓春[2]

（1.中国人民解放军战略支援部队信息工程大学 信息技术研究所，郑州 450000；
2.中国科学院信息工程研究所 信息安全国家重点实验室，北京 100093）

摘　要：图神经网络在面对节点分类、链路预测、社区检测等与图数据处理相关的任务时，容易受到对抗性攻击的安全威胁。基于梯度的攻击方法具有有效性和高效性，被广泛应用于图神经网络对抗性攻击，高效利用攻击梯度信息与求取离散条件下的攻击梯度是攻击离散图数据的关键。提出基于改进投影梯度下降算法的投毒攻击方法。将模型训练参数看作与扰动相关的函数，而非固定的常数，在模型的对抗训练中考虑了扰动矩阵的影响，同时在更新攻击样本时研究模型对抗训练的作用，实现数据投毒与对抗训练两个阶段的结合。采用投影梯度下降算法对变量实施扰动，并将其转化为二进制，以高效利用攻击梯度信息，从而解决贪婪算法中时间开销随扰动比例线性增加的问题。实验结果表明，当扰动比例为5%时，相比Random、DICE、Min-max攻击方法，在Citeseer、Cora、Cora_ml和Polblogs数据集上图卷积网络模型被该方法攻击后的分类准确率分别平均降低3.27%、3.06%、3.54%、9.07%，在时间开销和攻击效果之间实现了最佳平衡。

关键词：图卷积网络；对抗性攻击；投毒攻击；投影梯度下降；对抗训练

开放科学（资源服务）标志码（OSID）：

中文引用格式：金柯君，于洪涛，吴翼腾，等. 基于改进投影梯度下降算法的图卷积网络投毒攻击[J]. 计算机工程，2022，48(10)：176-183.

英文引用格式：JIN K J，YU H T，WU Y T，et al. Poisoning attack on graph convolutional network based on improved projection gradient descent algorithm[J]. Computer Engineering，2022，48(10)：176-183.

Poisoning Attack on Graph Convolutional Network Based on Improved Projection Gradient Descent Algorithm

JIN Kejun[1]，YU Hongtao[1]，WU Yiteng[1]，LI Shaomei[1]，CAO Xiaochun[2]

（1.Information Technology Research Institute，PLA Strategic Support Force Information Engineering University，Zhengzhou 450000，China；
2.State Key Laboratory of Information Security，Institute of Information Engineering，Chinese Academy of Sciences，Beijing 100093，China）

【Abstract】Graph neural network is widely used in graph data processing-related tasks，such as node classification，link prediction，and community detection. However，it is susceptible to security threats from adversarial attacks. Gradient-based attack methods are widely used in graph neural network adversary attacks because of their effectiveness and efficiency. The efficient use of attack gradient information and the acquisition of the attack gradient under discrete conditions are key to obtain attack discrete graph data. This study proposes a poisoning attack method based on an improved Projection Gradient Descent（PGD）algorithm. Using the model training parameters as a function related to disturbance instead of a fixed constant，the effect of disturbance is considered in the model adversarial training，and the effect of model adversarial training is considered when updating the attack samples to realize the combination of data poisoning and adversarial training. The PGD algorithm is used to perturb the variables and convert them into binary such that the attack gradient information can be used effectively to solve the linear increase in the time cost with the disturbance ratio in the greedy algorithm. Experimental results show that when the disturbance ratio is 5%，compared with the performances of Random，DICE，Min-max，and other attack methods，on Citeseer，Cora，Cora_ml，and Polblogs datasets，the classification accuracy of a Graph Convolutional Network（GCN）model attacked by the proposed method reduced by 3.27%，3.06%，3.54%，and 9.07% on average，respectively，demonstrating the best balance between time overhead and attack effect.

【Key words】Graph Convolutional Network（GCN）；adversarial attack；poisoning attack；Projection Gradient Descent（PGD）；adversarial training

DOI：10. 19678/j. issn. 1000-3428. 0063533

基金项目：国家自然科学基金创新研究群体项目（61521003）；郑州市协同创新重大专项（162/32410218）。

作者简介：金柯君（1993—），男，硕士研究生，主研方向为人工智能安全；于洪涛（通信作者），研究员、博士、博士生导师；吴翼腾，博士；李邵梅，副研究员、博士；操晓春，教授、博士。

收稿日期：2021-12-15　　**修回日期：**2022-02-18　　**E-mail：**yht_ndsc@126.com

0　概述

图数据具有强大的表达能力，已经成为数据挖掘和机器学习领域的重要研究对象。图神经网络是专门针对图数据设计的端到端的深度学习模型[1]，它可以对语义属性数据和图数据统一表达建模，从图数据中提取特征以完成许多图分析任务，例如，节点分类[2-3]、链路预测[4-5]、社区检测[6-7]等。研究表明，图神经网络很容易受到对抗性攻击的安全威胁[8]，攻击者只要对图神经网络的结构或属性特征施加微小的扰动，则会导致图神经网络模型误判并影响其在具体任务中的表现。文献[9]研究图神经网络的对抗性攻击问题。之后，图神经网络的对抗性攻击方法的研究逐渐受到研究人员的关注[10-12]，成为人工智能安全领域的研究热点。

对抗性攻击根据攻击阶段不同分为污染测试数据的逃逸攻击和污染训练数据的投毒攻击[13-15]。投毒攻击分为数据投毒和对抗训练[16]两个阶段，为双层优化问题[17]。在实际应用中，图神经网络模型受到投毒攻击后，则更新训练参数，在参数更新过程中均会受到扰动的影响。模型的每一轮训练都在“中毒”数据的基础上，而攻击者的每一轮训练也是基于上一轮模型训练参数，这是一个循环嵌套的优化过程，即模型会对攻击者做出对抗性训练，以尽可能地降低受攻击后的性能损失。从攻击者角度出发，在优化攻击方法中考虑模型的对抗训练过程，以提高攻击效果。传统的连续优化方法难以直接应用于扰动的离散数据。文献[9]针对指定目标，采用贪婪算法对连边或节点特征逐个扰动以攻击单个节点。文献[18]针对非指定目标，提出基于投影梯度下降（Projection Gradient Descent，PGD）算法的投毒攻击Min-max。

本文提出基于改进投影梯度下降算法的投毒攻击方法PGattack。结合数据投毒与对抗训练两个阶段，在投毒攻击场景下，将模型训练参数看作可重新训练的变量，而不是固定的常量，在更新扰动矩阵时考虑模型的对抗训练过程，同时在模型对抗训练过程中研究扰动矩阵的作用。在此基础上，将图的连边情况松弛为一个取值[0，1]的连续变量，采用投影梯度下降算法对其进行扰动后再转化为二进制，并对离散图数据实施有效扰动。

