Adversarial Attack and Defense on Graph Data: A Survey

Lichao Sun, Yingtong Dou, Carl Yang, Ji Wang, Philip S. Yu, *Fellow, IEEE,* Lifang He, and Bo Li,

Abstract—Deep neural networks (DNNs) have been widely applied to various applications, including image classification, text generation, audio recognition, and graph data analysis. However, recent studies have shown that DNNs are vulnerable to adversarial attacks. Though there are several works studying adversarial attack and defense strategies on domains such as images and natural language processing, it is still difficult to directly transfer the learned knowledge to graph structured data due to its representation challenges. Given the importance of graph analysis, an increasing number of works start to analyze the robustness of machine learning models on graph data. Nevertheless, current studies considering adversarial behaviors on graph data usually focus on specific types of attacks with certain assumptions. In addition, each work proposes its own mathematical formulation, which makes the comparison among different methods difficult. Therefore, in this paper, we aim to survey existing adversarial attack and defense strategies on graph data and first provide a unified formulation covering most adversarial learning studies on graph. Moreover, we also compare different attacks and defenses on graph data and discuss their corresponding contributions and limitations. In this work, we systemically organize the considered works based on the features of each topic. This survey not only serves as a reference for the research community but also brings a clear image to researchers outside this research domain. Besides, we also create an online resource and keep updating the relevant works during the last two years. More details of the comparisons of various studies based on this survey are open-sourced at https://github.com/safe-graph/graph-adversarial-learning-literature.

Index Terms—adversarial attack, adversarial defense, adversarial learning, graph data, graph neural networks

1 Introduction

Recent years have witnessed significant success brought by deep neural networks (DNNs) in various domains. Such highly expressive models outperform other models in fields including image recognition [48], natural language processing [32], graph data applications [47], [61], [100], [101], [119], as well as advanced applications such as healthcare analysis [78], brain circuits analysis [69], and functionality of mutations in DNA [124].

Given the outstanding performance, deep learning has been applied in some safety and security critical tasks such as self driving [7], malware detection [90], identification [91] and anomaly detection [37]. However, the lack of interpretability and robustness of DNNs makes them vulnerable to adversarial attacks. Szegedy et al. [95] pointed out the susceptibility of DNNs in image classification. The performance of a well-trained DNN can be significantly degraded by adversarial examples, which are carefully crafted inputs with a small magnitude of perturbations added. Goodfellow et al. [45] analyzed this phenomenon and proposed a gradient-based method (FGSM) to generate adversarial image samples. Different adversarial attack strategies are then proposed to demonstrate the vulnerabilities of DNNs in various settings [6], [16], [123]. For instance, black-box adversarial attacks are later explored based on transferability [70], [79] and query feedback from the DNN models [5], [13]. Several defense and detection methods have also followed up to mitigate such adversarial behaviors [74], [86], while various adaptive attacks continued to be proposed showing that detection/defense is hard in general [3], [15].

Although there are an increasing number of studies on adversarial attack and defense, such adversarial analysis mainly focuses on image, natural language, and speech domains. A related study on graph data is at its infancy despite the importance of graph data in many real-world applications. For example, in the credit prediction application, an adversary can easily disguise himself by adding a friendship connection with others, which may cause severe consequences [28]. Compared with previous adversarial analysis in non-graph data, the study on graph data raises several unique challenges: 1) Unlike images consisting of continuous features, the graph structure and nodes' features are discrete. It is difficult to design efficient algorithms that are able to generate adversarial examples in the discrete spaces. 2) Adversarial perturbations are designed to be imperceptible to humans in the image domain, so one can force a specific distance function, such as L_p norm distance to be small between adversarial and benign instances. However, in graph data, how to define "imperceptible" or "subtle perturbation" requires further analysis, measurement and study.

Given the importance of graph-related applications and the successful applications of graph neural networks

- Lichao Sun, Yingtong Dou, and Philip S. Yu are with the University of Illinois at Chicago, Chicago, IL, 60607 USA. E-mail: {lsun29,ydou5,psyu}@uic.edu
- Ji Wang is with the College of Systems Engineering, National University of Defense Technology, Changsha, Hunan, 410073 P. R. China. E-mail: wangji@nudt.edu.cn.
- Lifang He is with the Lehigh University, PA, USA, 18015. E-mail: lifanghescut@gmail.com.
- Carl Yang and Bo Li are with the University of Illinois Urbana-Champaign at Champaign, IL 61820 USA. E-mail: {jiyang3,lbo}@illinois.edu

TABLE 1
Attack and Defense works are categorized by GNN or Non-GNN oriented.

Category	Type	Paper
Attack Model	GNN	[8], [12], [17], [21], [23], [28], [73], [92], [93], [103], [106], [125], [145], [147] [20], [36], [42], [49], [67], [96], [97], [104], [120], [122], [134], [141], [146]
	Non-GNN	[2], [18], [19], [24], [27], [33], [43], [50], [108], [109], [128], [129], [136], [143] [26], [34], [39]
Defense Model	GNN	[22], [31], [40], [56], [57], [76], [89], [99], [105], [120], [127], [140], [144], [148] [9], [36], [42], [52]–[54], [59], [77], [81], [107], [126], [135], [137], [138]
	Non-GNN	[2], [14], [29], [34], [43], [50], [55], [60], [71], [83], [132], [136], [142]

(GNNs), both academia and industry are interested in the robustness of GNNs. In recent several months, some researchers begin to focus on adversarial attack for a set of GNN models. In this paper, we contribute the first study on summarizing different adversarial learning on graph data and providing taxonomies for them according to various criteria. All relevant attack and defense are listed in Tables 2 and 3. Despite the more than one hundred papers published in the last three years, there are several challenges remaining unsolved until now, which we contribute to summarize and introduce in this work as below.

Comprehensive Understanding. To our best knowledge, this is the first work to provide a comprehensive understanding of adversarial attack and defense on graph data. Our first version of this work was released on Arxiv in 2018, which summarized all published and pre-print works at that time. It has stimulated (and been cited by) various following-up novel research in this line [17], [57], [64], [130], [144] as well as other attempts of survey [25], [58]. In this work, we substantially improve the coverage over a wide range of relevant works, especially those released in the last two years, and we summarize novel elaborated taxonomies according to various criteria. This survey not only include works on adversarial attack and defense approaches targeting specific GNNs, but also discuss many non-gradient and non-model-based approaches in this area.

Online Updating Resource. We created an open-source repository that includes all relevant works and maintained the update on it in the last two years [1]. This repository provides all paper links and released code links, which makes it easier for relevant researchers to use and has served as a major benchmark library in this area. Currently, many of the works are only pre-print versions, and please feel free to contact us when the pre-print papers are accepted in any conference or journal. We will update the information and keep updating the new works in this domain in the future. We hope this open-sourced repository can keep shedding light on future research about adversarial analysis on structured graph data.

Unified Problem Definition. As we know, there have been various attack and defense strategies on graph data. It is a challenge and takes much time to understand the big picture of all works in this domain. In order to facilitate easy understanding over existing research on this line, we pioneer to provide a unified formulation and definition for adversarial attacks on graph data in this work. Unlike attacks, defenses on graph data often go beyond adversarial learning, for which we provide additional categories based on their unique strategies.

Taxonomy of adversarial analysis on graph data. There are already over a hundred papers in this domain. Compared with image data and text data, graph data are more complicated due to various data representations and tasks. List all papers can help but is not good enough for readers to quickly understand the similarity and difference between the works. To this end, we summaries all existing works based on GNN and Non-GNN methods, aiming to help readers find the most relevant papers easily. We present our taxonomy with more details in Table 1.

Dataset and Metrics. Due to the various goals and data of attack and defense works, it is hard to compare the results between each pair of different methods. Currently, no work can directly answer the question "What attack or defense is the best work in this domain (for this task)?". The only way to alleviate this is by building a benchmark like in other areas [30], [102]. In order to address this problem, we not only develop taxonomies for all papers based on different criteria, but also summarize the corresponding datasets and metrics that are frequently used. We hope our work can pave the way for the community to build a good benchmark in this area for future empirical analysis and in-depth technical understanding.

The rest of this survey is organized as follows. Section 2 provides the necessary background information of graph data and common applications. Section 3 provides the unified problem formulation and discusses the existing adversarial attack studies on graph data. Section 4 provides discusses and summaries the existing defense studies on graph data. Section 5 provides the evaluation and attack metrics used in different papers. Section 6 provides the details of each dataset, and summarizes existing works based on the datasets they use. The last section concludes this survey.

2 Graph

In this section, we first give the notations of graph data, and then introduce the preliminaries about graph types, learning settings, and application tasks.

2.1 Notations

We use $\mathcal{G} = \{G_i\}_{i=1}^N$ to represent a set of graphs, where N is the number of graphs. Each graph G_i is generally denoted by a set of nodes $V_i = \{v_j^{(i)}\}$ and edges $E_i = \{e_j^{(i)}\}$, where $e_j^{(i)} = (v_{j,1}^{(i)}, v_{j,2}^{(i)}) \in V_i \times V_i$ is the edge between the nodes $v_{j,1}^{(i)}$ and $v_{j,2}^{(i)}$. Optionally, the nodes and the edges can have

1. https://github.com/safe-graph/graph-adversarial-learning-literature

other features such as node features, edge weights, and edge direction. According to these features, graph data can be classified into different types.

2.2 Types of Graph Data

From a *temporal perspective*, graphs can be grouped into static graphs and dynamic graphs.

Dynamic Graph and Static Graph. A graph is dynamic, denoted as $G^{(t)}$, if any of its nodes, edges, node features, or edges features change over time. In the contrast, a static graph, denoted as G, consists of a fixed set of nodes and edges that do not change over time.

A typical example of static graph is the molecular structure of drugs [35]. Once a drug is developed, its molecular structure does not change over time. Social network [82] is a good example of dynamic graphs. As people often add or remove friendship links in their social network, the graph extracted changes over time. In most existing attack works, the researchers study the attacks on dynamic graphs.

Directed Graph and Undirected Graph. A directed graph, denoted as $G^{(Dr)}$, has direction information associated with each edge, where any directed edge $e_1^{(i)} = (v_1^{(i)}, v_2^{(i)}) \neq (v_2^{(i)}, v_1^{(i)}) = e_2^{(i)}$. In the contrast, any two nodes of an undirected graph share the same edge.

Twitter, an online social network, is one typical example, where the directed edge represents the following information from one user to another. If there is a directed edge connecting from user A to user B, it means A follows B, and necessarily the other way around. The graphs extracted from such online social networks are directed graphs. Facebook is a classic undirected graph that A is B's friend means B is A's friend too.

Attributed Graph on Edge. An attributed graph on edge, denoted as $G^{(A_e)}$, has some features associated with each edge, which is denoted by $x(e_j^{(i)}) \in \mathbb{R}^{D_{edge}}$.

The weighted graph where each edge has a weight, $x(e_j^{(i)}) \in \mathbb{R}$, is a special case of attributed graph on edges. A traffic flow graph [68] is a typical example of weighted graph where roads are modeled as edges and road conditions are represented by weights of edges.

Attributed Graph on Node. An attributed graph on node, denoted as $G^{(A_n)}$, has some features associated with each node, which is denoted by $x(v_j^{(i)}) \in \mathbb{R}^{D_{node}}$.

The e-commerce network [38] with different users can be regarded as an example of attributed graph on node where each user is modeled as nodes with some features like demographics and clicking history.

Note that, directed graph and heterogeneous information networks are special cases of *attributed graph*, which are widely used to model different applications.

