计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2022,31(1):212−217 [doi: 10.15888/j.cnki.csa.008269]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于梯度选择的图卷积网络针对性通用对抗攻击①

曹海芳

(天津大学 数学学院, 天津 300350)
通信作者: 曹海芳, E-mail: caohaifang@tju.edu.cn

摘　要: 图卷积网络 (GCN) 是处理图结构化数据的一种十分重要的方法, 最新的研究表明, GCN 极易受到对抗性攻击, 即通过修改少量数据, 就能显著影响 GCN 的结果. 在对 GCN 的所有对抗攻击中, 有一种特殊的对抗攻击方法——通用对抗攻击. 这种攻击能产生应用于所有样本的扰动, 并使 GCN 得到错误的结果. 本文主要研究针对性通用对抗攻击, 通过在现有算法 TUA 的基础上引入梯度选择的方法, 提出了 GTUA. 在 3 个流行数据集上的实验结果表明: 仅仅在少数类别上, 本文方法与现有方法结果相同, 在多数类别上, 本文方法均优于现有方法, 并且平均攻击成功率 (ASR) 得到 1.7% 的提升.

关键词: 梯度选择; 图神经网络; 图卷积网络; 通用对抗攻击; 针对性攻击

引用格式: 曹海芳.基于梯度选择的图卷积网络针对性通用对抗攻击.计算机系统应用,2022,31(1):212–217. http://www.c-s-a.org.cn/1003-3254/8269.html

Targeted Universal Adversarial Attack on GCN Based on Gradient Selection

CAO Hai-Fang

(School of Mathematics, Tianjin University, Tianjin 300350, China)

Abstract: The graph convolutional network (GCN) is a very important method of processing graph-structured data. The latest research shows that it is highly vulnerable to adversarial attacks, that is, modifying a small amount of data can significantly affect its result. Among all the adversarial attacks on a GCN, there is a special attack method—the universal adversarial attack. This attack can produce disturbances to all samples and cause an erroneous GCN result. This study mainly studies targeted universal adversarial attacks and proposes a GTUA by adding gradient selection to the existing algorithm TUA. The experimental results of three popular datasets show that only in a few classes, the method proposed in this study has the same results as those of the existing methods. In most classes, the method proposed in this study is superior to the existing ones. The average attack success rate (ASR) is improved by 1.7%.

Key words: gradient selection; graph neural network; graph convolutional network (GCN); universal adversarial attack; targeted attack

图结构数据已广泛应用于许多现实世界的应用程序中, 例如社交网络 (Facebook 和 Twitter), 生物网络 (蛋白质或基因相互作用) 以及属性图 (PubMed 和 Arxiv) 等[1−3]. 节点分类任务是图结构数据上最重要的任务之一, 即给定一个节点子集及其标签, 预测其余节点的标签. 对于节点分类任务, 基于图的深度学习模型——图神经网络已实现了最先进的性能[4], 而图卷积网络 (GCN) 作为一种特殊的图神经网络, 在此任务上取得了更好的结果.

目前的研究更多的是将重点放在如何提高 GCN 的性能上, 却很少有人关注 GCN 模型的鲁棒性. 但是, 研究表明, GCN 是极易受到对抗攻击的. 例如, 只须对

① 收稿时间: 2021-03-20; 修改时间: 2021-04-19; 采用时间: 2021-05-07; csa 在线出版时间: 2021-12-17

图数据的拓扑结构或者节点的特征进行微小的修改就能使 GCN 得到错误的分类结果[5]. 目前的攻击方法中, 绝大多数是通过修改图数据的拓扑结构和节点属性来进行攻击的, 然而, 这样的攻击在现实场景中是不适用的. 例如, 在社交网络应用程序中, 攻击者必须登录用户的帐户才能更改现有的连接和功能, 而获得登录访问权限几乎是不可能的. 相比之下, 在实践中添加与用户相对应的伪节点 (fake node) 会容易得多.