1　基本概念与相关工作

1.1　图与图神经网络

图定义为$G(V,E)$，其中$V=\{v_1,v_2,\cdots,v_N\}$表示节点数$|V|=N$的节点集合，$E=\{e_1,e_2,\cdots,e_M\}$表示连边集合，节点之间的相邻关系通过邻接矩阵$\boldsymbol{A}$表示。在无权无向图中，$\boldsymbol{A}=\{0,1\}^{N\times N}$，$\boldsymbol{A}^{\mathrm{T}}=\boldsymbol{A}$。当节点$v_i$和节点$v_j$存在直接连边时，$\boldsymbol{A}_{i,j}\neq 0$，否则$\boldsymbol{A}_{i,j}=0$。在特征图中每个节点有$n$维的特征向量，节点特征可用矩阵$\boldsymbol{X}=\{0,1\}^{N\times n}$表示。节点分类任务[19]根据图$G(V,E)$和有标签节点的信息训练节点分类模型，并利用该模型正确预测无标签节点的类别。模型的表达如式（1）所示：

$$\underset{N\times m}{\hat{\boldsymbol{Y}}}=\mathrm{softmax}(\underset{N\times N}{\boldsymbol{A}}\ \underset{N\times n}{\boldsymbol{X}}\ \underset{n\times m}{\boldsymbol{W}}) \tag{1}$$

其中：$\boldsymbol{A}$为邻接矩阵；$\boldsymbol{X}$为输入特征向量；$\boldsymbol{W}$为训练参数矩阵；$\hat{\boldsymbol{Y}}$为模型输出。

交叉熵损失函数如式（2）所示：

$$L_{\mathrm{train}}(\boldsymbol{A},\boldsymbol{X},\boldsymbol{Y};\ \boldsymbol{W})=-\mathrm{tr}[\boldsymbol{Y}^{\mathrm{T}}\ln[\hat{\boldsymbol{Y}}]] \tag{2}$$

其中：$\boldsymbol{Y}$为节点的标签。

图神经网络的学习过程如式（3）所示：

$$\boldsymbol{W}^{*}=\underset{\boldsymbol{W}}{\mathrm{argmin}}\ L_{\mathrm{train}}(\boldsymbol{A},\boldsymbol{X},\boldsymbol{Y};\ \boldsymbol{W}) \tag{3}$$

其中：$\boldsymbol{W}^{*}$为图神经网络模型训练参数。

1.2　相关工作

1.2.1　投毒攻击

本文以图卷积网络（Graph Convolutional Network，GCN）作为研究对象，研究节点分类模型中非指定目标数据的投毒攻击。非指定目标攻击通过施加对抗性扰动以影响模型的整体性能，导致测试集的预测准确率整体下降[19]。投毒攻击是针对训练阶段的一种攻击方式。攻击者将投毒样本注入到训练数据集中，图卷积网络对“中毒”数据重新训练后，造成测试数据集的准确率降低[20]。文献[9，17]基于GCN构建生成投毒攻击模型。根据上述定义，攻击方法可以统一概括为约束优化问题，如式（4）所示：

$$\begin{aligned}&\hat{\boldsymbol{A}}=\underset{\hat{\boldsymbol{A}}}{\mathrm{argmin}}\ L_{\mathrm{atk}}[\hat{\boldsymbol{A}},\boldsymbol{X},\boldsymbol{Y};\boldsymbol{W}^{*}]\\&\text{s.t. }\boldsymbol{W}^{*}=\underset{\boldsymbol{W}}{\mathrm{argmin}}\ L_{\mathrm{train}}[\hat{\boldsymbol{A}},\boldsymbol{X},\boldsymbol{Y};\boldsymbol{W}]\\&\left\|\hat{\boldsymbol{A}}-\boldsymbol{A}\right\|_0\leqslant\delta\end{aligned} \tag{4}$$

其中：$\hat{\boldsymbol{A}}$为加入扰动后图卷积网络的邻接矩阵；$\|\cdot\|_0$为矩阵中非0元素的个数；δ为扰动预算，即允许扰动的最大数。投毒攻击以降低训练值的损失函数L_{atk}为目标，当L_{atk}越小时，攻击效果越好。本文令$L_{\mathrm{atk}}=-L_{\mathrm{train}}$。投毒攻击允许对参数进行重新训练，从式（4）可以看出，投毒攻击属于双层优化问题。

1.2.2　基于投影梯度下降算法的投毒攻击方法

文献[18]提出利用投影梯度下降算法对图数据实施攻击的方法，该方法能有效处理离散图数据，并提出一阶攻击生成框架的2种攻击场景：1）逃逸攻击，攻击一个预定义的图神经网络；2）投毒攻击，攻击一个可再训练的图神经网络。

针对第1种攻击场景，文献[18]采用投影梯度下降算法对邻接矩阵$\hat{\boldsymbol{A}}$进行扰动，攻击模型的表示如式（5）所示：

$$\hat{\boldsymbol{A}}=\underset{\hat{\boldsymbol{A}}}{\operatorname{argmin}} L_{\text{atk}}[\hat{\boldsymbol{A}},\boldsymbol{X},\boldsymbol{Y};\boldsymbol{W}]$$
$$\text{s.t.}\left\|\hat{\boldsymbol{A}}-\boldsymbol{A}\right\|_0\leqslant\delta \tag{5}$$

针对第2种可再训练模型参数的场景，文献[18]对模型参数$\boldsymbol{W}$进行优化。GCN模型以扰动损失大的方向作为优化目标，GCN攻击模型则以扰动损失减小的方向作为优化目标。攻击生成问题转变成最小最大化的问题，如式(6)所示：

$$(\hat{\boldsymbol{A}},\boldsymbol{W})=\underset{\hat{\boldsymbol{A}}}{\operatorname{argmin}}\ \underset{\boldsymbol{W}}{\operatorname{argmax}} L_{\text{atk}}[\hat{\boldsymbol{A}},\boldsymbol{X},\boldsymbol{Y};\ \boldsymbol{W}] \tag{6}$$

外部最小化原始数据的损失利用投影梯度下降算法解决，内部最大化生成对抗数据的损失由普通梯度下降算法进行优化。

投影梯度下降算法可以对离散图数据进行有效的扰动，具有较优的攻击效果。但是在投毒攻击的场景下，式(6)将模型参数看成与扰动无关的变量，而非扰动的函数，忽略了两者之间的联系。