2.3 Learning Settings on Graph Data

This section introduces the different machine learning settings used on graph data. Before introducing the learning settings, we first provide the notations for mathematical formulation. We associate the target component c_i within a graph $G^{c_i} \in \mathcal{G}$ with a corresponding ground truth label $y_i \in \mathcal{Y} = \{1, 2, \ldots, Y\}$. Here $i \in [1, K]$, K represents the number of the total target components, and Y is the number of classes being predicted. The dataset $\mathcal{D}^{(ind)} = \{(c_i, G^{c_i}, y_i)\}_{i=1}^{K}$ is represented by the target graph component, graph containing c_i, and the corresponding ground truth label of c_i. For instance, in a node classification task, c_i represents the node to be classified, and y_i denotes its label within G^{c_i}. Based on the features of training and testing processes, the learning settings can be classified as inductive and transductive learning.

Inductive Learning. It is the most realistic machine learning setting where the model is trained by labeled examples, and then predicts the labels of examples never seen during training. Under the supervised inductive learning setting, the classifier $f^{(ind)} \in F^{(ind)} : \mathcal{G} \to \mathcal{Y}$ is optimized:

$$\mathcal{L}^{(ind)} = \frac{1}{K} \sum_{i=1}^{K} \mathcal{L}(f_\theta^{(ind)}(c_i, G^{c_i}), y_i),$$

where $\mathcal{L}(\cdot, \cdot)$ is the cross entropy by default, and c_i can be node, link or subgraph of its associated graph G^{c_i}. Note that, two or more different instances, $c_1, c_2, \ldots, c_n$ can be associated with the same graph $G \in \mathcal{G}$.

Transductive Learning. Different from inductive learning, the testing graphs have been seen during training in the transductive learning. In this case, the classifier $f^{(tra)} \in F^{(tra)} : \mathcal{G} \to \mathcal{Y}$ is optimized:

$$\mathcal{L}^{(tra)} = \frac{1}{K} \sum_{i=1}^{K} \mathcal{L}(f_\theta^{(tra)}(c_i, G^{c_i}), y_i).$$

Transductive learning predicts the label of *seen* instances, but inductive learning predicts the label of *unseen* instances.

Unified Formulation of Learning on Graph Data. We give an uniform formula to represent both supervised inductive and transductive learning as below:

$$\mathcal{L}^{(\cdot)} = \frac{1}{K} \sum_{i=1}^{K} \mathcal{L}(f_\theta^{(\cdot)}(c_i, G^{c_i}), y_i), \tag{1}$$

where $f_\theta^{(\cdot)} = f_\theta^{(ind)}$ is inductive learning and $f^{(\cdot)} = f_\theta^{(tra)}$ is transductive learning.

In the unsupervised learning setting, we can use the unlabelled dataset $\mathcal{D}^{(ind)} = \{(c_i, G_j)\}_{i=1}^{K}$ and replace the supervised loss $\mathcal{L}$ and function $f(c_i, G_i)$ of Eq. 1.

In this survey, we mainly focus on the supervised learning setting, while also introducing a few new works in the unsupervised learning setting.

2.4 Application

In this section, we will introduce the main tasks on graph data, including node-level, link-level and graph-level applications. Moreover, we also introduce how to use the unified formulation of Eq. 1 to define each application task below.

Node-Level Application. The node-level application is the most popular one in both academia and industry. A classic example is labeling the nodes in the Web and social network graphs, which may contain millions of nodes, such as Facebook and Twitter.

Most existing papers [8], [9], [28], [103], [120], [125], [144], [145], [147], [148] focus on node-level applications. All of these papers study node classification in the transductive learning setting whose objective function can be formulated by modifying Eq. 1 where $f_\theta^{(\cdot)} = f_\theta^{(tra)}$, c_i here is the representation of node target and its associated graph G^{c_i} is set as a single graph G.

Few existing works have discussed the node-level applications in the inductive leaning setting. However, these applications frequently appear in real life. For example, the first party only has several large and public network information, such as Facebook and Twitter. The second party has private unlabeled graph data in which the nodes can be predicted by using the information from the first party. In this case, the node-level classification task is no longer transductive learning. It can be easily formulated by modifying Eq. 1 with $f_\theta^{(\cdot)} = f_\theta^{(ind)}$ and c_i here is still the representation of node target.

Link-Level Application. Link prediction on dynamic graphs is one of the most common link-level applications. The models try to predict missing links in current networks, as well as new or dissoluted links in future networks. The corresponding attacks and defenses have been discussed in [92], [143].

Compared with node classification tasks, link predication tasks still use node features, but target at the missing or unlabelled links in the graph. Therefore, we can formulate the link predication task by slightly modifying Eq. 1 with c_i being the representation of link target, and $y_i \in \{0, 1\}$.

Graph-Level Application. Graph-level tasks are frequently seen in the chemistry or medical areas, such as the modeling of drug molecule graphs and brain graphs. In [28], the whole graph is used as the sample instance. Different from this setting, some other graph-level applications use the subgraphs of a larger graph for particular tasks [122], [141].

Compared with the existing works on node classification and link predication, graph classification uses the graph-structure representation as the features to classify the unlabelled graph instances. Therefore, we can formulate the graph classification task by slightly modifying Eq. 1 by setting c_i as the representation of graph target.

3 Adversarial Attacks on Graph Data

In this section, we give a general definition and taxonomies of adversarial attacks on graph data, and then introduce the imperceptibility metrics, attack types, attack tasks and levels of attack knowledge.

3.1 An Unified Definition and Formulation

Definition 3.1. (General Adversarial Attack on Graph Data) Given a dataset $\mathcal{D} = (c_i, G_i, y_i)$, after slightly modifying G_i (denoted as $\widehat{G}^{c_i}$), the adversarial samples $\widehat{G}^{c_i}$ and G_i should be similar under the imperceptibility metrics, but the performance of graph task becomes much worse than before.

Existing papers [8], [17], [21], [27], [28], [36], [50], [67], [92], [103], [120], [125], [145], [147] considering adversarial behaviors on graph data usually focus on specific types of attacks with certain assumptions. In addition, each work proposes its own mathematical formulation which makes the comparison among different methods difficult. In order to help researchers understand the relations between different problems, we propose a unified problem formulation that can cover all current existing works.

Definition 3.2. (Adversarial Attack on Graph Data: A Unified Formulation) f can be any learning task function on graph data, e.g., link prediction, node-level embedding, node-level classification, graph-level embedding and graph-level classification. $\Phi(G_i)$ denotes the space of perturbation on the original graph G_i, and dataset $\widehat{\mathcal{D}} = \{(c_i, \widehat{G}^{c_i}, y_i)\}_{i=1}^N$ denote the attacked instances. The attack can be depicted as,

$$\begin{aligned} \max_{\widehat{G}^{c_i} \in \Phi(G_i)} \quad & \sum_i \mathcal{L}(f_{\theta^*}^{(\cdot)}(c_i, \widehat{G}^{c_i}), y_i)) \\ \text{s.t.} \quad & \theta^* = \arg\min_\theta \sum_j \mathcal{L}(f_\theta^{(\cdot)}(c_j, G'_j), y_j)). \end{aligned} \tag{2}$$

When G'_j equals to $\widehat{G}^{c_j}$, Eq. 4 represents the poisoning attack, whereas when G'_j is the original G without modification, Eq. 4 denotes the evasion attack. $f_\theta^{(\cdot)} = f_\theta^{(ind)}$ represents inductive learning and $f_\theta^{(\cdot)} = f_\theta^{(tra)}$ transductive learning.

Note that, with $\widehat{G}^{c_i} \in \Phi(G)$, $(c_i, \widehat{G}^{c_i})$ can represent node manipulation, edge manipulation, or both. For any $\widehat{G}^{c_i} \in \Phi(G_i)$, $\widehat{G}^{c_i}$ is required to be similar or close to the original graph G_j, and such similarity measurement can be defined by the general distance function below:

$$\begin{aligned} & \mathcal{Q}(\widehat{G}^{c_i}, G_i) < \epsilon \\ \text{s.t.} \quad & \widehat{G}^{c_i} \in \Phi(G_i) \end{aligned} \tag{3}$$

where $\mathcal{Q}(\cdot, \cdot)$ represents the distance function, and ϵ is a parameter denoting the distance/cost budget for each sample.

Discussion: Graph Distance Function. Graph distance functions can be defined in many ways, a lot of which have been discussed on graph privacy-preserving related work [62]. Such distance functions include the number of common neighbours of given nodes, cosine similarity, Jaccard similarity and so on. However, few of them are discussed in depth regarding adversarial behaviors (adversarial cost in game theory). In general, an attacker aims to make "minimal" perturbations on the existing graph and therefore such distance measurement is important to measure the quality of attacks. How to design and choose proper distance function to quantify the attack ability under different attack scenarios is also critical towards developing defensive approaches regarding specific threat model. We will discuss potential perturbation evaluation metrics in detail in Sec 3.2.

In addition to the unique properties of each graph distance function, it would also be interesting to analyze the "equivalence" among them. For instance, an attacker aims to attack one node by adding/removing one edge in the graph can encounter similar "adversarial cost" as adding/removing edges. It is not hard to see that by using a graph distance function or similarity measures, only a few targets would be the optimal choices for the attacker (*with different distance*), so this can also help to optimize the adversarial targets. In summary, due to the complexity and diversity of graph representations and adversarial behaviors, perturbation evaluation or graph similarity measurement will depend on various factors such as different learning tasks, adversarial strategies, and adversarial cost types.

TABLE 2
Summary of adversarial attack works on graph data (time ascending).