TUA 就是一种通过添加伪节点进行攻击的针对性通用攻击方法. 在针对 GCN 的所有攻击方法中, 通用攻击方法是一种特殊的攻击方法, 此方法要求 GCN 将所有的受害节点都错误分类, 而不是某个单独的节点[6–8]. 针对性通用攻击则要求 GCN 将所有受害节点都错误地分到某一个指定的类别[9]. 本文则是基于 TUA 算法, 通过引入梯度选择的方法, 使得本文方法在所有类别的实验中都取得了与 TUA 方法相当甚至优于 TUA 方法的结果, 平均 ASR 相对 TUA 得到了 1.7% 的提升.

1 相关知识介绍

1.1 图卷积网络 (GCN)[4,10]

给定属性图$G(A,X)$, 其中$A\in\{0,1\}^{N\times N}$为邻接矩阵, $X\in\{0,1\}^{N\times d}$为特征矩阵, 即图 G 有 N 个节点, 且每个节点伴随一个d维的特征. 令$V=\{v_1,v_2,\cdots,v_N\}$为节点集, $C=\{c_1,c_2,\cdots c_k\}$为类别集. 节点分类任务的目标就是通过在含有节点标签的训练集上学习从而成功预测测试集节点的标签. GCN 首先通过聚合邻居节点的信息来得到节点的嵌入表示 (第l层如下):

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

其中, $\tilde{A}=A+I_N$, $\tilde{D}_{ii}=\sum_j\tilde{A}_{ij}$分别是添加自连接之后的邻接矩阵和度矩阵, $W^{(l)}$是权重矩阵, $\sigma(x)$是激活函数. 初始状态$H^{(0)}=X$, $H^{(l)}$则是第l层的嵌入输出. 根据 Kipf 等[4], 仅用一个隐藏层, 最后 GCN 的输出为:

$$Z=f(A,X)=Softmax(\hat{A}ReLU(\hat{A}XW^{(0)})XW^{(1)}) \quad (2)$$

其中, $\hat{A}=\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$.

1.2 图对抗攻击

在过去的几年中, Zungner 等[6] 和 Dai 等[5] 首先发现了 GCN 容易受到对抗性攻击的特性. 而根据攻击的不同阶段, GCN 中的对抗攻击分为两种类型: 投毒攻击 (训练期间的攻击) 和逃避攻击 (测试期间的攻击). 通常, 投毒攻击的重点是通过干扰训练数据来降低 GCN 模型的性能, 而逃避攻击则通过修改属性或拓扑结构来构造对抗性样本, 从而使 GCN 模型的性能降低. 另外, 根据攻击的不同目的, 对图结构化数据的对抗攻击可分为节点分类攻击, 链接预测攻击和图分类攻击.节点分类攻击的目的是使某些节点被 GCN 误分类. 链接预测攻击的重点是减少节点之间的关联, 从而导致 GCN 提供错误的预测结果. 图分类攻击则旨在增强指定图与目标分类之间的相关性, 以使 GCN 无法正确分类给定图样本. 本文提出的 GTUA 可以归为逃避攻击和节点分类攻击.

在对图结构数据的所有对抗攻击中, 伪节点攻击是一种常见的攻击方法, 通过将一组伪节点注入到图中来实现, 从而可以避免对原始图进行拓扑或属性修改. 例如, GreedyAttack 和 GreedyGAN 通过将伪造的节点直接添加到受害节点来进行目标节点攻击[11]. Wang 等[12] 引入近似快速梯度符号法, 该方法在受害节点和其他节点之间添加了一个恶性节点, 从而导致受害节点被错误分类. 但是, 大多数现有的伪节点攻击并非旨在进行普遍的对抗攻击. 而在本文提出的 GTUA 中, 伪节点充当受害节点的 2 跳邻居. 由于 GCN 的攻击过程, 伪节点特征的影响通过攻击节点传递到受害节点, 从而进行针对性通用对抗攻击.