2 基于改进投影梯度下降算法的投毒攻击方法

2.1 攻击模型

本文基于改进投影梯度下降算法，使用一个两层GCN构建投毒攻击模型，将特征矩阵看作常数，仅对图的邻接矩阵进行扰动，构建一个扰动矩阵$\boldsymbol{S}$并对邻接矩阵$\boldsymbol{A}$实施扰动。当$\boldsymbol{S}_{i,j}=\boldsymbol{S}_{j,i}=1$时，表示节点$v_i$和节点$v_j$之间的连边被扰动，即删除或添加；当$\boldsymbol{S}_{i,j}=\boldsymbol{S}_{j,i}=0$时，表示节点$v_i$和节点$v_j$之间的连边未被扰动。扰动矩阵$\boldsymbol{S}$对邻接矩阵$\boldsymbol{A}$的扰动过程如式(7)所示：

$$\hat{\boldsymbol{A}}=\boldsymbol{A}+(\overline{\boldsymbol{A}}-\boldsymbol{A})\circ\boldsymbol{S}$$
$$\text{s.t. }\overline{\boldsymbol{A}}=\boldsymbol{1}-\boldsymbol{A} \tag{7}$$

其中：$\hat{\boldsymbol{A}}$为扰动后的邻接矩阵；$\overline{\boldsymbol{A}}$为$\boldsymbol{A}$的补矩阵，补矩阵可以表示2个节点之间是否存在边；当$(\overline{\boldsymbol{A}}-\boldsymbol{A})_{i,j}=1$时，节点$v_i$和节点$v_j$之间不存在连边，攻击者可以添加连边；当$(\overline{\boldsymbol{A}}-\boldsymbol{A})_{i,j}=-1$时，节点$v_i$和节点$v_j$之间存在连边，攻击者可以将其删除；$\boldsymbol{1}$为元素全1的矩阵；$\circ$为矩阵中逐元素乘积。$\delta$为扰动预算，本文攻击方法寻求一个扰动矩阵$\boldsymbol{S}$导致GCN模型性能下降，在有限$\delta$内对节点进行错误分类，对应式(2)中的损失函数最小。本文攻击方法可以概括为以下约束优化问题：

$$\boldsymbol{S}=\underset{\boldsymbol{S}}{\operatorname{argmin}} L_{\text{atk}}[\hat{\boldsymbol{A}}(\boldsymbol{S}),\boldsymbol{X},\boldsymbol{Y};\boldsymbol{W}^*(\boldsymbol{S})]$$
$$\text{s.t.}\boldsymbol{W}^*(\boldsymbol{S})=\underset{\boldsymbol{W}(\boldsymbol{S})}{\operatorname{argmin}} L_{\text{train}}[\hat{\boldsymbol{A}}(\boldsymbol{S}),\boldsymbol{X},\boldsymbol{Y};\boldsymbol{W}(\boldsymbol{S})]$$
$$\|\boldsymbol{S}\|_0\leqslant 2\delta \tag{8}$$

本文将GCN模型训练参数$\boldsymbol{W}^*(\boldsymbol{S})$看作扰动矩阵$\boldsymbol{S}$的函数，而不是独立于扰动之外的变量，这是在投毒攻击场景下实施有效攻击的关键。在参数$\boldsymbol{W}^*(\boldsymbol{S})$训练过程中需要考虑扰动矩阵$\boldsymbol{S}$的影响。在本文每轮GCN模型参数$\boldsymbol{W}^*(\boldsymbol{S})$的训练过程中都基于扰动矩阵$\boldsymbol{S}$，投毒攻击过程主要分为3个步骤：1)构建扰动矩阵$\boldsymbol{S}$，根据上一轮的GCN模型参数$\boldsymbol{W}^*(\boldsymbol{S})$，采用投影梯度下降算法对$\boldsymbol{S}$进行训练，并对邻接矩阵$\boldsymbol{A}$实施扰动；2)模型根据扰动后的邻接矩阵$\hat{\boldsymbol{A}}(\boldsymbol{S})$训练GCN模型参数$\boldsymbol{W}^*(\boldsymbol{S})$；3)重复前2个步骤直至满足迭代总次数。

本文方法将数据投毒和对抗训练这2个阶段相结合，在GCN模型的对抗训练过程中考虑了扰动矩阵的影响，在更新扰动矩阵时考虑了GCN模型对抗训练的影响。根据式(8)，本文将投毒攻击分为数据投毒阶段和对抗训练阶段。

2.1.1 数据投毒阶段

在此阶段，攻击者对数据进行投毒并训练，以优化自身攻击方法。从式(8)可知，扰动矩阵$\boldsymbol{S}$是一个对角线元素为0的对称矩阵，$\boldsymbol{S}$中含有$N(N-1)/2$个独立的扰动变量。这些扰动变量之和应不超过扰动总数δ，$\boldsymbol{S}$中的元素$s\in\{0,1\}$。在数据投毒阶段中，将$\boldsymbol{S}$凸松弛成连续的$\boldsymbol{S}^*$，$\boldsymbol{S}^*$中的元素$s^*\in[0,1]$。$\boldsymbol{S}^*$每个元素都是0~1之间的连续值，是一个不断被优化的矩阵。$\boldsymbol{S}$对邻接矩阵$\boldsymbol{A}$的扰动数不能超过最大扰动数δ，不能简单地根据梯度下降算法对$\boldsymbol{S}^*$中的元素s^*进行梯度更新，而需采用投影梯度下降算法。该算法是解决简单约束的连续优化算法，其基本思想是先使用梯度下降算法更新参数，再投影以保证更新的参数在可行域即约束条件之中。该过程如式(9)所示：

$$\boldsymbol{S}^{*(t)}=\boldsymbol{S}^{*(t-1)}-\eta_t\nabla_{\boldsymbol{S}^*}L_{\text{atk}}a[\boldsymbol{S}^{*(t-1)},\boldsymbol{X},\boldsymbol{Y};\ \boldsymbol{W}^{t-1}(\boldsymbol{S})] \tag{9}$$

其中：t为梯度下降算法的迭代次数；η_t为第t轮更新$\boldsymbol{S}^*$的学习率；$\boldsymbol{W}^{t-1}(\boldsymbol{S})$为第$t-1$轮扰动后GCN模型的训练参数。$\boldsymbol{S}^*$在更新过程中需要考虑到约束条件$\|\boldsymbol{S}^*\|_0\leqslant 2\delta$，在约束条件的范围内求解一个最接近$\boldsymbol{S}^*$的向量并记作$\boldsymbol{S}_p$，如式(10)所示：

$$\boldsymbol{S}_p=\prod_{\Phi}(\boldsymbol{S}^{*(t)})$$
$$\text{s.t.}\prod_{\Phi}(\boldsymbol{S}^{*(t)}):=\underset{\Phi}{\operatorname{argmin}}\|\boldsymbol{S}_p-\boldsymbol{S}^{*(t)}\|_2^2 \tag{10}$$