Ref.	Year	Venue	Task	Model	Strategy	Approach	Baseline	Metric	Dataset
[27]	2017	CCS	Graph Clustering	SVD, Node2vec, Community detection algs	Noise injection, Small community attack	Add/Delete edges	-	ASR, FPR	NXDOMAIN, Reverse Engineered DGA Domains
[108]	2018	Nature Human Behavior	Hide nodes and communities in a graph	Community detection algs	Heuristic	Rewire edges	-	Concealment measures, Graph statistics	WTC 9/11, Scale-free Facebook, Twitter, Google+, Random
[145]	2018	KDD	Node classification	GCN, CLN, DeepWalk	Incremental attack	Add/Delete edges, Modify node features	Random, FGSM	Accuracy, Classifcation margin	Cora-ML, Citeseer, PolBlogs
[28]	2018	ICML	Graph classification, Node classification	GNN family models	Reinforcement learning	Add/Delete edges	Rnd. sampling, Genetic algs.	Accuracy	Citeseer,Finance, Pubmed, Cora
[109]	2018	Scientific Reports	Link prediction	Similarity measures	Heuristic	Add/Delete edges	-	AUC, AP	WTC 9/11, Random, Scale-Free, Facebook
[23]	2018	arXiv	Node classification, Community detection	DeepWalk, GCN, Node2vec, LINE	Check GCN gradients	Rewire edges	Random, DICE, Nettack	ASR, AML	Cora, Citeseer, PolBlogs
[106]	2018	arXiv	Node classification	GCN	Greedy, GAN	Add fake nodes with fake features	Random, Nettack	Accuracy, F1, ASR	Cora, Citeseer
[92]	2018	arXiv	Link prediction	GAE, DeepWalk, Node2vec, LINE	Project gradient descent	Add/Delete edges	Degree sum, Shortest path, Random, PageRank	AP, Similarity score	Cora, Citeseer, Facebook
[39]	2018	ACSAC	Recommender system	Random walk recommender algs	Optimization	Add nodes&edges	Bandwagon, Co-visitation, Random, Average	HR@N	MovieLens 100K, Amazon Video
[8]	2019	ICML	Node classification, Link prediction	Node2vec, GCN LP, DeepWalk	Check gradient, Approximate spectrum	Add/Delete edges	Random, Degree, Eigenvalue	F1 score, Misclassification rate	Cora, Citeseer, PolBlogs
[147]	2019	ICLR	Node classification	GCN, CLN DeepWalk	Meta learning	Add/Delete edges	DICE, Nettack, First-order attack	Accuracy, Misclassification rate	Cora, Pubmed, Citeseer, PolBlogs
[143]	2019	AAMAS	Link prediction	Local&Global Similarity measures	Submodular	Hide edges	Random, Greedy	Similarity score	Random, Facebook
[18]	2019	TCSS	Community detection	Community detection algs	Genetic algs	Rewire edges	Random, Degree, Community detection	NMI, Modularity	Karate, Dolphin, Football, Polbooks
[103]	2019	CCS	Node classification	LinBP, LBP, JW, DeepWalk, LINE, GCN, RW, Node2vec	Optimization	Add/Delete edges	Random, Nettack	FNR, FPR	Google+, Epinions, Twitter, Facebook, Enron
[136]	2019	IJCAI	Knowledge graph fact plausibility prediction	RESCAL, TransE, TransR	Check target entity embeddings	Add/Delete fact	Random	MRR, Hit Rate@K	FB15k, WN18
[2]	2019	arXiv	Vertex nomination	VN·GMM·ASE	Random	Add/Delete edges	-	Achieving rank	Bing entity transition graph
[12]	2019	arXiv	Node classification	GCN	Adversarial generation	Modify node features	Nettack	ASR	Cora, Citeseer
[120]	2019	IJCAI	Node classification	GCN	Check gradients	Add/Delete edges, Modify node features	Random, Nettack FGSM, JSMA	Accuracy, Classification margin	Cora, Citeseer, PolBlogs
[125]	2019	IJCAI	Node Classification	GCN	First-order optimization	Add/Delete edges	DICE, Greedy, Meta-self	Misclassification rate	Cora, Citeseer
[17]	2019	AAAI	Node classification	GCN, LINE, SGC, DeepWalk	Approximate spectrum, Devise new loss	Add/Delete edges	Random, Degree, RL-S2V,	Accuracy	Cora, Citeseer, Pubmed
[73]	2019	arXiv	Node classification	GCN	Reinforcement learning	Rewire edges	RL-S2V, Random	ASR	Reddit-Multi, IMDB-Multi
[50]	2019	CIKM	Malware detection, Node classification	Metapath2vec	Greedy	Inject new nodes	Anonymous attack	%TPR, TP-FP curve	Private dataset
[24]	2019	arXiv	Dynamic link prediction	Deep dynamic network embedding algs	Check gradients	Rewire edges	Random, Gradient, Common neighbor	ASR, AML	LKML, FB-WOSN, RADOSLAW
[33]	2020	AAMAS	Node Similarity	Similarity measures	Graph theory	Remove edges	Greedy, Random, High jaccard similarity	# Removed edges	Power,web-edu, hamsterster, euroroad
[93]	2020	WWW	Node classification	GCN	Reinforcement learning	Inject new nodes	Random, FGA, Preferential attack	Accuracy, Graph statistics	Cora-ML, Pubmed, Citeseer
[67]	2020	WWW	Hide node in community	Surrogate community detection model	Graph auto-encoder	Add/Delete edges	DICE, Random, Modularity based attack	Personalized metric	DBLP, Finance
[36]	2020	WSDM	Node classification	GCN, t-PINE	Low-rank approximation	Add/Delete edges	Nettack	Correct classification rate	Cora-ML, Citeseer, PolBlogs
[134]	2020	arXiv	Node classification	GCN, DeepWalk, Node2vec, GAT	Check gradients	Add/Delete edges	Random, FGA, Victim-class attack	ASR, AML	Cora, Citeseer, PolBlogs
[96]	2020	BigData	Node classification	GCN	Check gradients	Modify node features	Nettack	ASR	Cora-ML, Citeseer
[43]	2020	arXiv	Manipulating opinion	Graph model	Adversarial optimization	Change initial opinion vector	-	-	-
[146]	2020	TKDD	Node classification	GCN, CLN, DeepWalk	Incremental attack	Add/Delete edges, Modify node features	Random, FGSM	Accuracy, Classifcation margin	Cora-ML, Citeseer, PolBlogs, Pubmed
[97]	2020	arXiv	Graph classification	HGP	Surrogate model	Add/Delete edges, Modify node features	-	Accuracy, Error rate	ER_MD, BZR Mutagenicity, DD DHFR, AIDS
[34]	2020	KDD	Fraud detection	Graph-based Fraud detectors	Reinforcement learning	Add/Delete edges	-	Practical effect	YelpChi, YelpNYC, YelpZip
[141]	2020	arXiv	Graph classification	GIN	Graph generation	Inject subgraphs	-	ASR, Clean accuracy, Backdoor accuracy	Twitter, Bitcoin, COLLAB

3.2 Adversarial Perturbation

To generate adversarial samples on graph data, we can modify the nodes or edges from the original graph. However, the modified graph $\widehat{G}$ need to be "similar" with the original graph G based on certain perturbation evaluation metrics and remain "imperceptible". The following metrics help understand how to define "imperceptible perturbation".

Edge-level Perturbation. In most current papers, the attacker is capable of adding/removing/rewiring edges in the whole original graph within a given budget. In this case, the number of modified edges is usually used to evaluate the magnitude of perturbation. In addition to other perturbations, edge perturbation is hardly found by the defender, especially in dynamic graphs.

Node-level Perturbation. The attacker is also capable of adding/removing nodes, or manipulating the features of target nodes. The evaluation metric in this case can be calculated based on the number of nodes modified or the distance between the benign and adversarial feature vectors.

Structure Preserving Perturbation. Similar to edge-level perturbation, an attacker can modify edges in the graph within a given budget in terms of graph structure. Compared to general edge-level perturbation, this considers more structural preservation, such as total degree, node distribution, etc. For instance, in [145], the attacker is required to preserve the key structural features of a graph such as the degree distribution. Therefore, the perturbation here can be measured by the graph structure drift.

Attribute Preserving Perturbation. In the attributed graphs, each node or edge has its own features. In addition to manipulating the graph structure, the attacker can choose to modify the features of nodes or edges to generate adversarial samples on graph data. Various measurements based on graph-attribute properties can be analyzed to characterize the perturbation magnitude. For instance, in [145], the authors argue adding a feature is imperceptible if a probabilistic random walker on the co-occurrence graph can reach it with high probability by starting from existing features.

Note that, most GNN methods learn the feature representation of each node, which means it could be easily attacked by structure-only, feature-only perturbations or both.

Principles of imperceptible perturbation evaluation. Given various graph distance discussion, there is no clear discussion in existing research about how to set the adversarial cost for attacks on graph data so far. Therefore, we summarize some principles of defining the perturbation evaluation metrics as below for future research.

- For static graph, both the number of modified edges and the distance between the benign and adversarial feature vectors should be small.
- For a dynamic graph, we can set the distance or adversarial cost based on the intrinsic changing information over time. For example, by using statistic analysis, we can get the upper bound of the information manipulated in practice, and use this information to set an imperceptible bound.
- For various learning tasks on graph data, e.g., node or graph classification, we need to use a suitable graph distance function to calculate the similarity between the benign and its adversarial sample. For example, we can use the number of common neighbours to evaluate the similarity of two nodes, but this is not applicable for two individual graphs.

In summary, compared to image and text data, an attacker first can modify more features on the information network, and also can explore more angles to define "imperceptible" based on the format of graph data and the application task.

3.3 Attack Stage

The adversarial attacks can happen at two stages: evasion attack (model testing) and poisoning attacks (model training). It depends on the attacker's capacity to insert adversarial perturbations:

Poisoning Attack. Poisoning attack tries to affect the performance of the model by adding adversarial samples into the training dataset. Most existing works are poisoning attacks, and their node classification tasks are performed in the transductive learning setting. In this case, once the attacker changes the data, the model is retrained. Mathematically, by setting $G'_j = \widehat{G}^{c_j}$ in Eq. 4, we have a general formula for adversarial attack on graph data under poisoning attacks.

Evasion Attack. Evasion attack means that the parameters of the trained model are assumed to be fixed. The attacker tries to generate the adversarial samples of the trained model. Evasion attack only changes the testing data, which does not require to retrain the model. Mathematically, by setting G'_j to original G_j in Eq. 4, we have a general formula for adversarial attack on graph data under evasion attacks.

3.4 Attack Objective

Though all adversarial attacks are modifying the data, an attacker needs to choose their attack targets or objectives: model or data. In this case, we can summarize them as model objective and data objective.

Model Objective. Model objective is attacking a particular model by using any approaches. It could be either evasion attack or poisoning attack. Most current adversarial attack is related to model objective attack. The target could be either GNN or other learning models. An attacker wants to make the model become non-functional working in multiple scenarios. Model objective attack can be categorized by whether using the gradient information of the model or not.

- **Gradient-based Attack.** In most studies, we can see that the gradient-based attack is always the simplest and most effective approach. Most gradient-based attack, no matter white-box or black-box, tries to get or estimate the gradient information to find the most important features to the model. Based on the above knowledge, an attacker can choose to modify the limited information based on the feature importance to the model and make the model inaccurate when using the modified information [8], [28], [145].
- **Non-gradient-based Attack.** In addition to gradient information, an attack could destroy the model without any gradient information. As we know, besides the gradients, many reinforcement learning based

attack methods can attack the model based on long-term rewards [28], [73], [92]. Some works can also construct the adversarial samples with generative models [12], [19], [42]. All the above approaches can attack the model without the gradient information but attack the model in practice.

Data Objective. Unlike model objective attacks, data objective attacks do not attack a specific model. Such attacks happen when the attacker only has access to the data, but does not have enough information about the model. In general there are two settings when data become the target.

- **Model Poisoning.** Unsupervised feature analysis approaches can still get useful information from the data without any knowledge of the training approach. Even with a small perturbation on the data, it can make general training approaches cease to work. Besides, backdoor attack is another relevant hot topic where an attacker only injects the adversarial signals in the dataset, but does not destroy the model performance on regular samples [122], [141].
- **Statistic Information.** In addition to using the data to train a model, in many studies, researchers use statistical results or simulation results from the graph data. In this case, an attacker can break the model based on the capturing of the valuable statistical information on graph data. For example, by modifying a few edges between different communities based on structural information and analysis, one can make communities counting inaccurate under this attack.

3.5 Attack Knowledge

The attacker would receive different information to attack the system. Based on this, we can characterize the dangerous levels of existing attacks.

While-box Attack. In this case, an attacker can get all information and use it to attack the system, such as the prediction result, gradient information, etc. The attack may not work if the attacker does not fully break the system first.

Grey-box Attack. An attacker gets limited information to attack the system. Comparing to white-box attack, it is more dangerous to the system, since the attacker only need partial information.

Black-box Attack. Under this setting, an attacker can only do black-box queries on some of the samples. Thus, the attacker generally can not do poisoning attack on the trained model. However, if black-box attack can work, it would be the most dangerous attack compared with the other two, because the attacker can attack the model with the most limited acknowledge.

Most existing papers only studies white-box attack on the graph, and there are lots of opportunities to study other attacks with different levels of knowledge.

3.6 Attack Goal

Generally, an attacker wants to destroy the performance of the whole system, but sometimes they prefer to attack a few important target instances in the system. Based on the goal of an attack, we have:

Availability Attack. The adversarial goal of availability attack is to reduce the total performance of the system. For example, by giving a modification budget, we want the performance of the system decreasing the most as the optimal attack strategy.

Integrity Attack. The adversarial goal of integrity attack is to reduce the performance of target instances. For example, in recommendation systems, we want the model to not successfully predict the hidden relation between two target users. However, the total performance of the system is the same or similar to the original system.

Availability attack is easier to detect than integrity attack under the positioning attack setting. Therefore, meaningful availability attack studies are in general under the evasion attack setting.

3.7 Attack Task

Corresponding to various tasks on graph data, we show how to attack each task and explain the general idea by modifying the unified formulation.