2 基于梯度选择的图卷积网络针对性通用对抗攻击

基于梯度选择的图卷积网络针对性通用对抗攻击 (GTUA) 的目标是使得每一个与攻击节点 (从标签为目标类别的节点集中随机选择) 连接的受害节点都得到与攻击节点相同的标签 (如图 1 所示). 主要由 3 个步骤完成: 添加伴随 0 特征的伪节点; 计算目标函数关于伪节点特征矩阵的梯度矩阵; 按梯度矩阵元素大小进行梯度选择并确定扰动特征. 下面分别详细介绍每一个步骤.

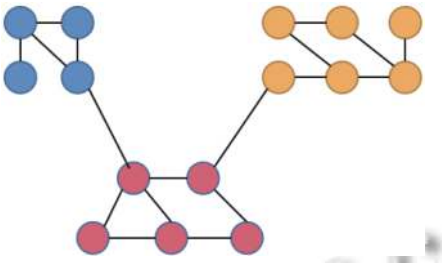
2.1 添加伴随 0 特征的伪节点

在添加伪节点之前, 先简单介绍几步预处理过程: 对于给定的图$G(A,X)$, 首先选定一个类别c_o作为目标类别, 随后在标签为c_o的节点集中随机选择N_A个节点作为攻击节点$V_A=\{v_A^1,v_A^1,\cdots,v_A^{N_A}\}$, 同时为了简便起见, 规定为每个攻击节点连接相同数量N_F个伪节点. 即, 给定图$G(A,X)$, 目标类别c_o, 攻击节点V_A, 每个攻击节点的伪节点数目N_F, 通过给每个攻击节点连接特征

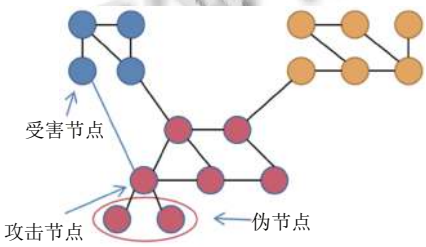
为 0 的伪节点得到新图$G'=(A',X')$, 其中,

$$A'=\begin{bmatrix} A & E \\ E^{\mathrm{T}} & P \end{bmatrix},\ X'=\begin{bmatrix} X \\ X_F \end{bmatrix} \tag{3}$$

其中, $E\in\{0,1\}^{N\times(N_A\cdot N_F)}$, $P\in\{0,1\}^{(N_A\cdot N_F)\times(N_A\cdot N_F)}$, $X_F\in\{0,1\}^{(N_A\cdot N_F)\times d}$, 且初始化为全 0, 本文的目的就是通过给伪节点添加某些特征, 也就是X_F的某些元素由 0 改为 1, 从而使得下面的目标函数取得最大值.



(a) 原始图



(b) 扰动图

图 1 攻击前后的受害节点分类结果

2.2 计算目标函数关于伪节点特征矩阵的梯度矩阵

在此首先明确本文的目的是, 对于给定的图$G(A,X)$, 目标类别c_o, 攻击节点V_A, 希望每一个标签不是c_o的节点, 当其与V_A连接时, 能够使得 GCN 为其得到c_o的标签, 这也就是针对性通用攻击的含义. 由此含义, 首先给出一个随机选定的辅助节点集$V_T=\{v_T^1,v_T^2,\cdots,v_T^{N_T}\}$来帮助建立目标函数, 其中$N_T$表示辅助节点的数量, 要求$V_T$中的每一个节点都不属于标签$c_o$, 因此有下面的目标函数:

$$\begin{cases} \arg\max\limits_{X_F} \sum\limits_{v\in V_T} F(A',X',v) \\ \text{s.t. } \|E\|_0+\|X_F\|_0\le\Delta \end{cases} \tag{4}$$

其中, $\|E\|_0$表示伪节点与攻击节点之间的连边的数量, $\|X_F\|_0$表示给伪节点添加的特征的数量, 二者都受到参数Δ的限制, 因为扰动必须是微小的. 其中,

$$F(A',X',v)=\left[f\left(A'_{(v,V_A)},X'\right)\right]_{v,c_o}-\left[f\left(A'_{(v,V_A)},X'\right)\right]_{v,c_v} \tag{5}$$