其中：$\boldsymbol{\Phi}$表示满足条件$\|\boldsymbol{S}^*\|_0\leqslant 2\delta$的约束空间。式(10)是$\boldsymbol{S}^{*(t)}$在约束空间$\boldsymbol{\Phi}$上的投影算子，将$\boldsymbol{S}^{*(t)}$投射到$\boldsymbol{\Phi}$连续空间中。此时，经过投影梯度下降算法得到的$\boldsymbol{S}_p$并不能直接作为扰动矩阵使用，使用二项分布随机抽样[21]得到二值化的$\boldsymbol{S}$，如式(11)所示：

$$\boldsymbol{S}=\text{random binomial}(\boldsymbol{S}_p) \tag{11}$$

再由式(7)得到扰动后的邻接矩阵$\hat{\boldsymbol{A}}$。至此，本文完成了对数据的一轮“投毒”。

2.1.2 对抗训练阶段

在此阶段，GCN模型对攻击者投毒后的数据做出反应并展开对抗训练。在攻击者对GCN进行一

轮扰动之后，GCN模型根据扰动后的邻接矩阵 $\hat{\boldsymbol{A}}(\boldsymbol{S})$ 对训练参数 $\boldsymbol{W}(\boldsymbol{S})$ 进行对抗训练，使得模型的损失函数 L_{train} 最小。GCN模型参数 $\boldsymbol{W}(\boldsymbol{S})$ 采用梯度下降算法训练，过程如式(12)所示：

$$\boldsymbol{W}^{t}(\boldsymbol{S})=\boldsymbol{W}^{t-1}(\boldsymbol{S})-\alpha\nabla_{\boldsymbol{W}(\boldsymbol{S})}L_{\text{train}}[\hat{\boldsymbol{A}}(\boldsymbol{S}^{t}),\boldsymbol{X},\boldsymbol{Y};\ \boldsymbol{W}^{t-1}(\boldsymbol{S})] \tag{12}$$

其中：α 为GCN模型训练参数 $\boldsymbol{W}(\boldsymbol{S})$ 更新的学习率。式(12)表示训练参数 $\boldsymbol{W}(\boldsymbol{S})$ 为扰动矩阵 $\boldsymbol{S}$ 的函数，通过对扰动矩阵 $\boldsymbol{S}$ 求梯度以最小化损失函数，如式(13)所示：

$$\nabla_{\boldsymbol{S}}L_{\text{train}}[\hat{\boldsymbol{A}}(\boldsymbol{S}^{t}),\boldsymbol{X},\boldsymbol{Y};\ \boldsymbol{W}^{t-1}(\boldsymbol{S})]=\nabla_{\boldsymbol{S}}L_{\text{train}}[\hat{\boldsymbol{A}}(\boldsymbol{S}^{t}),\boldsymbol{X},\boldsymbol{Y}]+\nabla_{\boldsymbol{W}(\boldsymbol{S})}L_{\text{train}}[\boldsymbol{X},\boldsymbol{Y},\boldsymbol{W}^{t-1}(\boldsymbol{S})]\cdot\nabla_{\boldsymbol{S}}[\boldsymbol{W}^{t-1}(\boldsymbol{S})] \tag{13}$$

其中：$\nabla_{\boldsymbol{S}}L_{\text{train}}[\hat{\boldsymbol{A}}(\boldsymbol{S}^{t}),\boldsymbol{X},\boldsymbol{Y}]$ 为损失函数 $L_{\text{train}}[\hat{\boldsymbol{A}}(\boldsymbol{S}^{t}),\boldsymbol{X},\boldsymbol{Y};\boldsymbol{W}^{t-1}(\boldsymbol{S})]$ 固定 $\boldsymbol{W}^{t-1}(\boldsymbol{S})$ 对 $\boldsymbol{S}$ 的梯度；$L_{\text{train}}[\boldsymbol{X},\boldsymbol{Y},\boldsymbol{W}^{t-1}(\boldsymbol{S})]$ 为损失函数 $L_{\text{train}}[\hat{\boldsymbol{A}}(\boldsymbol{S}^{t}),\boldsymbol{X},\boldsymbol{Y};\ \boldsymbol{W}^{t-1}(\boldsymbol{S})]$ 固定 $\hat{\boldsymbol{A}}(\boldsymbol{S}^{t})$ 对 $\boldsymbol{W}^{t-1}(\boldsymbol{S})$ 的梯度。由式(13)可知，$\nabla_{\boldsymbol{S}}L_{\text{train}}[\hat{\boldsymbol{A}}(\boldsymbol{S}^{t}),\boldsymbol{X},\boldsymbol{Y};\boldsymbol{W}^{t-1}(\boldsymbol{S})]$ 由 t 轮扰动后的邻接矩阵 $\hat{\boldsymbol{A}}(\boldsymbol{S}^{t})$ 和上一轮的训练参数 $\boldsymbol{W}^{t-1}(\boldsymbol{S})$ 求得，而它们均在上一轮投毒阶段训练的扰动矩阵 $\boldsymbol{S}^{t-1}$ 的基础上，经过梯度下降算法所得。由此向前迭代可知，此梯度可由第一轮训练参数决定。因此，本文将上述2个过程相结合并建立攻击模型，攻击者在攻击过程中考虑GCN模型的对抗性训练，每一轮的训练对攻击的优化都基于上一轮GCN模型的训练参数。

2.2　算法架构

根据上述理论，基于改进投影梯度下降算法的投毒攻击方法(PGattack)主要分为5个步骤：1)根据上一轮的扰动矩阵 $\boldsymbol{S}$ 求得扰动后的邻接矩阵；2)图卷积网络进行正向训练，获得GCN模型训练参数；3)求攻击梯度矩阵并根据投影梯度下降算法更新扰动矩阵 $\boldsymbol{S}_p$；4)将矩阵 $\boldsymbol{S}_p$ 二值化得到扰动矩阵 $\boldsymbol{S}$；5)根据扰动矩阵 $\boldsymbol{S}$ 对邻接矩阵进行扰动，重复步骤1~步骤4，直至满足迭代总次数iters停止。

本文攻击算法的伪代码如算法1所示。

算法1　基于改进投影梯度下降的图卷积网络投毒攻击算法

输入　邻接矩阵 $\boldsymbol{A}$，特征矩阵 $\boldsymbol{X}$，标签 $\boldsymbol{Y}$，扰动预算 δ，学习率 η_t，初始扰动矩阵 $\boldsymbol{S}$，训练轮数iters

输出　扰动矩阵 $\boldsymbol{S}$

```
1.FOR t in range (iters):
2.Â = get_modified(A,S);
3.W* = train_GCN(Â,X,Y)
4.S_p = PGD(S,W*,δ,η_t)
5.S = random binomial (S_p)
6.RETURN S
```