Node-relevant Task. As mentioned before, most attack papers focus on node-level tasks, including node classification [17], [28], [103], [120], [125], [145], [147] and node embedding [8], [136]. The main difference is that node embedding uses the low dimensional representations of each node for an adversarial attack. Mathematically, by setting c_i as representation of node target in Eq. 4, we have a general formula for adversarial attack on node-relevant tasks.

Link-relevant Task. Other several existing works study node embedding [8], [21], [92] or topological similarity [109], [143] and use them for link prediction. Compared with node classification, link prediction requires to use different input data, where c_i represents link target, i.e., the information of a pair of nodes. By setting c_i as representation of link target and $y_i \in [0, 1]$ in Eq. 4, we have a general formula for adversarial attack on link-relevant tasks.

Graph-relevant Task. Compared with node classification, graph classification needs the graph representation instead of the node representation [28], [97], [122], [141]. By setting c_i as representation of graph target in Eq. 4, we have a general formula for adversarial attack on graph-relevant tasks.

3.8 Summary: Attack on Graph

In this subsection, we talk about the contributions and limitations of existing works. Then we discuss the potential research opportunities in this area.

Contributions. First, we list all released papers and their characteristics in Table 2, and then categorize them into selected main topics in Table 1. Then, we summarize the unique contributions of existing adversarial attacks. Note that, because 11 of 34 papers we discuss are pre-print version, we especially list the venue in Table 2. We also firstly use *Strategy* and *Approach* to differ individual attack method. *Strategy* refers to the high-level design philosophy of an attack, while *Approach* represents the concrete approach the attacker takes to perturb the graph data.

Graph Neural Networks. Most adversarial attacks are relevant to graph neural networks. [28] used reinforcement learning approach to discover adversarial attack, which is the only approach that supports black-box attack compared to other works. [145] studied adversarial graph samples with traditional machine learning and deep learning. Meanwhile, they are the first and only group to discuss the adversarial attack on attributed graph. [21], [92] mainly attacked the link predication task with a deep graph convolutional embedding model. [8] attacked multiple models by approximating the spectrum and use the gradient information. [103] attacked node classification though optimization approach and systematically discussed adversarial attacks on graph data. Previous works focused on edge or node modification, whereas [120] also modified the node features and proposed a hybrid attack on the graph convolutional neural networks (GCN) [61]. In addition to gradient check, [36], [125] attacked GCN by using the first-gradient optimization and low-rank approximation which makes an attack more efficient. [17] attacked general learning approaches by devising new loss and approximating the spectrum. [50] used graph attack knowledge into the malware detection problem, which showed various graph-based applications to be vulnerable to adversarial attacks. Without gradient check and optimization design, [93] used reinforcement learning to attack GCN. However, it contains an obvious issue that it needs to break the graph structure by injecting new nodes. [67] tried to hide nodes in the community by attacking the graph auto-encoder model. Instead of using a gradient check or other optimization approaches, this work leverage the surrogate community detection model to achieve the attacking goal. More recent works investigates the vulnerability of GNNs under backdoor attacks [122], [141]. Backdoor attack modifies the labels of the triggers (e.g., subgraphs with typical patterns) in the training data, and it aims to make the GNNs misclassify those triggers without affecting the overall performance of GNNs on the testing data.

Others. Though many attack works are relevant to GNN, many recent papers start to focus on other types of adversarial attacks on graph data. [27] is one of the first works to attack the graph data, and it also first proposed the attack approach in the unsupervised learning setting. [108] first attacked community detection though edge rewriting based on a heuristic approach. [109] attacked link prediction based on a heuristic approach which is based on the similarity measures. [143] used a greedy approach to attack link prediction based local and global similarity measure. In addition to traditional graph applications, [136] first attacked knowledge graph and destroyed the basic relational graph prediction model. [18] attacked community detection based on genetic algorithms. Unlike previous approaches, it chose to use rewiring instead of adding/removing edges while attacking the data. [42] used a generation approach to create a new isomorphism network to attack node classification. In addition to all previous works, [33] started to study attacks through theoretical analysis, and we believe more theoretical works will be seen in this domain. They can help us understand the attacks better on graph data. Besides the applications mentioned above, attacking graphs in recommender system [39], [81], [137], fraud detection [14], [34], opinion dynamic [26], [43], and graph classification [97], [122], [141] tasks have been drawing attention from researchers as well.

Limitations. The limitations of most current works are summarized below. Most existing works do not give very clear strategies about the setting of the budget and distance with reasonable explanations in real applications. Different from other adversarial attacks, most graph modifications can hardly be noticed by humans in real life. To solve this problem, we give a more detailed discussion on perturbation and evaluation metrics in Section 5. Meanwhile, about graph imperceptible evaluation metrics, most papers [8], [21], [28] use one metric for attack, but these adversarial samples could be detected by other existing imperceptible evaluation metrics. In this work, we list all existing evaluation metrics, and recommend future adversarial samples to be imperceptible with more listed evaluation metrics. Another main issue is due to the different problem formulations. To this end, we give the unified problem formulation for all existing works discussed in this survey.

Future Directions. Adversarial attack on graph data is a new and hot area, and many research opportunities are summarized below: 1) Most graphs are associated with attributes or more complex contents on nodes or edges in the real life. Currently, very few existing works well studied adversarial attack on attributed graphs, e.g., heterogeneous information networks and the Web. 2) Some advanced ideas can be applied for generating the adversarial samples, e.g., homomorphism graph. 3) Various learning settings are not sufficiently studied yet, such as graph-level attacks and inductive learning on node-level attacks. 4) Most existing attacks do not consider various imperceptibility metrics into their attack model. Concise and comprehensive imperceptibility metrics are necessary in different tasks. A good and explainable evaluation metric may easily discover more existing adversarial samples created by current methods. 5) Last but not least, the distance or similarity measures of high quality adversarial samples are not well studied in this area.

4 Adversarial Defense on Graph Data

With graph data, recent intensive studies on adversarial attacks have also triggered the research on adversarial defenses. Here we survey existing works in this line and classify them into the two popular categories of *Adversarial Training* and *Attack Detection*. After them, we use an additional *Other Methods* subsection to summarize the remaining methods that do not fit into the two generic categories.

4.1 Adversarial Training

While adversarial training has been widely used by attackers to perform effective adversarial intrusion, the same sword can be used by defenders to improve the robustness of their models against adversarial attacks [45]. In the graph setting, we formulate the objective of adversarial defense by slightly modifying our unified formulation of adversarial attacks, *i.e.*, Eq. 4, as follows

$$\min_{\theta} \max_{\widehat{G}^{c_i} \in \Phi(G_i)} \sum_{i} \mathcal{L}(f_\theta(c_i, \widehat{G}^{c_i}), y_i)). \tag{4}$$

where meanings of the notations remain the same as defined in Section 3. The idea is to alternatively optimize two competing modules during training, where the attacker tries to maximize task-oriented loss by generating adversarial

TABLE 3
Summary of adversarial defense works on graph data (time ascending).

Ref.	Year	Venue	Task	Model	Corresp. Attack	Strategy	Baseline	Metric	Dataset
[127]	2018	Open Review	Added edges detection	GNN, GCN	-	Link prediction, Graph generation	LP	AUC	Cora, Citeseer
[40]	2019	TKDE	Node classification	GCN	-	Adversarial training	DeepWalk, GCN, Planetoid, LP, GraphVAT, GraphSCAN	Accuracy	Cora, NELL, Citeseer
[140]	2019	ICLR Workshop	Node classification	GCN, GAT	Nettack, Random	First&Second order KL divergence proximity	-	Classification margin, Accuracy, AUC	Cora, Citeseer, PolBlogs
[29]	2019	WWW	Node classification	DeepWalk	-	Adversarial training	DeepWalk, LINE Node2vec, GraRep, Graph Factorization	Accuracy, AUC	Cora, Wiki Citeseer, CA-GrQc, CA-HepTh
[89]	2019	PRCV	Node classification	GCN	-	Virtual adversarial training	GCN	Accuracy	Cora, Citeseer, Pubmed
[22]	2019	arXiv	Node embedding	GNN	Nettack, FGA	Smoothing gradients	Adversarial training	ADR, ACD	Cora, Citeseer, PolBlogs
[144]	2019	KDD	Node classification	GCN	Nettack, RL-S2V, Random	Gaussian distribution layer, Variance-based attention	GCN, GAT	Accuracy	Cora, Citeseer, Pubmed
[83]	2019	NAACL	Link prediction	Knowledge graph embeddings	-	Adversarial modification	-	Hits@K, MRR	Nations, WN18, Kinship, YAGO3-10
[2]	2019	arXiv	Vertex nomination	Graph embedding models	-	Network regularization with graph trimming	-	Achieving rank	Bing entity transition graph
[105]	2019	arXiv	Node classification	GCN	Nettack, RL-S2V, Random	GAN, Graph encoder refining, Contrastive learning	GCN, GraphSAGE, Refined GCN&GraphSAGE	Classification margin	Cora, Citeseer, PolBlogs
[57]	2019	arXiv	Node classification	GCN	-	Graph powering	GCN, ICA, MeniReg	Accuracy, Robustness merit, Attack deterioration	Cora, Pubmed, Citeseer
[120]	2019	IJCAI	Node classification	GCN	Random, Nettack FGSM, JSMA	Drop edges	GCN	Accuracy, Classification margin	Cora, Citeseer, PolBlogs
[125]	2019	IJCAI	Node classification	GCN	DICE, Meta-self	Check gradients, Adversarial training	GCN	Accuracy, Misclassification rate	Cora, Citeseer
[56]	2019	ICML Workshop	Node classification	GCN	Nettack	Adversarial training	GCN, SGCN, FastGCN, SGC	ASR, Accuracy	Citeseer, Cora, Pubmed, Cora-ML, DBLP, PolBlogs
[31]	2019	ICML Workshop	Node classification	GCN	-	Adversarial training	GCN, GAT, LP, DeepWalk, Planetoid, Monet, GPNN	Accuracy	Citeseer, Cora, Pubmed, NELL
[132]	2019	TKDE	Link prediction	Link prediction methods	Resource Allocation Index	Estimation of Distribution Algorithm	RLR, RLS HP, GA	Precision, AUC	Mexican, Dolphin Bomb, Lesmis, Throne, Jazz
[148]	2019	KDD	Node classification	GCN, GNN	-	Convex optimization	GNN	Accuracy, Average worst-case margin	Cora-ML, Pubmed, Citeseer
[76]	2019	KDD Workshop	Node classification	GCN, Node2vec	-	Change training set	GCN, Node2vec	Adversary budget, Classification margin	Cora, Citeseer
[142]	2019	ICDM	Link prediction	Similarity measures	-	Bayesian Stackelberg game and optimization	Protect Potential Neighbors	Damage prevention ratio	PA, TV Show, PLD, Gov
[9]	2019	NIPS	Node classification	GCN	-	Robust training, MDP to get bound	GNN	Accuracy, Worst-case margin	Cora-ML, Pubmed, Citeseer
[50]	2019	CIKM	Malware detection, Node classification	Heterogeneous graph, Metapath2vec	-	Attention mechanism	Other malware detection algs	Accuracy, F1, Precision, Recall	Private dataset
[107]	2019	arXiv	Node classification	GCN, GraphSAGE	-	Adversarial training	Drop edges, Discrete adversarial training	Accuracy, Correct classification rate	Cora, Citeseer, Reddit
[99]	2020	WSDM	Node classification	GNN	Metattack	Meta learning, Transfer from clean graph	GCN, GAT, RGCN, VPN	Accuracy	Pubmed, Yelp, Reddit
[36]	2020	WSDM	Node classification	GCN, t-PINE	Nettack, LowBlow	Low-rank approximation	-	Correct classification rate	Cora-ML, Citeseer, PolBlogs
[55]	2020	WWW	Community detection	Community detection algs	-	Robust certification with optimization	-	Certified accuracy	Email, DBLP, Amazon
[14]	2020	WWW	Fraud Detection	Graph-based Sybil detectors	Change label, Graph generation	Probability estimation	VoteTrust, SybilRank, SybilSCAR, SybilBelief	AUC	Facebook, Synthetic graphs
[43]	2020	arXiv	Manipulating opinion	Graph model	-	Minimax game, Convex optimization	-	-	-
[137]	2020	SIGIR	Recommender system	GCN	Mixed, Hate, Average, Random	Fraud detection	RCF, GCMC, GraphRec, MF, AutoRec, PMF	RMSE, MAE	Yelp, Moive&TV
[59]	2020	KDD	Node classification	GNN	Nettack, Meta-self, Random	Graph structure learning	GCN, GCN-SVD, RGCN, GAT, GCN-Jaccard	Accuracy	Cora, Pubmed, Polblogs, Citeseer
[34]	2020	KDD	Fraud detection	Graph-based Fraud detectors	IncBP, IncDS, IncPR, Random, Singleton	Minimax game, Reinforcement learning	SpEagle, GANG Fraudar, fBox	Practical effect	YelpChi, YelpNYC, YelpZip
[141]	2020	arXiv	Graph classification	GIN	Graph generation	Randomized subsampling	-	ASR, Clean accuracy, Backdoor accuracy	Twitter, Bitcoin, COLLAB

perturbations $\widehat{G}$ on the graph, and the defender tries to minimize the same loss by learning the more robust graph model parameters θ under the generated adversarial perturbations. In this way, the learned graph model is expected to be resistant to future adversarial attacks.