式 (5) 是关于每一个辅助节点的目标函数部分, $v\in V_T$, $A'_{(v,A)}$表示辅助节点 v 与攻击节点V_A连接之后的新的邻接矩阵, $[f(\cdot)]_{v,c_o}$和$[f(\cdot)]_{v,c_v}$分别表示 GCN 将节点 v 判定为目标类别和其当前类别的输出概率. 而制定此目标函数的依据是: 如果本文的攻击或者说扰动能够使得 GCN 将这些非c_o的辅助节点在连接到V_A后被分类到c_o, 那么对于所有的非c_o的节点, 当其与V_A连接后, 就会有很大的概率被分类为c_o, 并且如果考虑一种极端情况: 将所有非c_o节点作为辅助节点, 那么就能更好地理解此目标函数.

前面确定了目标函数, 那么如何计算扰动? 在这里首先介绍基于梯度的方法. 因为只考虑对X_F进行扰动, 因此只需要先计算目标函数关于X_F的梯度:

$$Grad=\nabla_{X_F}\sum_{v\in V_T}F(A',X',v) \tag{6}$$

2.3 按梯度矩阵元素大小进行梯度选择并确定扰动特征

以往的基于梯度的方法基本都是采用一种贪婪式的选择方法[13]: 在每一次迭代中只改动一个元素, 因此找到每一次迭代中的梯度矩阵中的最大元素作为修改的对象即可. TUA 也是采用这样一种方式, 首先找到$Grad$中的最大元素$Grad_{\max}$:

$$\begin{cases} Grad_{\max}=\arg\max\limits_{i,j} Grad(i,j) \\ i\in 1,2,\cdots,N_F;\ j\in 1,2,\cdots,d \end{cases} \tag{7}$$

然后找到$Grad_{\max}$在X_F对应哪一个伪节点的哪一个特征, 再找到它在X'中的对应位置, 将该位置的元素由 0 置为 1, 这样就完成了一次迭代, 直到到达一定的阈值就结束扰动. 需要提到的一点是: 如果某一次迭代时最大梯度对应的位置已经是 1, 那么就寻找第二大的梯度位置, 依次下去.

本文提出的 GTUA 也是采用一种基于梯度的贪婪式方法, 但是在这个过程中加上一个梯度选择的过程. 具体做法就是在得到$Grad$之后, 选出其中从大到小前 k 个元素:

$$\begin{cases} Grad_1=\max\{Grad(i,j)\} \\ Grad_2=\max\{\{Grad(i,j)\}\setminus\{Grad_1\}\} \\ \quad\vdots \\ Grad_k=\max\{\{Grad(i,j)\}\setminus\{Grad_1,\cdots,Grad_{k-1}\}\} \end{cases} \tag{8}$$

然后分别计算将其在X'中对应位置的特征值由 0 置为 1 后的损失函数值:

$$\begin{cases} Loss_1 = \sum\limits_{v\in V_T} F(A', X_1', v) \\ Loss_2 = \sum\limits_{v\in V_T} F(A', X_2', v) \\ \quad\vdots \\ Loss_k = \sum\limits_{v\in V_T} F(A', X_k', v) \end{cases} \tag{9}$$

其中, $X_1', X_2', \cdots, X_k'$ 分别是对 $Grad_1, Grad_2, \cdots, Grad_k$ 位置进行扰动后的新的特征矩阵. 最后再选择其中得到最大损失函数值的特征修改作为这一次迭代的扰动:

$$Grad_{\text{obj}} = \arg\limits_{i,j}\left\{\arg\max\limits_{n} Loss_n\right\} \tag{10}$$

加入这样一个梯度选择的过程, 是出于以下思考: 因为考虑 A 和 X 都是取值为 0 或 1 的离散数据类型, 因此在选择 $Grad$ 中最大元素 $Grad_{\max}$ 对应的位置并由 0 置为 1 的时候相当于是选择了长度 1 的步长, 并且每一次迭代都是固定步长 1. 因此, 就很可能出现一种情况, $Grad$ 中第二大元素 $Grad_{\text{sec}}$ 对应的位置由 0 置为 1 会得到更大的损失函数, 如图 2 所示.