本文提出的PGattack攻击方法架构如图1所示，该方法综合考虑投毒攻击的双层优化问题，实现数据投毒与GCN模型对抗训练2个过程的结合。

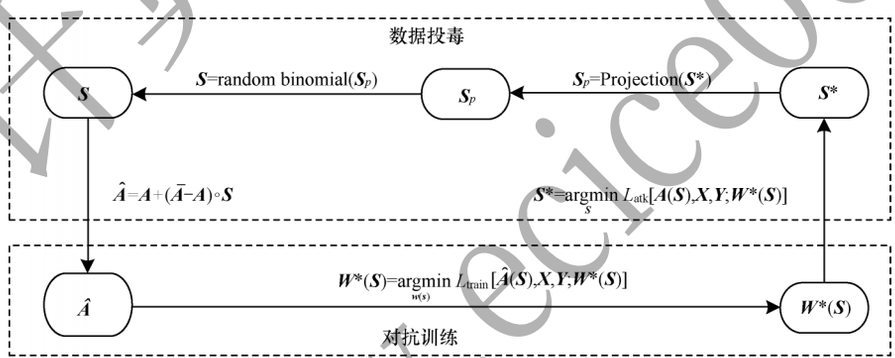

图1　PGattack攻击方法架构

Fig.1　Framework of PGattack attack method

3　实验验证

为分析本文攻击方法的主要影响因素并评估其攻击效果，本文实验采用型号为TITAN Xp的GPU显卡，运行环境为ubuntu 16.04系统，cuda10.0、Python3.7以及Pytorch1.2.0。采用图深度学习常用的Citeseer[22]、Cora[22]、Cora_ml[23]和Polblogs[24]数据集，表1所示为这些数据集的统计特征。数据集随机划分为标记节点和未标记节点，其中标记节点全部用于训练(10%)。在未标记的节点中，一部分用于测试(80%)，另一部分用于验证(10%)。本文将数据集随机划分10次，并记录10次实验结果的平均值。

表1　数据集统计特征

Table 1　Statistical characteristics of datasets

数据集	节点数	连边数	标签类别
Citeseer	2 110	3 668	6
Cora	2 486	5 069	7
Cora_ml	2 812	7 981	7
Polblogs	1 224	16 714	2

本文实验对GCN模型的连边进行扰动，将扰动连边数与连边总数的比值称作扰动比例。GCN模型在不同数据集上未受到干扰时的分类准确率对比如表2所示。

表2 图卷积网络模型未受到干扰的分类准确率对比

Table 2 Classification accuracy comparison of graph convolutional network model without disturbance

模型	分类准确率			
	Citeseer数据集	Cora数据集	Cora_ml数据集	Polblogs数据集
图卷积网络模型	0.720 7	0.830 0	0.853 6	0.955 0

3.1 参数设置

本文攻击方法的性能与参数设置相关,攻击过程主要由数据投毒与对抗训练2个阶段组成,训练轮数与学习率直接影响攻击效果。为尽可能提高攻击性能,本文在以上4个数据集上进行实验,并根据实验数据设置相应参数。

3.1.1 训练轮数设置

在扰动比例为5%的情况下,本文方法的数据投毒阶段的训练轮数从10次增至200次,分别记录分类准确率。文献[18]将数据投毒阶段投影梯度下降的学习率设置为iters/$\sqrt{t}$,t为当前迭代轮数,iters为训练总轮数,对抗训练的学习率设为0.01,对抗训练轮数设为100。在Citeseer、Cora、Cora_ml和Polblogs数据集上,不同训练轮数下GCN模型被PGattack攻击后的分类准确率如表3和图2所示。

表3 在不同训练轮数下图卷积网络模型被PGattack攻击后的分类准确率

Table 3 Classification accuracy of graph convolutional network model by PGattack attack in different training rounds

训练轮数	分类准确率			
	Citeseer数据集	Cora数据集	Cora_ml数据集	Polblogs数据集
0	0.720 7	0.830 0	0.853 6	0.955 0
10	0.718 3	0.816 5	0.846 1	0.903 9
20	0.706 0	0.809 0	0.836 2	0.898 9
30	0.696 3	0.803 4	0.824 6	0.881 4
40	0.694 9	0.801 4	0.825 8	0.876 4
50	0.690 5	0.800 3	0.821 8	0.865 2
60	0.690 5	0.800 6	0.820 9	0.864 6
70	0.691 3	0.799 9	0.819 0	0.862 4
80	0.691 3	0.798 6	0.821 6	0.866 4
90	0.690 5	0.799 9	0.820 9	0.862 2
100	0.691 3	0.799 1	0.820 7	0.866 2
110	0.689 0	0.798 0	0.819 2	0.866 9
120	0.690 9	0.797 3	0.818 2	0.862 6
130	0.690 0	0.800 1	0.819 2	0.865 8
140	0.688 4	0.795 3	0.818 3	0.863 7
150	0.690 0	0.799 6	0.820 3	0.862 9
160	0.688 0	0.798 3	0.817 8	0.859 9
170	0.689 7	0.795 2	0.819 4	0.860 9
180	0.688 4	0.798 5	0.820 1	0.860 6
190	0.688 4	0.798 8	0.821 4	0.862 4
200	0.689 0	0.799 4	0.818 2	0.864 3

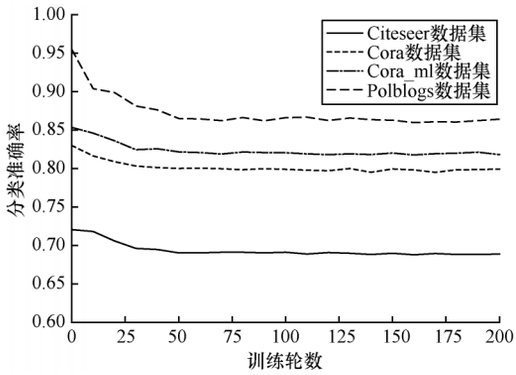

图2 图卷积网络模型被本文方法攻击后的分类准确率

Fig.2 Classification accuracy of graph convolutional network model by the proposed method attack

从图2可以看出,当训练轮数超过50之后,GCN模型的分类准确率在一定范围内波动,呈现收敛趋势;同理,对GCN模型对抗训练阶段的轮数进行对比实验,得出类似结果。后续实验将2个阶段的训练轮数均设置为50。

3.1.2 对抗训练学习率设置

本文方法由数据投毒与对抗训练2部分组成,本文实验选取扰动比例为5%,首先根据实验经验和粗粒度测试,将学习率从0.01增至0.10,分别记录GCN模型的分类准确率。在Citeseer、Cora、Cora_ml和Polblogs数据集上,不同学习率下GCN模型受PGattack攻击后的分类准确率如表4所示。综合表4数据,为取得更优的攻击性能,本文将学习率设为0.02。