Structure Perturbations. The earliest and most primitive way of perturbing the graph is to randomly drop edges [28]. The joint training of such cheap adversarial perturbations is shown to slightly improve the robustness of standard GNN models towards both graph and node classification tasks. One step further, [125] proposed a topology attack generation method based on projected gradient descent to optimize edge perturbation. The topology attack is shown to improve the robustness of the adversarially trained GNN models against different gradient-based attacks and greedy attacks without sacrificing node classification accuracy on the original graph. In the meantime, [29] proposed to learn the perturbations in an unsupervised fashion by maximizing the influence of random noises in the embedding space, which improved the generalization performance of DeepWalk [82] on node classification. Towards similarity-based link prediction, [142] formalized a Bayesian Stackelberg game to optimize the most robust links to preserve with an adversary deleting the remaining links.

Attribute Perturbations. Besides links, [31], [40], [89] also perturb node features to enable virtual adversarial training that enforces the smoothness between original nodes and adversarial nodes. In particular, [40] designed a dynamic regularizer forcing GNN models to learn to prevent the propagation of perturbations on graphs, whereas [89] smooths GCN in its most sensitive directions to improve generalization. [31] further conducts virtual adversarial training in batch to perceive the connectivity patterns between nodes in each sampled subsets. [105] leveraged adversarial contrastive learning to tackle the vulnerabilities of GNN models to adversarial attacks due to training data scarcity and applied conditional GAN to utilize graph-level auxiliary information. Instead of approximating the discrete graph space, [107] proposed to directly perturb the adjacency matrix and feature matrix by ignoring the discreteness, whereas [56] proposed to focus on the first hidden layer of GNN models to continuously perturb the adjacency matrix and feature matrix. These frameworks are all shown to improve GNN models on the node classification task.

Attack-oriented Perturbation Based on existing network adversarial attack methods of FGA [23] and Nettack [145], [22] designed the adversarial training pipelines with additional smooth defense strategies. The pipeline is shown to improve GNN models against different adversarial attacks on node classification and community detection tasks. [34] employed reinforcement learning to train a robust detector against mixed attacks proposed in the paper.

4.2 Attack Detection

Instead of generating adversarial attacks during training, another effective way of defense is to detect and remove (or reduce the effect of) attacks, under the assumption that data have already been polluted. Due to the complexity of graph data, the connection structures and auxiliary features can be leveraged based on various ad hoc yet intuitive principles to essentially differentiate clean data from poison ones and combat certain types of attacks.

Graph Preprocessing. [127] proposed different approaches to detect potential malicious edges based on graph generation models, link prediction and outlier detection. Instead of edges, [52] proposed to filter out sets contaminated by anomalous nodes based on graph-aware criteria computed on randomly drawn subsets of nodes; [140] proposed to detect nodes subject to topological perturbations (particularly by Nettack [145]) based on empirical analysis on the discrepancy between the proximity distributions of nodes and their neighbors. These models only rely on network topology for attack detection. On attributed graphs, based on the observations that attackers prefer adding edges over removing edges and the edges are often added between dissimilar nodes, [120] proposed to compute the Jaccard Similarity to remove suspicious edges between suspicious nodes. [126] sampled sub-graphs from the poisoned training data and then employed outlier detection methods to detect and filter adversarial edges. All of these models can be used for graph preprocessing before training normal graph models like GNNs.

Model Training. Rather than direct detection of suspicious nodes or edges before training, several works designed specific attention mechanisms to dynamically uncover and down-weigh suspicious data during training. [144] assumed high prediction uncertainty for adversarial nodes and computed the attention weights based on the embedding variance in a Gaussian-based GCN. [99] suggested to train an attack-aware GCN based on ground-truth poisoned links generated by Nettack [145] and transfer the ability to assign small attention weights to poisoned links based on meta-learning.

Robustness Certification. On the contrary of detecting attacks, [9], [148] designed robustness certificates to measure the safety of individual nodes under adversarial perturbation. In particular, [9] considered structural perturbation and [148] considers attribute perturbation. Training GNN models jointly with these certificates can lead to a rigorous safety guarantee of more nodes. From a different perspective, [55] derived the robustness certificate of community detection methods under structural perturbation. [60] proved polynomial spectral graph filters are stable under structural perturbation.

Complex Graphs Beyond traditional homogeneous graphs, [83] studied the sensitivity of knowledge graph link prediction models towards adversarial facts (links) and the identification of facts. [50] studied the detection of poisoning nodes in heterogeneous graphs to enhance the robustness of Android malware detection systems.

4.3 Other Methods

Now we summarize the remaining graph adversarial defense algorithms that are neither based on adversarial training nor aiming at attack detection. We further group them into three subcategories based on their modifications to the graph data and graph models.

Data Modifications. We have presented several attack detection algorithms that can be used for modifying graph data, *i.e.*, graph preprocessing [52], [127], [140]. There exist methods that modify graph data without directly detecting attacks. Based on the insight that Nettack [145] only affects the high-rank singular components of the graph, [36] proposed to reduce the effect of attacks by computing the low-rank approximation of the graphs before training GNN models. [42] proposed an augmented training procedure by generating more structurally noisy graphs to train GNN models for improved robustness, and showed it to be effective for structural role identification of nodes. [77] analyzed the topological characteristics of graphs and proposed two training data selection techniques to raise the difficulty of effective adversarial perturbations towards node classification. These methods are all based on graph topology alone, and they only modify the graph data instead of the graph models. [135] leveraged variational graph autoencoders to reconstruct graph structures from perturbed graphs and the reconstructed graphs can reduce the effects of adversarial perturbations.

Model Modifications. On the contrary, there exist methods that only modify the graph models, such as model-structure redesign or loss-function redesign. The simplest way is to redesign the loss function. From several existing works, the results show some loss functions perform better performance against the adversarial examples. For example, [57] designed an alternative operator based on graph powering to replace the classical Laplacian in GNN models with improved spectral robustness. They demonstrated the combination of this operator with vanilla GCN to be effective in node classification and defense against evasion attacks. [81] proposed a hierarchical GCN model to aggregate neighbors from different orders and randomly dropped neighbor messages during the aggregation. Such mechanism could improve the robustness of GCN-based collaborative filtering models. [138] introduced neighbor importance estimation and the layer-wise graph memory components which can be integrated with GNNs. Those two components could help increase the robustness of GNN models agaisnt various attacks.

Hybrid Modifications. One step further, some methods modify both the graph data and graph models. [53] designed an edge-dithering approach to restoring unperturbed node neighborhoods with multiple randomly edge-flipped graphs and proposed an adaptive GCN model that learns to combine the multiple graphs. The proposed framework is shown to improve the performance and robustness of GCN towards node classification (in particular, protein function prediction) on attributed graphs. [76] proposed a heuristic method to iteratively select training data based on the degrees and connection patterns of nodes. They further proposed to combine node attributes and structural features and use SVM for node classification instead of any GNN models. Guided by graph properties like sparsity, rank, and feature smoothness, [59] presented Pro-GNN which jointly learns clean graph structure and trains robust GNN models together.

4.4 Summary: Defense on Graph

From the perspective of defenders, the defense approaches can be designed with or without knowing the specific attacks. Thus, current defense works can be classified into two categories: 1) *Attack-agnostic defenses* are designed to enhance the robustness of graph models against any possible attacks instead of a fixed one. 2) *Attack-oriented defenses* are designed according to the characteristics of specific attacks. The attack-agnostic defenses usually have a wider assumption space of attacks comparing to attack-oriented attack. Last, we discuss some future opportunities on adversarial defense in this area.

Attack-agnostic Defense. As we summarized in Section 4.1, adversarial training is a typical instance of attack-agnostic defense approach [28], [31], [40], [89], [125]. It usually generates simple perturbations on graphs or models to train a defense model. In the test phase, some models trained in this way could exhibit good robustness against those perturbations. Some methods [125] trained in this way even attain good defense performance against other specific attacks like Meta-self proposed in [147]. Note that the defense methods are designed and trained without knowing other new attacks.

Besides adversarial training, other works secure the graph model with heuristic assumptions on the attack strategies and outcomes. [99] assumes that there are unpolluted graphs to aid the detection of attacks. [50], [54], [57], [144] propose new GNN architectures to enhance their robustness. [76], [77] directly curates an optimal training set to mitigate the vulnerability of trained models.

Attack-oriented Defense. Attack-oriented defenses are designed based on the strategy and approach of specific attacks. Namely, the defender has full knowledge of an attack method and the defense method could detect the corresponding attack or curb its performance. Among current defense works, [36] first argues the weakness of Nettack [145] and leverages SVD to defend against Nettack. [56] analyzes the strategies and approaches of Nettack [145] and RL-S2V [28] and propose an adversarial training method. [120] inspects two gradient-based attack (i.e., FGSM [45] and JSMA [80]) and applies edge-dropping technique during model training to alleviate the influence of such attacks. Similar to attack-agnostic defenses, some attack-oriented methods exhibit good generability which means it can defend against other unknown attacks. For instance, the defense method proposed in [120] could defend the Nettack as well. Along with the **Corresp. Attack** column of Table 3, we could see that Nettack and RL-S2V have become benchmark attack methods for defense design and evaluation. Some works employ the framework of minimax game [43] or optimization [9], [55], [148] to certify the robustness bounds of graph models under given attacks and defenses. Such kind of defense works are attack-oriented since they have assumed specific attacks.

Limitations and future directions. We have been focusing on the contributions of different existing works on graph adversarial defense. Now we summarize some common limitations we observe in this line of research and hint on future directions: 1) Most defense models focus on node-level tasks, especially node classification, while it may be intriguing to shed more light on link- and graph-level tasks like link prediction and graph classification. There is also large potential in more real-life tasks like graph-based search, recommendation, advertisement and etc. 2) While network data are often associated with complex contents nowadays (e.g., timestamps, images, texts), existing defense models have hardly considered the effect of attacks and defenses under the settings of dynamic or other content-rich complex networked systems. 3) Most defense models are relevant to GNNs or GCN in particular, but there are many other graph models and analysis methods, possibly more widely used and less studied (e.g., random walk based models, stochastic block models, and many computational graph properties). How are they sensitive and prone to graph adversarial attacks? Can the improvements in GNN models transfer and generalize to these traditional methods and measures? 4) Most existing works do not study the efficiency and scalability of defense models. As we know, real-world networks can be massive and often frequently evolve, so how to efficiently learn the models and adapt to changes is very important for defenders. 5) While there are standard evaluation protocols and optimization goals for down-stream tasks like node classification and link prediction, defense methods are optimized towards heterogeneous goals like accuracy, robustness, generalizability and so on, and they tend to define their own experimental settings and metrics, rendering fair and comprehensive evaluations challenging.