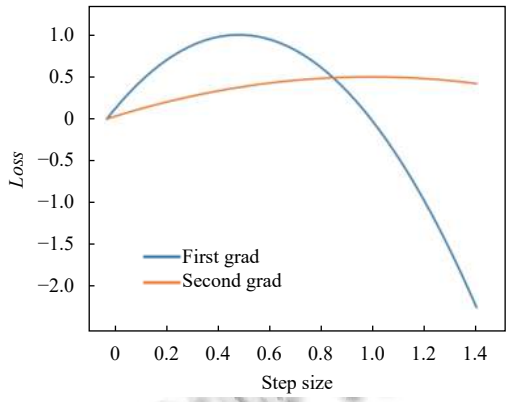


图 2 不同梯度下 loss 与步长的关系

3 实验分析

3.1 数据集和评价指标

本文在 3 个常用的属性图数据集上进行实验: Cora (2 708 节点, 5 429 边, 1 433 特征, 7 类别), Citeseer (3 312 节点, 4 732 边, 3 703 特征, 6 类别) 和 PubMed (19 717 节点, 44 338 边, 500 特征, 3 类别)[14–16]. 此外, 根据 Kipf&Welling 的设置, 在 3 个相应的数据集上训练 GCN 模型. 最后用平均攻击成功率 (ASR) 作为模型性能的评价标准, ASR 越高, 表明模型的攻击效果越好.

3.2 实验设置

首先按照 TUA 的方法加快微扰计算. 因为大多数基于梯度的攻击都存在时间和内存成本高的问题, 为了解决这个问题, Li 等[16] 提出了一种有效加速攻击的框架, 该攻击框架攻击由目标节点的 k 跳邻居组成的较小子图 (k 取决于 GCN 层数), 从而可以避免不必要的图信息存储和计算[17]. 因为本文是基于 Kipf&Welling 的设置来训练 GCN, 也就是一个 2 层的 GCN, 因此只需要关注以目标节点为中心, 以其一阶邻居和二阶邻居组成的子图即可. 根据 TUA 的实验发现, 当 N_F, N_T 固定时, 随着 N_A 的取值大于 3 之后, ASR 几乎不再随着 N_A 的增大而增大; 同样的, 固定 N_A, N_T 的取值时, 当 N_F 大于 2 之后, ASR 也不再随 N_F 的增大而增大; 固定 N_A, N_F 时, 当 N_T 达到 20 后, 随着 N_T 的增大, ASR 几乎不再增大. 但是, 无论其中哪一个参数增大, 对计算与存储的消耗都会成倍的增加. 同时, 对 GTUA 进行了相关参数的实验, 发现与 TUA 有着相似的规律. 因此在后续的实验中, 规定参数设置 $N_A = 3$, $N_F = 2$, $N_T = 20$.

3.3 实验结果与分析

本文进行两组实验, 第一组实验用来查看在梯度选择过程中选择不同数量的梯度值 N_G 对 ASR 带来的影响, 这里本文考虑 $N_G \in \{1,2,3,4,5,6\}$, 实验结果如图 3 所示. 由图 3 可知, 当加入了梯度选择的步骤之后, 多数情况下 ASR 值都是高于原始 ASR 值的, 且在这里的实验中发现 N_G 取 5 的时候能在多数情况下得到最大的 ASR 值, 因此后面的实验就选定 $N_G = 5$.