表4 在不同学习率下图卷积网络模型被PGattack攻击后的分类准确率

Table 4 Classification accuracy of graph convolutional network model by PGattack attack in different learning rates

学习率	分类准确率			
	Citeseer数据集	Cora数据集	Cora_ml数据集	Polblogs数据集
0.01	0.691 3	0.801 5	0.820 9	0.866 2
0.02	0.688 0	0.799 4	0.818 2	0.864 3
0.03	0.694 9	0.798 8	0.821 4	0.862 4
0.04	0.689 0	0.803 4	0.818 2	0.865 2
0.05	0.688 4	0.809 0	0.821 8	0.871 4
0.06	0.690 5	0.798 6	0.820 7	0.868 9
0.07	0.690 9	0.799 9	0.818 2	0.862 2
0.08	0.688 4	0.801 4	0.821 4	0.864 3
0.09	0.691 3	0.800 6	0.817 8	0.866 4
0.10	0.690 0	0.800 1	0.820 9	0.871 4

3.2 攻击效果实验

本文根据训练模型对比分类准确率,以评估攻击效果,分类准确率下降幅度越大,说明GCN模型的性能下降越明显,则攻击方法的效果越好。本文在Citeseer、Cora、Cora_ml和Polblogs数据集上进行

实验，训练轮数为50轮，对抗训练学习率为0.02，将扰动比例从1%增至10%，分别记录GCN模型在不同扰动比例下的分类准确率。随着扰动比例的增加，GCN模型分类准确率背离初始准确率并逐步递减。在Citeseer、Cora、Cora_ml和Polblogs数据集上，在不同的扰动比例下本文攻击方法对GCN模型进行扰动后的分类准确率如表5和图3所示。扰动比例越大（即扰动的连边数越多），GCN模型的分类准确率越小，与预期相符，验证了攻击效果。

表5　在不同扰动比例下图卷积网络模型被PGattack攻击后的分类准确率

Table 5　Classification accuracy of graph convolutional network model by PGattack attack in different disturbance ratios

扰动比例/%	分类准确率			
	Citeseer数据集	Cora数据集	Cora_ml数据集	Polblogs数据集
0	0.720 7	0.830 0	0.853 6	0.955 0
1	0.710 5	0.814 9	0.847 8	0.907 3
2	0.705 5	0.815 6	0.839 4	0.884 2
3	0.696 4	0.809 6	0.835 0	0.869 2
4	0.693 8	0.802 0	0.823 9	0.865 0
5	0.688 0	0.799 4	0.818 2	0.864 3
6	0.683 2	0.791 0	0.819 6	0.862 6
7	0.678 0	0.785 0	0.819 8	0.861 6
8	0.676 7	0.780 1	0.811 5	0.860 2
9	0.668 8	0.774 5	0.809 1	0.858 1
10	0.666 1	0.770 4	0.808 1	0.856 2

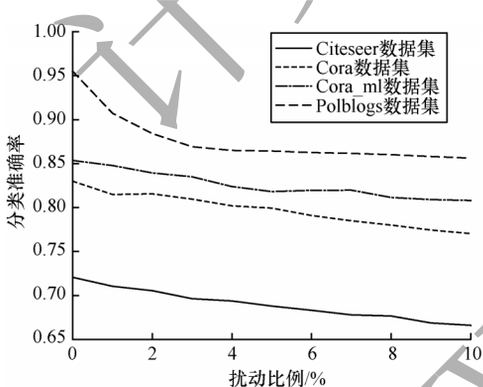

图3　图卷积网络模型被本文方法攻击后的分类准确率

Fig.3　Classification accuracy of graph convolutional network model by the proposed method attack

3.3　与其他攻击方法对比

本节将基于改进投影梯度下降算法的攻击方法PGattack与基准方法的攻击性能与复杂度进行对比。基准方法包括以下4个：1）随机攻击方法Random[19]，从训练集中随机选择增加连边或删除；2）删除同类连接异类[25]DICE攻击方法，根据“从同类节点中删除连边，在不同类节点间增加连边”规则，利用启发式算法删除部分连边，随后通过添加连边恢复其影响力[24]；3）Min-max攻击方法[18]，将攻击生成问题变成最小最大化的问题，最小化攻击的损失由投影梯度下降算法解决，最大化模型生成对抗数据的损失由梯度上升算法来解决；4）Metattack攻击方法[26]，使用元学习中的元梯度方法解决投毒攻击的双层优化问题，通过计算元梯度以指导攻击行为，采用贪婪算法对邻接矩阵遍历扰动以实现攻击。

3.3.1　攻击性能分析

本文在Citeseer、Cora、Cora_ml和Polblogs数据集上，对GCN模型中1%~5%的连边数进行攻击，训练轮数为50，对抗训练学习率为0.02，记录攻击后的分类准确率。不同方法攻击后图卷积网络模型的分类准确率如表6和图4所示。其中，图4的Clean表示模型在攻击前的准确率。

表6　图卷积网络模型被不同方法攻击后的分类准确率

Table 6　Classification accuracy of graph convolutional network model after attacks by different methods

方法	扰动比例/%	分类准确率			
		Citeseer数据集	Cora数据集	Cora_ml数据集	Polblogs数据集
Random	0	0.720 7	0.830 0	0.853 6	0.955 0
	1	0.715 0	0.818 4	0.849 2	0.928 4
	2	0.718 0	0.814 9	0.843 4	0.913 1
	3	0.714 5	0.811 4	0.841 2	0.896 7
	4	0.718 0	0.809 4	0.839 0	0.891 6
	5	0.712 1	0.806 8	0.835 0	0.885 5
DICE	0	0.720 7	0.830 0	0.853 6	0.855 0
	1	0.718 0	0.820 7	0.853 6	0.920 2
	2	0.716 8	0.815 9	0.849 0	0.909 0
	3	0.715 0	0.813 4	0.843 0	0.886 5
	4	0.712 7	0.813 4	0.841 6	0.878 3
	5	0.711 5	0.807 3	0.842 5	0.868 9
Min-max	0	0.720 7	0.830 0	0.853 6	0.955 0
	1	0.723 9	0.826 0	0.856 8	0.930 5
	2	0.718 6	0.817 9	0.847 0	0.929 4
	3	0.706 8	0.809 9	0.844 3	0.887 5
	4	0.711 5	0.805 2	0.838 1	0.890 6
	5	0.696 7	0.800 2	0.826 1	0.875 3
Metattack	0	0.720 7	0.830 0	0.853 6	0.955 0
	1	0.731 4	0.821 3	0.856 4	0.918 3
	2	0.708 3	0.815 6	0.843 4	0.886 8
	3	0.697 5	0.810 4	0.835 7	0.874 5
	4	0.688 0	0.799 4	0.821 1	0.868 3
	5	0.684 6	0.788 0	0.803 6	0.857 0
PGattack	0	0.720 7	0.830 0	0.853 6	0.955 0
	1	0.710 5	0.914 0	0.847 8	0.907 3
	2	0.705 3	0.815 6	0.839 4	0.884 2
	3	0.696 4	0.809 6	0.835 0	0.869 2
	4	0.693 8	0.802 0	0.823 9	0.865 0
	5	0.668 0	0.799 4	0.818 2	0.864 3