5 METRIC

In this section, we summarize the metrics for evaluating attack and defense performance on graph data. We first briefly introduce the general evaluation metrics along with some notes on their specific usage in adversarial performance evaluation. We then give a detailed introduction of particular evaluation metrics designed for attacks and defenses.

5.1 General Metric

5.1.1 Accuracy-based Metric

According to Table 2 and Table 3, many existing works tackle the node classification problem which is usually a *binary* or *multi-class* classification problem. The accuracy-based metrics like **Accuracy**, **Recall**, **Precision**, and **F1 score** are all used by existing works to reflect the classification accuracy from different angles. Readers can refer to [112] for detailed explanations of those metrics. Note that the **False Negative Rate** (**FNR**) and **False Positive Rate** (**FPR**) used by [27], [103] are two metrics derived from the confusion matrix. FNR is the percentage of false negatives among all actual positive instances, which describes the proportion of positive instances missed by the classifier. Similarly, FPR reflects the proportion of negative instances misclassified by the classifier. **Adjusted Rand Index** (**ARI**) [117] is an accuracy-based metric without label information. [19] uses it to measure the similarity between two clusters in a graph.

Besides the above metrics, Area-under-the-ROC-curve (**AUC**) [118] and **Average Precision** (**AP**) [111] are widely used, such as by [52], [92], [109], [127], [144]. AUC is sensitive to the probability rank of positive instances, which is larger when positive instances are ranked higher than negative instances according to the predicted probability of a classifier. AP is a metric balancing the Precision and Recall where AP is higher when Precision is higher as Recall threshold increase from 0 to 1. Those two metrics could better reflect the classification performance as single scores since they provide an all-around evaluation over the predicted probabilities of all instances.

5.1.2 Ranking-based Metric

Mean Reciprocal Rank (**MRR**) [113] and **Hits@K** are two ranking metrics used by [83], [136] to evaluate the performance of link prediction on knowledge graphs. Given a list of items retrieved regarding a query and ranked by their probabilities, the reciprocal rank of a query response is the multiplicative inverse of the rank of the first correct item: 1 for first place, 1⁄2 for second place, 1⁄3 for third place and so on. Hits@K is the number of correct answers among the *top K* items in the ranking list. It can be used to evaluate the performance of recommender system as well [39]. **nDCG@K** [116] is another metric to evaluate the robustness of recommendation models [81].

5.1.3 Graph-based Metric

The graph-based metrics indicate the specific properties of a graph. **Normalized Mutual Information** (**NMI**) [115] and **Modularity** [114] are two metrics used by [18], [19], [129] to evaluate the performance of community detection (i.e., clustering) on graphs. NMI is originated from information theory that measures the mutual dependence between two variables. In a community detection scenario, NMI is used to measure the amount of shared information (i.e., similarity) between two communities. Modularity is designed to measure the strength of the division of a graph into clusters. Graphs with high Modularity have dense connections between the nodes within clusters but sparse connections between nodes in different clusters.

[93] employs a couple of graph property statistics as metrics to evaluate how much the attacker changed the graph (i.e., the imperceptibility of attacks). The metrics include **Gini Coefficient**, **Characteristic Path Length**, **Distribution Entropy**, **Power Law Exponent**, and **Triangle Count**. Please refer to [10] for more details about those metrics. Some more graph statistics metrics include **Degree Ranking**, **Closeness Ranking**, **Betweenness Ranking** used by [108] and **Clustering Coefficient**, **Shortest Path-length**, **Diagonal Distance** used by [128].

5.2 Adversarial Metric

Besides the general metrics above, a number of metrics which measure the attack and defense performance on graph data have been proposed or used by existing works. We first present the detailed formulations and descriptions of widely used metrics, and then briefly summarize some unique metrics used by particular papers. The reference after each metric name refers to the first paper that proposes or uses this metric and the references inside the parentheses refer to other attack and defense papers using this metric.

5.2.1 Common Metric

- **Attack Success Rate (ASR) [27]** ([12], [20], [21], [23], [24], [56], [73], [96], [106], [122], [134], [141]). ASR is the most frequently used metric to measure the performance of a giving attack approach:

$$\text{ASR} = \frac{\text{\# Successful attacks}}{\text{\# All attacks}}.$$

- **Classification Margin (CM) [145]** ([76], [105], [120], [140], [146]). CM measures the performance of the integrity attack:

$$\text{CM}(t) = p_{t,c_t} - \max_{c \neq c_t} p_{t,c},$$

where t is the target instance, c_t is the ground-truth class for t, $p_{t,c}$ is the probability of t being c. The above equation calculates the maximum difference between the probability of ground-truth class and that of other classes. In other words, it shows the extent of an attack flipping the predicted class of a target instance. [76] proposed another version of CM:

$$\text{CM}(t) = \log \frac{p_{t,c_t}}{\max_{c \neq c_t} p_{t,c}}.$$

When the instance is correctly classified, CM will be positive; otherwise it will be negative.

- **Correct/Mis Classification Rate [8]** ([36], [97], [107], [125], [147]). Those two metrics evaluate the attack/defense performance based on the classification results among all instances.

$$\text{MCR} = \frac{\text{\# Misclassified instances}}{\text{\# All instances}};$$
$$\text{CCR} = 1 - \text{MCR}.$$

- **Attacker Budget [76]** ([33], [77]). Attacker budget is a general metric to measure the minimum perturbations the attacker needs to fulfill its objective. The lower value indicates a better attack performance and a worse defense performance respectively. [33] takes number of removed edges as the attacker budget. [76], [77] take the smallest number of perturbations for the attacker to successfully cause the target to be misclassified as the budget.
- **Average Modified Links (AML) [23]** ([21], [23], [24], [134]). AML is a variance of Adversary budget introduced above. It describes the average number of modified links the attacker needed to meet the attack objective:

$$\text{AML} = \frac{\text{\# Modified links}}{\text{\# All attacks}}.$$

- **Concealment Measures [108]** ([67], [109], [128]). The concealment measures are used to evaluate the performance of hiding nodes or communities in a graph [67], [108], [109]. From another perspective, the structural changes introduced by an attack can be used to quantify the concealment of the attack as well [128].
- **Similarity Score [92]** ([143]). Similarity score is a general metric to measure the similarity of given instance pairs. It can be used as the goal of integrity attack where the attacker's goal is either to increase or decrease the similarity score of a target instance pair. For a node instance in a graph, both of its local structure and node embedding can be used to compute the similarity score.

5.2.2 Unique Metric

- **Averaged Worst-case Margin (AWM) [9].** The worst-case margin is the minimum value of the classification margin defined above. The averaged worse-case margin means the value is averaged across a worst-case margin of each batch of data.
- **Robustness Merit (RM) [57].** RM is the difference between the post-attack accuracy of the propose method and the post-attack accuracy of the vanilla GCN model. A greater value indicates a better defense performance.
- **Attack Deterioration (AD) [57].** AD is the ratio of decreased amount of accuracy after an attack to the accuracy without attack.
- **Average Defense Rate (ADR) [22].** ADR is a metric evaluating the defense performance according to the ASR defined above. It compares the ASR after attacks with or without applying the defense approach.
- **Average Confidence Different (ACD) [22].** ACD is a metric evaluating the defense performance based on the average difference between the classification margin after and before the attack of a set of nodes. Such a set of nodes includes correctly classified nodes before the attack.
- **Damage Prevention Ratio (DPR) [142].** Damage prevention measures the amount of damage that can be prevented by the defense. Let L_0 be the defender's accumulated loss when there is no attack. Let L_A be the defender's loss under some attack A when the defender cannot make any reliable queries. L_D denotes the loss when the defender make reliable queries according to a certain defense strategy D. DPR can be defined as follows:

$$\text{DPR}_A^D = \frac{L_A - L_D}{L_A - L_0}.$$

- **Certified Accuracy [55].** It is proposed to evaluate the certification method for robust community detection models against adversarial attacks. The certified accuracy $CK(l)$ is the fraction of sets of victim nodes that proposed method can provably detect as in the same community when an attacker adds or removes at most l edges in the graph.
- **Practical Effect [34].** Since the attacker may target at practical effect of attacks like boosting item revenue or reputation, [34] proposed a revenue-based metric to measure the performance of attacks and defenses from a practical angle.

6 DATASET AND APPLICATION

Table 4 summarizes some common datasets used in adversarial attack and defense works on graph data. The first four citation graphs have been widely used as node classification benchmarks in previous work [61], [100], [101], [119]. [92] also studies the adversarial link prediction problem on Cora and Citeseer. DBLP includes multiple citation datasets with more metadata information. Thus it can be used to study the community detection task [55]. Among the social network datasets, PolBlogs is another dataset used especially in adversarial settings where blogs are nodes and their

TABLE 4
Summary of datasets (ordered by the frequency of usage within each graph type).

Type	Task	Dataset	Source	# Nodes	# Edges	# Features	# Classes	Paper
Citation Network	Node/Link	Citeseer	[87]	3,327	4,732	3,703	6	[145], [28], [23], [106], [92], [8], [147], [12], [129], [120], [125], [17], [127], [40], [140], [89], [22], [144], [105] , [57], [148], [9], [52], [53], [107], [36], [134], [96], [20], [54], [29], [56], [31], [76], [77], [146], [59], [49], [104], [135], [138], [126]
	Node/Link	Cora	[87]	2,708	5,429	1,433	7	[28], [23], [106], [92], [8], [147], [12], [129], [120], [125], [17], [127], [40], [140], [89], [22], [144], [105] , [57], [52], [53], [107], [134], [20], [54], [29], [56], [31], [76], [77], [59], [49], [104], [135], [138], [126]
	Node	Pubmed	[87]	19,717	44,338	500	3	[28], [147], [17], [93], [89], [144], [57], [148], [9], [52], [53], [99], [54], [56], [31], [77], [146], [146], [59], [49], [104]
	Node	Cora-ML	[75]	2,995	8,416	2,879	7	[145], [148], [9], [93], [36], [96], [56], [146]
	Node/Community	DBLP	[98]	-	-	-	-	[67], [56], [55], [104]
Social Network	Node/Link	PolBlogs	[1]	1,490	19,025	-	2	[145], [23], [8], [120], [19], [140], [22], [105] , [52], [53], [36], [134], [20], [54], [56], [77], [59], [135]
	Node/Link	Facebook	[65]	-	-	-	-	[109], [21], [92], [143], [103]
	Node/Community	Google+	[65]	107,614	13,673,453	-	-	[108], [109], [103]
	Node	Reddit	[47]	1,490	19,090	300	2	[107], [99], [104]
	Community	Dolphin	[72]	62	159	-	-	[18], [129], [132]
	Community	WTC 9/11	[63]	36	64	-	-	[108], [109]
	Community	Email	[65]	1,005	25,571	-	-	[19], [55]
	Community	Karate	[133]	34	78	-	-	[18], [129]
	Community	Football	[44]	115	613	-	-	[18], [19]
	Fraud Detection	Yelp	[84]	-	-	-	-	[34], [137]
	Recommendation	MovieLens	[46]	-	-	-	-	[81], [39]
Knowledge Graph	Fact/Link	WN18	[11]	-	-	-	-	[136], [83]
	Fact	FB15k	[11]	-	-	-	-	[136]
Others	Node	Scale-free	[4]	-	-	-	-	[108], [109], [128]
	Node	NELL	[131]	65,755	266,144	5,414	210	[40], [31]
	Graph	Bitcoin	[110]	-	-	-	-	[122], [141]
	Graph/Node	AIDS	[85]	-	-	-	-	[97], [122], [49]
	Graph/Node	DHFR	[94]	-	-	-	-	[97], [49]

cross-references are edges. Reddit and Facebook are two larger graph datasets compared to citation datasets. Since there are multiple versions of Facebook datasets used across different papers, we omit its statistics. WTC 9/11, Email, Dolphin, Karate, and Football are five benchmark datasets for community detection. Some recent works also studied attacks and defenses of recommender system [39], [81] and review system [34], [137] based on the Yelp and MovieLens data. [83], [136] investigated the adversarial attacks and defenses on knowledge graphs using two knowledge graph benchmarks WN18 and FB15k. Scale-free network is a typical type of graph synthesized by graph generation models. Some works also employ other graph generation models like Erdős-Rényi model to generate graphs to facilitate their experiments [14], [26], [60], [108], [109], [141], [143]. Besides the node-level tasks, Bitcoin, AIDS, and DHFR datasets which contain multiple graphs are used to investiagte the robustness of graph classfication models [49], [97], [122], [141]. Among them, Bitcoin is a Bitcoin transaction dataset, AIDS contains biological graphs to represent the antiviral character of different biology compounds, and DHFR contains graphs to represent the chemical bond type.