第二组实验选择一个恰当的 N_G 值, 将本文方法 GTUA 与 TUA 的 ASR 值进行对比来验证 GTUA 的有效性. 由实验一的结果, 本文选择 $N_G = 5$. 结果如表 1 所示, 可以很清楚地看到 GTUA 在多数情况下优于 TUA, 在少量情况下取得与 TUA 同样的结果, 而取得相同结果的原因是每一次迭代都在梯度最大值处的扰动能得到最大的损失函数值, 表 1 的最后一行也表明 GTUA 与 TUA 相比, 平均 ASR 提高了 1.7%.

表 2 展示了 GTUA 与 TUA 的一次训练加测试的耗时对比, 可以发现: 因为 GTUA 梯度选择的过程需要计算 N_G 次损失函数, 因此耗时要远大于 TUA, 且通过实验发现, 随着图的节点与边的增多, 两者之间的耗时

差距越来越大, 因此 GTUA 比较适用于小图, 而对于大图, 时间成本略高. 但是, 由于 GTUA 选择梯度的过程, 从而导致它并不依赖于损失函数对特征矩阵的最大梯度, 而 TUA 则严重依赖于上述最大梯度, 所以一些微小的改动并不会对 GTUA 模型的结果造成影响, 却会大大影响到 TUA 的结果, 所以 GTUA 相比 TUA 有更好的鲁棒性.

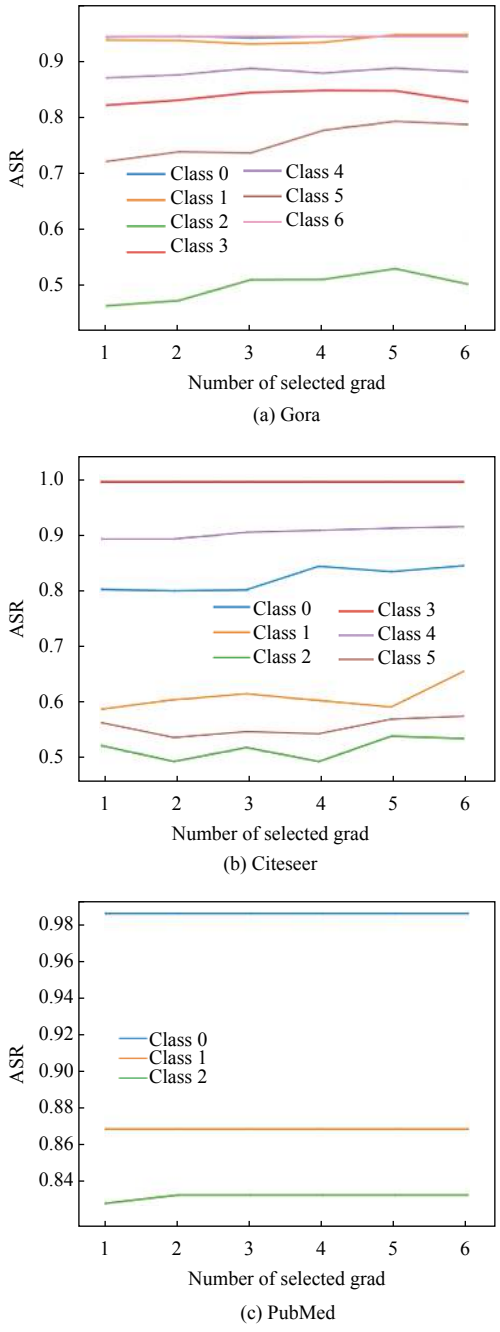


图 3 选择不同数量的梯度对 ASR 的影响

表 1 GTUA 与 TUA 的性能比较 (ASR)

数据集	类别	算法	
		TUA	GTUA
Cora	0	0.959 5	**0.960 5**
	1	0.954	**0.964**
	2	0.471	**0.538 5**
	3	0.835 8	**0.862**
	4	0.885 5	**0.903 5**
	5	0.733 5	**0.806 5**
	6	0.960 5	0.960 5
Citeseer	0	0.802	**0.834 5**
	1	0.58	**0.584**
	2	0.512 5	**0.53**
	3	1.0	1.0
	4	0.895	**0.915**
	5	0.555	**0.561 5**
PubMed	0	0.985 5	0.985 5
	1	0.869	0.869
	2	0.829	**0.833 5**
平均值	/	0.801 7	**0.819 3**