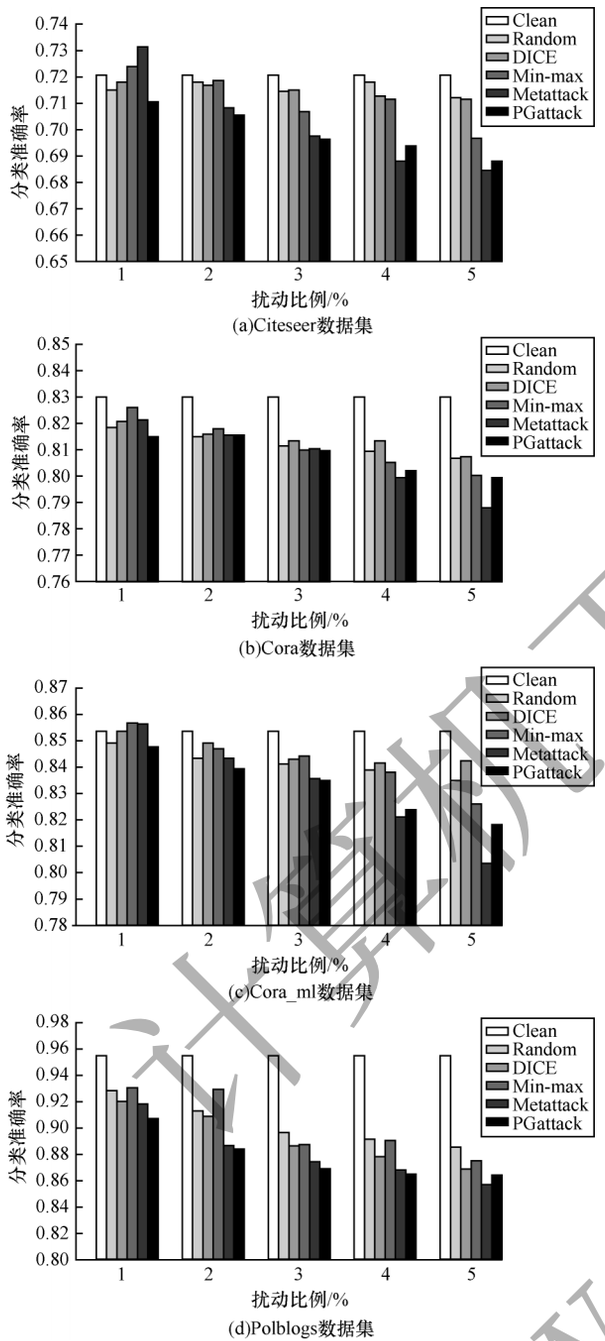

图4 图卷积网络模型的分类准确率

Fig.4 Classification accuracy of graph convolutional network model

从表6可以看出，当扰动比例为5%以下时，相比Random、DICE、Min-max攻击方法，本文PGattack方法具有更好的攻击效果，GCN模型的分类准确率下降辐度更大。

本文方法与传统投影梯度下降攻击方法Min-max相比，扰动筛选算法均采用投影梯度下降算法。而Min-max方法将训练后的参数视为固定常数，在此基础上使用投影梯度下降算法进行扰动；PGattack方法将参数视为扰动的函数而非独立变量，在梯度攻击过程中考虑模型的对抗训练阶段以优化攻击算法。实验结果验证了PGattack方法的有效性，即在攻击中考虑模型的对抗训练过程以提升攻击效果。

本文方法与Metattack相比，当扰动比例为3%以下时，PGattack的攻击效果优于Metattack；当扰动比例为4%以上时，PGattack的攻击效果与Metattack相比较差。Metattack采用贪婪算法实施攻击，随着扰动比例的增大，攻击效果逐渐提升。当扰动数据量较大时，贪婪算法对离散数据扰动的准确性更高，但是需要逐一对元素实施扰动，计算复杂度较高。

3.3.2 复杂度分析

随机增删连边的Random和基于简单的规则增删连边的DICE攻击方法均不需要对模型进行训练，计算复杂度较低。Min-max先求解训练参数$\boldsymbol{W}$，再将$\boldsymbol{W}$视为常量，使用投影梯度下降算法实施扰动，未考虑模型的对抗训练过程，复杂度中等，但是高于DICE和Random。PGattack将训练参数$\boldsymbol{W}$视为与扰动相关的变量，在采用投影梯度下降算法实施扰动时考虑模型的对抗训练过程，计算复杂度大于Min-max。Metattack攻击方法先求解元梯度，之后采用贪婪算法对连边逐个扰动，计算复杂度随扰动比例增加呈线性增长。

在4个数据集上不同攻击方法的时间开销对比如表7所示。

表7 不同攻击方法的时间开销对比

Table 7 Time costs comparison among different attack methods

方法	扰动比例/%	时间开销/s			
		Citeseer数据集	Cora数据集	Cora_ml数据集	Polblogs数据集
Random	1	16	15	15	16
	2	16	16	16	16
	3	16	16	15	16
	4	16	15	16	16
	5	16	16	16	16
DICE	1	18	16	19	17
	2	17	16	19	16
	3	17	16	19	17
	4	18	16	19	17
	5	17	16	19	17
Min-max	1	45	53	65	24
	2	45	54	65	24
	3	45	54	65	24
	4	45	54	65	25
	5	45	55	66	25
Metattack	1	63	58	120	93
	2	93	97	121	156
	3	152	155	122	248
	4	187	193	123	335
	5	235	245	123	438
PGattack	1	95	75	152	57
	2	96	76	265	57
	3	97	76	387	57
	4	98	77	517	57
	5	98	78	623	58

从表7可以看出，除Metattack以外，其他4种攻击方法的耗时排序：PGattack>Min-max>DICE>Random。Metattack随着扰动连边数增多，耗时呈线

性增长趋势。实验结果与理论分析一致。Metattack攻击方法在扰动比例较大时，尽管具有更好的攻击效果，但时间开销较大。本文所提的PGattack方法能够兼顾攻击性能与复杂度，具有更好的实用性。

4　结束语

本文提出基于改进投影梯度下降算法的投毒攻击方法，将模型训练参数看作扰动的函数，采用投影梯度下降算法对图的连边进行扰动，同时考虑模型的对抗训练过程。实验结果表明，当扰动比例为5%时，相比Random、DICE、Min-max攻击方法，本文攻击方法能够有效降低图卷积网络模型的的分类准确率，在攻击性能与时间开销方面实现最佳平衡。后续将在图神经网络的链路预测、图分类、社区发现等任务中构建投毒攻击模型，分析图神经网络的脆弱机理，以提高本文方法的可扩展性。

参考文献

[1] WU Z H, PAN S R, CHEN F W, et al. A comprehensive survey on graph neural networks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(1): 4-24.