Future Directions. Besides the datasets listed in Table 4, it is worth noting some other datasets which get less attention but could be studied in future researches. [50] is the first and only paper investigating the vulnerability of Heterogeneous Information Network (HIN) which is a graph model with heterogeneous node and edge types [88]. Though HIN has been applied to many security applications like malicious user detection [139], spam detection [66], and financial fraud detection [51], its robustness against adversarial attacks remain largely unexplored. A recent work [43] firstly gives a formulation of adversarial attacks on opinion propagation on graphs with a spectral form that could be used to study the opinion dynamics of social network. [83], [136] are the first two works studying the adversarial attacks and defenses on Knowledge Graph (KG) models. As the research of KG becomes popular in recent years, its security issue needs to be noticed as well. The security of dynamic graph models [24] is another avenue of research as well.

Besides the above works and datasets. there has been little discussion on the security issues of many other graph types and their related applications. To name a few, the biology graph, causal graph, and bipartite graph have attracted significant research attention but few work has studied potential attacks and their countermeasures on those graphs. From the perspective of applications, as the GNNs having been successfully applied to recommender system, computer vision and natural language processing [121], adversarial attacks and defenses on graph data under those specific applications is another promising research direction with de facto impacts.

7 Conclusion

In this work, we cover the most released papers about adversarial attack and defense on graph data as we know. We firstly provide an unified problem formulation for adversarial learning on graph data, and give definitions and taxonomies to category the papers. Next, we summary most existing imperceptible perturbations evaluation metrics, datasets and discuss several principles about imperceptibility metric. Then, we analyze the contributions and limitations of the existing works. Finally, we point out the potential research opportunities and directions in future studies.

Appendix

In this work, we not only develop taxonomies for all relevant works based on different criteria, but also summarize

TABLE 5
Summary of open-source implementations of algorithms.

Type	Paper	Algorithm	Link
Graph Attack	[23]	FGA	https://github.com/DSE-MSU/DeepRobust
	[108]	DICE	https://github.com/DSE-MSU/DeepRobust
	[145]	Nettack	https://github.com/danielzuegner/nettack
	[28]	RL-S2V, GraArgmax	https://github.com/Hanjun-Dai/graph_adversarial_attack
	[120]	IG-Attack	https://github.com/DSE-MSU/DeepRobust
	[147]	Meta-self, Greedy	https://github.com/danielzuegner/gnn-meta-attack
	[8]	ICML-19	https://github.com/abojchevski/node_embedding_attack
	[125]	PGD, Min-max	https://github.com/KaidiXu/GCN_ADV_Train
	[17]	GF-Attack	https://github.com/SwiftieH/GFAttack
	[93]	NIPA	https://github.com/DSE-MSU/DeepRobust
	[134]	GUA	https://github.com/chisam0217/Graph-Universal-Attack
	[34]	IncBP, IncDS	https://github.com/YingtongDou/Nash-Detect
Graph Defense	[40]	GraphAT	https://github.com/fulifeng/GraphAT
	[29]	AdvT4NE	https://github.com/wonniu/AdvT4NE_WWW2019
	[144]	RGCN	https://github.com/DSE-MSU/DeepRobust
	[120]	GCN-Jaccard	https://github.com/DSE-MSU/DeepRobust
	[125]	Adversarial Training	https://github.com/KaidiXu/GCN_ADV_Train
	[148]	Robust-GCN	https://github.com/danielzuegner/robust-gcn
	[99]	PA-GNN	https://github.com/tangxianfeng/PA-GNN
	[57]	r-GCN, VPN	https://www.dropbox.com/sh/p36pzx1ock2iamo/AABEr7FtM5nqwC4i9nICLIsta?dl=0
	[9]	Graph-cert	https://github.com/abojchevski/graph_cert
	[36]	GCN-SVD	https://github.com/DSE-MSU/DeepRobust
	[59]	Pro-GNN	https://github.com/DSE-MSU/DeepRobust
	[135]	DefenseVGAE	https://github.com/zhangao520/defense-vgae
	[60]	SPGF	https://github.com/henrykenlay/spgf
	[34]	Nash-Detect	https://github.com/YingtongDou/Nash-Detect
Other Baseline	[45]	FGSM	https://github.com/1Konny/FGSM
	[80]	JSMA	https://github.com/tensorflow/cleverhans
	[6]	Gradient Attack (GA)	https://github.com/bethgelab/foolbox/blob/master/foolbox/attacks/gradient.py
	[41]	First-order	https://github.com/cbfinn/maml

the corresponding datasets and metrics that are frequently used. Moreover, in Table 5, we also provide links to the open-source implementation of popular methods. We hope our work can facilitate the community towards the construction of benchmarks like in other areas [30], [102].

References

[1] Lada A Adamic and Natalie Glance. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43, 2005.

[2] Joshua Agterberg, Youngser Park, Jonathan Larson, Christopher White, Carey E Priebe, and Vince Lyzinski. Vertex nomination, consistent estimation, and adversarial modification. *arXiv preprint arXiv:1905.01776*, 2019.

[3] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

[4] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[5] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. Exploring the space of black-box attacks on deep neural networks. *arXiv:1712.09491v1*, 2017.

[6] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013.

[7] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[8] Aleksandar Bojchevski and Stephan Günnemann. Adversarial attacks on node embeddings via graph poisoning. In *International Conference on Machine Learning*, pages 695–704, 2019.

[9] Aleksandar Bojchevski and Stephan Günnemann. Certifiable robustness to graph perturbations. In *Advances in Neural Information Processing Systems*, pages 8317–8328, 2019.

[10] Aleksandar Bojchevski, Oleksandr Shchur, Daniel Zügner, and Stephan Günnemann. Netgan: Generating graphs via random walks. *arXiv preprint arXiv:1803.00816*, 2018.

[11] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.

[12] Avishek Joey Bose, Andre Cianflone, and William Hamiltion. Generalizable adversarial attacks using generative models. *arXiv preprint arXiv:1905.10864*, 2019.

[13] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

[14] Adam Breuer, Roee Eilat, and Udi Weinsberg. Friend or faux: Graph-based early detection of fake accounts on social networks. In *Proceedings of The Web Conference 2020*, pages 1287–1297, 2020.

[15] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14. ACM, 2017.

[16] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

[17] Heng Chang, Yu Rong, Tingyang Xu, Wenbing Huang, Honglei Zhang, Peng Cui, Wenwu Zhu, and Junzhou Huang. A restricted black-box adversarial framework towards attacking graph embedding models. AAAI, 2020.

[18] Jinyin Chen, Lihong Chen, Yixian Chen, Minghao Zhao, Shanqing Yu, Qi Xuan, and Xiaoniu Yang. Ga-based q-attack on community detection. *IEEE Transactions on Computational Social Systems*, 6(3):491–503, 2019.

[19] Jinyin Chen, Yixian Chen, Lihong Chen, Minghao Zhao, and Qi Xuan. Multiscale evolutionary perturbation attack on community detection. *arXiv preprint arXiv:1910.09741*, 2019.

[20] Jinyin Chen, Yixian Chen, Haibin Zheng, Shijing Shen, Shanqing Yu, Dan Zhang, and Qi Xuan. Mga: Momentum gradient attack on network. *arXiv preprint arXiv:2002.11320*, 2020.

[21] Jinyin Chen, Ziqiang Shi, Yangyang Wu, Xuanheng Xu, and Haibin Zheng. Link prediction adversarial attack. *arXiv preprint arXiv:1810.01110*, 2018.

[22] Jinyin Chen, Yangyang Wu, Xiang Lin, and Qi Xuan. Can adversarial network attack be defended? *arXiv preprint arXiv:1903.05994*, 2019.

[23] Jinyin Chen, Yangyang Wu, Xuanheng Xu, Yixian Chen, Haibin Zheng, and Qi Xuan. Fast gradient attack on network embedding. *arXiv preprint arXiv:1809.02797*, 2018.

[24] Jinyin Chen, Jian Zhang, Zhi Chen, Min Du, Feifei Li, and Qi Xuan. Time-aware gradient attack on dynamic network link prediction. *arXiv preprint arXiv:1911.10561*, 2019.

[25] Liang Chen, Jintang Li, Jiaying Peng, Tao Xie, Zengxu Cao, Kun Xu, Xiangnan He, and Zibin Zheng. A survey of adversarial learning on graphs. *arXiv preprint arXiv:2003.05730*, 2020.

[26] Mayee Chen and Miklos Z Racz. Network disruption: maximizing disagreement and polarization in social networks. *arXiv preprint arXiv:2003.08377*, 2020.

[27] Yizheng Chen, Yacin Nadji, Athanasios Kountouras, Fabian Monrose, Roberto Perdisci, Manos Antonakakis, and Nikolaos Vasiloglou. Practical attacks against graph-based clustering. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1125–1142, 2017.

[28] Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. Adversarial attack on graph structured data. *arXiv preprint arXiv:1806.02371*, 2018.

[29] Quanyu Dai, Xiao Shen, Liang Zhang, Qiang Li, and Dan Wang. Adversarial training methods for network embedding. In *The World Wide Web Conference*, pages 329–339, 2019.

[30] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[31] Zhijie Deng, Yinpeng Dong, and Jun Zhu. Batch virtual adversarial training for graph convolutional networks. *arXiv preprint arXiv:1902.09192*, 2019.

[32] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[33] Palash Dey and Sourav Medya. Manipulating node similarity measures in network. *arXiv preprint arXiv:1910.11529*, 2019.

[34] Yingtong Dou, Guixiang Ma, Philip S Yu, and Sihong Xie. Robust spammer detection by nash reinforcement learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

[35] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 2224–2232, 2015.

[36] Negin Entezari, Saba A Al-Sayouri, Amirali Darvishzadeh, and Evangelos E Papalexakis. All you need is low (rank) defending against adversarial attacks on graphs. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 169–177, 2020.

[37] Sarah M. Erfani, Sutharshan Rajasegarar, Shanika Karunasekera, and Christopher Leckie. High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. *Pattern Recognition*, 58:121 – 134, 2016.

[38] Dhivya Eswaran, Stephan Günnemann, Christos Faloutsos, Disha Makhija, and Mohit Kumar. Zoobp: Belief propagation for heterogeneous networks. *Proceedings of the VLDB Endowment*, 10(5):625–636, 2017.

