Chapter 5
Adversarial Attacks on Graph Neural Network: Techniques and Countermeasures

Nimish Kumar
B.K. Birla Institute of Engineering and Technology, Pilani, India

Himanshu Verma
B.K. Birla Institute of Engineering and Technology, Pilani, India

Yogesh Kumar Sharma
Koneru Lakshmaiah Education Foundation (Deemed), India

ABSTRACT

Graph neural networks (GNNs) are a useful tool for analyzing graph-based data in areas like social networks, molecular chemistry, and recommendation systems. Adversarial attacks on GNNs include introducing malicious perturbations that manipulate the model's predictions without being detected. These attacks can be structural or feature-based depending on whether the attacker modifies the graph's topology or node/edge features. To defend against adversarial attacks, researchers have proposed countermeasures like robust training, adversarial training, and defense mechanisms that identify and correct adversarial examples. These methods aim to improve the model's generalization capabilities, enforce regularization, and incorporate defense mechanisms into the model architecture to improve its robustness against attacks. This chapter offers an overview of recent advances in adversarial attacks on GNNs, including attack methods, evaluation metrics, and their impact on model performance.

DOI: 10.4018/978-1-6684-6903-3.ch005

INTRODUCTION

Graph Neural Networks (GNNs) have emerged as a popular tool for modeling and analyzing complex data structures, such as social networks, biological systems, and infrastructure networks. GNNs learn representations of graph-structured data by propagating information from the neighboring nodes and edges, and have achieved state-of-the-art performance in various tasks such as node classification, link prediction, and graph classification (Zhou et al., 2018). However, the increased use of GNNs has also attracted attention from malicious actors who seek to exploit vulnerabilities in these models. Adversarial attacks on GNNs refer to a class of techniques that aim to manipulate the model's behavior by injecting carefully crafted inputs. These attacks can have serious consequences, including privacy violations, financial losses, and safety risks (Zhao et al., 2021).

Adversarial attacks can be broadly classified into two categories: evasion attacks and poisoning attacks. Evasion attacks aim to manipulate the model's output by modifying the input in a way that is imperceptible to humans but leads to a misclassification or incorrect prediction. Poisoning attacks, on the other hand, aim to modify the training data in a way that alters the model's behavior during inference. In this chapter, we focus on evasion attacks on GNNs. We survey the recent literature on adversarial attacks on GNNs and the countermeasures that have been proposed to mitigate these attacks.

Adversarial Attacks on GNNs

Evasion attacks on GNNs can be broadly categorized into two types: node-level attacks and graph-level attacks. Node-level attacks aim to manipulate the model's output by perturbing the feature vectors of individual nodes in the graph. Graph-level attacks, on the other hand, aim to manipulate the model's output by adding or deleting edges in the graph or by perturbing the graph's global properties.

Node-Level Attacks

One of the most common node-level attacks on GNNs is the perturbation attack (Zügner et al., 2018). In this attack, an adversary adds a small perturbation to the feature vector of a single node in the graph to manipulate the model's output. The perturbation is typically generated by maximizing the loss function with respect to the perturbation subject to a constraint on the maximum allowed Lp-norm of the perturbation (Dai et al., 2018). The resulting perturbation is small enough to be imperceptible to humans but can cause the model to misclassify the node. Figure 1 illustrates an example of a perturbation attack on a GNN.

Another node-level attack is the feature imitation attack (Xu et al., 2019). In this attack, an adversary generates a synthetic feature vector that is similar to the feature vector of a target node but leads to a different output from the GNN. The synthetic feature vector is generated by solving an optimization problem that aims to minimize the distance between the synthetic feature vector and the original feature vector subject to a constraint on the distance between the outputs of the GNN on the original and synthetic feature vectors. This attack can be used to create backdoor attacks on GNNs (Wu et al., 2019).