[2] TANG J, QU M, MEI Q. PTE: predictive text embedding through large-scale heterogeneous text networks[C]// Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2015: 1165-1174.

[3] WANG S H, TANG J L, AGGARWAL C, et al. Linked document embedding for classification[C]//Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. New York, USA: ACM Press, 2016: 115-124.

[4] PEROZZI B, Al-RFOU R, SKIENA S. Deepwalk: online learning of social representations[EB/OL]. [2021-11-10]. https://arxiv. org/pdf/1403. 6652. pdf.

[5] WANG S H, TANG J L, AGGARWAL C, et al. Signed network embedding in social media[C]//Proceedings of SIAM International Conference on Data Mining. Philadelphia, USA: Society for Industrial and Applied Mathematics, 2017: 327-335.

[6] TIAN F, GAO B, CUI Q, et al. Learning deep representations for graph clustering[C]//Proceedings of the 28th AAAI Conference on Artificial Intelligence. [S. l.]: AAAI Press, 2014: 1293-1299.

[7] ALLAB K, LABIOD L, NADIF M. A semi-NMF-PCA unified framework for data clustering[J]. IEEE Transactions on Knowledge and Data Engineering, 2017, 29(1): 2-16.

[8] DAI H J, LI H, TIAN T, et al. Adversarial attack on graph structured data[EB/OL]. [2021-11-10]. https://arxiv. org/pdf/1806. 02371. pdf.

[9] ZÜGNER D, AKBARNEJAD A, GÜNNEMANN S. Adversarial attacks on neural networks for graph data[C]// Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York, USA: ACM Press, 2018: 2847-2856.

[10] WANG X Y, EATON J, HSIEH C J, et al. Attack graph convolutional networks by adding fake nodes[EB/OL]. [2021-11-10]. https://arxiv. org/pdf/1810. 10751. pdf.

[11] 吴翼腾，刘伟，于淑乔. 基于参数差异假设的图卷积网络对抗性攻击[J/OL]. 电子学报：1-13[2021-11-10]. http://kns. cnki. net/kcms/detail/11. 2087. TN. 20211018. 1732. 002. html.
WU Y T, LIU W, YU X Q. Adversarial attacks on graph convolution networks based on parameter discrepancy hypothesis[J/OL]. Acta Electronica Sinica: 1-13[2021-11-10]. http://kns. cnki. net/kcms/detail/11. 2087. TN. 20211018. 1732. 002. html. (in Chinese)

[12] 吴翼腾，刘伟，于洪涛. 图神经网络的标签翻转对抗攻击[J]. 通信学报，2021，42(9)：65-74.
WU Y T, LIU W, YU H T. Label flipping adversarial attack on graph neural network[J]. Journal on Communications, 2021, 42(9): 65-74. (in Chinese)

[13] 姜妍，张立国. 面向深度学习模型的对抗攻击与防御方法综述[J]. 计算机工程，2021，47(1)：1-11.
JIANG Y, ZHANG L G. Survey of adversarial attacks and defense methods for deep learning model[J]. Computer Engineering, 2021, 47(1): 1-11. (in Chinese)

[14] JIN W, LI Y X, XU H, et al. Adversarial attacks and defenses on graphs: a review and empirical study[EB/OL]. [2021-11-10]. https://arxiv. org/abs/2003. 00653v2.

[15] CHEN L, LI J T, PENG J Y, et al. A survey of adversarial learning on graphs[EB/OL]. [2021-11-10]. https://arxiv. org/abs/2003. 05730?.

[16] WU Y T, LIU W, HU X B, et al. Parameter discrepancy hypothesis: adversarial attack for graph data[J]. Information Sciences, 2021, 577: 234-244.

[17] LIN X X, ZHOU C, YANG H, et al. Exploratory adversarial attacks on graph neural networks[C]//Proceedings of IEEE International Conference on Data Mining. Washington D. C., USA: IEEE Press, 2020: 1136-1141.

[18] XU K D, CHEN H G, LIU S J, et al. Topology attack and defense for graph neural networks: an optimization perspective[C]//Proceedings of the 28th International Joint Conference on Artificial Intelligence. New York, USA: ACM Press, 2019: 3961-3967.

[19] 陈晋音，张敦杰，黄国瀚，等. 面向图神经网络的对抗攻击与防御综述[J]. 网络与信息安全学报，2021，7(3)：1-28.
CHEN J Y, ZHANG D J, HUANG G H, et al. Adversarial attack and defense on graph neural networks: a survey[J]. Chinese Journal of Network and Information Security, 2021, 7(3): 1-28. (in Chinese)

[20] SUN M J, TANG J, LI H C, et al. Data poisoning attack against unsupervised node embedding methods[EB/OL]. [2021-11-10]. https://arxiv. org/pdf/1810. 12881. pdf.

[21] LIU S J, CHEPURI S P, FARDAD M, et al. Sensor selection for estimation with correlated measurement noise[J]. IEEE Transactions on Signal Processing, 2016, 64(13): 3509-3522.

[22] SEN P, NAMATA G, BILGIC M, et al. Collective classification in network data[J]. AI Magazine, 2008, 29(3): 93.

[23] MCCALLUM A, NIGAM K, RENNIE J D M, et al. Automating the construction of Internet portals with machine learning[J]. Information Retrieval, 2000, 3(2): 127-163.

[24] ADAMIC L A, GLANCE N. The political blogosphere and the 2004 US election: divided they blog[C]//Proceedings of the 3rd International Workshop on Link Discovery. New York, USA: ACM Press, 2005: 36-43.

[25] WANIEK M, MICHALAK T P, WOOLDRIDGE M J, et al. Hiding individuals and communities in a social network[J]. Nature Human Behaviour, 2018, 2(2): 139-147.

[26] ZÜGNER D, GÜNNEMANN S. Adversarial attacks on graph neural networks via meta learning[EB/OL]. [2021-11-10]. https://arxiv. org/pdf/1902. 08412. pdf.

编辑　薛晋栋