表 2 GTUA 与 TUA 算法的效率比较 (s)

算法	Cora	Citeseer	PubMed
TUA	28	27	53
GTUA	140	251	875

4 结论与展望

本文提出了一种基于梯度的图卷积网络针对性通用对抗攻击 GTUA, 实验结果表明, 与当前流行的方法 TUA 相比, GTUA 最差能够达到与其一样的结果, 但在多数情况下优于 TUA, 由此可以看出梯度选择的过程确实提升了扰动的质量. 另外, 本文也留下了一个后续的研究方向: 对于离散数据类型上的基于梯度的方法, 都可以尝试加入梯度选择的过程, 由本文的结果可以大胆地预测, 其在很大程度上可能会带来效果上的提升.

参考文献

1 Grover A, Leskovec J. Node2Vec: Scalable feature learning for networks. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco: ACM, 2016. 855–864.

2 Rhee SM, Seo S, Kim S. Hybrid approach of relation network and localized graph convolutional filtering for breast cancer subtype classification. Proceedings of the 27th International Joint Conference on Artificial Intelligence. Stockholm: IJCAI, 2018. 3527–3534.

3 Ying R, He RN, Chen KF, *et al*. Graph convolutional neural networks for web-scale recommender systems. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. London: ACM, 2018. 974–983.
4 Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. arXiv: 1609.02907, 2016.
5 Dai HJ, Li H, Tian T, *et al*. Adversarial attack on graph structured data. Proceedings of the 35th International Conference on Machine Learning. Stockholm: ICML, 2018. 1115–1124.
6 Zügner D, Akbarnejad A, Günnemann S. Adversarial attacks on neural networks for graph data. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. London: ACM, 2018. 2847–2856.
7 Bojchevski A, Günnemann S. Adversarial attacks on node embeddings via graph poisoning. Proceedings of the 36th International Conference on Machine Learning. Long Beach: ICML, 2019. 695–704.
8 Takahashi T. Indirect adversarial attacks via poisoning neighbors for graph convolutional networks. Proceedings of 2019 IEEE International Conference on Big Data (Big Data). Los Angeles: IEEE, 2019. 1395–1400.
9 Dai JZ, Zhu WF, Luo XF. A targeted universal attack on graph convolutional network. arXiv: 2011.14365, 2020.
10 刘杰, 李喜旺. 基于图神经网络的工控网络异常检测算法. 计算机系统应用, 2020, 29(12): 234–238. [doi: 10.15888/j.cnki.csa.007717]
11 Wang XY, Cheng MH, Eaton J, *et al*. Attack graph convolutional networks by adding fake nodes. arXiv: 1810.10751, 2018.
12 Wang JH, Luo MN, Suya F, *et al*. Scalable attack on graph data by injecting vicious nodes. Data Mining and Knowledge Discovery, 2020, 34(5): 1363–1389. [doi: 10.1007/s10618-020-00696-7]
13 Moosavi-Dezfooli SM, Fawzi A, Frossard P. DeepFool: A simple and accurate method to fool deep neural networks. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 2574–2582.
14 Sun MJ, Tang J, Li HC, *et al*. Data poisoning attack against unsupervised node embedding methods. arXiv: 1810.12881, 2018.
15 Guo C, Pleiss G, Sun Y, *et al*. On calibration of modern neural networks. Proceedings of the 34th International Conference on Machine Learning. Sydney: ICML, 2017. 1321–1330.
16 Li JT, Xie T, Chen L, *et al*. Adversarial attack on large scale graph. arXiv: 2009.03488, 2020.
17 Wu F, Zhang TY, De Souza Jr AH, *et al*. Simplifying graph convolutional networks. arXiv: 1902.07153, 2019.