[39] Minghong Fang, Guolei Yang, Neil Zhenqiang Gong, and Jia Liu. Poisoning attacks to graph-based recommender systems. In *Proceedings of the 34th Annual Computer Security Applications Conference*, pages 381–392, 2018.

[40] Fuli Feng, Xiangnan He, Jie Tang, and Tat-Seng Chua. Graph adversarial training: Dynamically regularizing based on graph structure. *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[41] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

[42] James Fox and Sivasankaran Rajamanickam. How robust are graph neural networks to structural noise? *arXiv preprint arXiv:1912.10206*, 2019.

[43] Jason Gaitonde, Jon Kleinberg, and Eva Tardos. Adversarial perturbations of opinion dynamics in networks. *arXiv preprint arXiv:2003.07010*, 2020.

[44] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[45] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv:1412.6572v3*, 2015.

[46] Grouplens. Movielens dataset. https://bit.ly/2YHzDnZ.

[47] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034, 2017.

[48] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[49] Xinlei He, Jinyuan Jia, Michael Backes, Neil Zhenqiang Gong, and Yang Zhang. Stealing links from graph neural networks. *arXiv preprint arXiv:2005.02131*, 2020.

[50] Shifu Hou, Yujie Fan, Yiming Zhang, Yanfang Ye, Jingwei Lei, Wenqiang Wan, Jiabin Wang, Qi Xiong, and Fudong Shao. αcyber: Enhancing robustness of android malware detection system against adversarial attacks on heterogeneous graph based model. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 609–618, 2019.

[51] Binbin Hu, Zhiqiang Zhang, Chuan Shi, Jun Zhou, Xiaolong Li, and Yuan Qi. Cash-out user detection based on attributed heterogeneous information network with a hierarchical attention mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 946–953, 2019.

[52] Vassilis N Ioannidis, Dimitris Berberidis, and Georgios B Giannakis. Graphsac: Detecting anomalies in large-scale graphs. *arXiv preprint arXiv:1910.09589*, 2019.

[53] Vassilis N Ioannidis and Georgios B Giannakis. Edge dithering for robust adaptive graph convolutional networks. *arXiv preprint arXiv:1910.09590*, 2019.

[54] Vassilis N Ioannidis, Antonio G Marques, and Georgios B Giannakis. Tensor graph convolutional networks for multi-relational and robust learning. *arXiv preprint arXiv:2003.07729*, 2020.

[55] Jinyuan Jia, Binghui Wang, Xiaoyu Cao, and Neil Zhenqiang Gong. Certified robustness of community detection against adversarial structural perturbation via randomized smoothing. *arXiv preprint arXiv:2002.03421*, 2020.

[56] Hongwei Jin and Xinhua Zhang. Latent adversarial training of graph convolution networks. In *ICML Workshop on Learning and Reasoning with Graph-Structured Representations*, 2019.

[57] Ming Jin, Heng Chang, Wenwu Zhu, and Somayeh Sojoudi. Power up! robust graph convolutional network against evasion attacks based on graph powering. *arXiv preprint arXiv:1905.10029*, 2019.

[58] Wei Jin, Yaxin Li, Han Xu, Yiqi Wang, and Jiliang Tang. Adversarial attacks and defenses on graphs: A review and empirical study. *arXiv preprint arXiv:2003.00653*, 2020.

[59] Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. *arXiv preprint arXiv:2005.10203*, 2020.

[60] Henry Kenlay, Dorina Thanou, and Xiaowen Dong. On the stability of polynomial spectral graph filters. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5350–5354. IEEE, 2020.

[61] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[62] Danai Koutra, Ankur Parikh, Aaditya Ramdas, and Jing Xiang. Algorithms for graph similarity and subgraph matching. In *Proc. Ecol. Inference Conf.*, 2011.

[63] Valdis E Krebs. Mapping networks of terrorist cells. *Connections*, 24(3):43–52, 2002.

[64] Chetan Kumar, Riazat Ryan, and Ming Shao. Adversary for social good: Protecting familial privacy through joint adversarial attacks. In *Conference on Artificial Intelligence (AAAI)*, 2020.

[65] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2–es, 2007.

[66] Ao Li, Zhou Qin, Runshi Liu, Yiqun Yang, and Dong Li. Spam review detection with graph convolutional networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2703–2711, 2019.

[67] Jia Li, Honglei Zhang, Zhichao Han, Yu Rong, Hong Cheng, and Junzhou Huang. Adversarial attack on community detection by hiding individuals. In *WWW*, 2020.

[68] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926v3*, 2018.

[69] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A.W.M. van der Laak, Bram van Ginneken, and Clara I. Sanchez. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60 – 88, 2017.

[70] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016.

[71] Alvis Logins, Yuchen Li, and Panagiotis Karras. On the robustness of cascade diffusion under node attacks. In *Proceedings of The Web Conference 2020*, pages 2711–2717, 2020.

[72] David Lusseau, Karsten Schneider, Oliver J Boisseau, Patti Haase, Elisabeth Slooten, and Steve M Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.

[73] Yao Ma, Suhang Wang, Lingfei Wu, and Jiliang Tang. Attacking graph convolutional networks via rewiring. *arXiv preprint arXiv:1906.03750*, 2019.

[74] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[75] Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.

[76] Benjamin A Miller, Mustafa Çamurcu, Alexander J Gomez, Kevin Chan, and Tina Eliassi-Rad. Improving robustness to attacks against vertex classification. In *MLG Workshop in KDD*, 2019.

[77] Benjamin A Miller, Mustafa Çamurcu, Alexander J Gomez, Kevin Chan, and Tina Eliassi-Rad. Topological effects on attacks against vertex classification. *arXiv preprint arXiv:2003.05822*, 2020.

[78] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 2017.

[79] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519. ACM, 2017.

[80] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

[81] Shaowen Peng and Tsunenori Mine. A robust hierarchical graph convolutional network model for collaborative filtering. *arXiv preprint arXiv:2004.14734*, 2020.

[82] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. *arXiv preprint arXiv:1403.6652v2*, 2014.

[83] Pouya Pezeshkpour, Yifan Tian, and Sameer Singh. Investigating robustness and interpretability of link prediction via adversarial modifications. *arXiv preprint arXiv:1905.00563*, 2019.

[84] Shebuti Rayana and Leman Akoglu. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*, pages 985–994, 2015.

[85] Kaspar Riesen and Horst Bunke. Iam graph database repository for graph based pattern recognition and machine learning. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 287–297. Springer, 2008.

[86] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018.

[87] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

[88] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and S Yu Philip. A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):17–37, 2016.

[89] Ke Sun, Zhouchen Lin, Hantao Guo, and Zhanxing Zhu. Virtual adversarial training on graph convolutional networks in node classification. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pages 431–443. Springer, 2019.

[90] Lichao Sun, Zhiqiang Li, Qiben Yan, Witawas Srisa-an, and Yu Pan. Sigpid: significant permission identification for android malware detection. In *Malicious and Unwanted Software (MALWARE), 2016 11th International Conference on*, pages 1–8. IEEE, 2016.

[91] Lichao Sun, Yuqi Wang, Bokai Cao, S Yu Philip, Witawas Srisa-An, and Alex D Leow. Sequential keystroke behavioral biometrics for mobile user identification via multi-view deep learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 228–240. Springer, 2017.

[92] M. Sun, J. Tang, H. Li, Bo Li, C. X., Y. Chen, and D. Song. Data poisoning attack against unsupervised node embedding methods. *arXiv preprint arXiv:1810.12881*, 2018.

[93] Yiwei Sun, Suhang Wang, Xianfeng Tang, Tsung-Yu Hsieh, and Vasant Honavar. Non-target-specific node injection attacks on graph neural networks: A hierarchical reinforcement learning approach. In *WWW*, 2020.

[94] Jeffrey J Sutherland, Lee A O'brien, and Donald F Weaver. Spline-fitting with a genetic algorithm: A method for developing classification structure- activity relationships. *Journal of chemical information and computer sciences*, 43(6):1906–1915, 2003.

[95] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv:1312.6199v4*, 2014.

[96] Tsubasa Takahashi. Indirect adversarial attacks via poisoning neighbors for graph convolutional networks. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1395–1400. IEEE, 2019.

[97] Haoteng Tang, Guixiang Ma, Yurong Chen, Lei Guo, Wei Wang, Bo Zeng, and Liang Zhan. Adversarial attack on hierarchical graph pooling neural networks. *arXiv preprint arXiv:2005.11560*, 2020.

[98] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008.

[99] Xianfeng Tang, Yandong Li, Yiwei Sun, Huaxiu Yao, Prasenjit Mitra, and Suhang Wang. Transferring robustness for graph neural network against poisoning attacks. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 600–608, 2020.

[100] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[101] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018.

[102] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[103] Binghui Wang and Neil Zhenqiang Gong. Attacking graph-based classification via manipulating the graph structure. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 2023–2040, 2019.

[104] Jihong Wang, Minnan Luo, Fnu Suya, Jundong Li, Zijiang Yang, and Qinghua Zheng. Scalable attack on graph data by injecting vicious nodes. *arXiv preprint arXiv:2004.13825*, 2020.

[105] Shen Wang, Zhengzhang Chen, Jingchao Ni, Xiao Yu, Zhichun Li, Haifeng Chen, and Philip S Yu. Adversarial defense framework for graph neural network. *arXiv preprint arXiv:1905.03679*, 2019.

[106] Xiaoyun Wang, Joe Eaton, Cho-Jui Hsieh, and Felix Wu. Attack graph convolutional networks by adding fake nodes. *arXiv preprint arXiv:1810.10751*, 2018.

[107] Xiaoyun Wang, Xuanqing Liu, and Cho-Jui Hsieh. Graphdefense: Towards robust graph convolutional networks. *arXiv preprint arXiv:1911.04429*, 2019.
[108] Marcin Waniek, Tomasz P Michalak, Michael J Wooldridge, and Talal Rahwan. Hiding individuals and communities in a social network. *Nature Human Behaviour*, 2(2):139–147, 2018.
[109] Marcin Waniek, Kai Zhou, Yevgeniy Vorobeychik, Esteban Moro, Tomasz P Michalak, and Talal Rahwan. Attack tolerance of link prediction algorithms: How to hide your relations in a social network. *arXiv preprint arXiv:1809.00152*, 2018.
[110] Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I Weidele, Claudio Bellei, Tom Robinson, and Charles E Leiserson. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv preprint arXiv:1908.02591*, 2019.
[111] Wikipedia. Average precision. https://bit.ly/2Uz06lL.
[112] Wikipedia. Confusion matrix. https://bit.ly/2wHUpcf.
[113] Wikipedia. Mean reciprocal rank. https://bit.ly/3aBadMk.
[114] Wikipedia. Modularity. https://bit.ly/3dMbsdB).
[115] Wikipedia. Mutual information. https://bit.ly/3bBeDCY.
[116] Wikipedia. ndcg. https://bit.ly/3dKYqf6.
[117] Wikipedia. Rand index. https://bit.ly/3azqoK6.
[118] Wikipedia. Roc. https://bit.ly/341yHfa.
[119] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr, Christopher Fifty, Tao Yu, and Kilian Q Weinberger. Simplifying graph convolutional networks. *arXiv preprint arXiv:1902.07153*, 2019.
[120] Huijun Wu, Chen Wang, Yuriy Tyshetskiy, Andrew Docherty, Kai Lu, and Liming Zhu. Adversarial examples for graph data: Deep insights into attack and defense. In *International Joint Conference on Artificial Intelligence, IJCAI*, pages 4816–4823, 2019.
[121] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019.
[122] Zhaohan Xi, Ren Pang, Shouling Ji, and Ting Wang. Graph backdoor. *arXiv preprint arXiv:2006.11890*, 2020.
[123] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. *arXiv preprint arXiv:1801.02612*, 2018.
[124] Hui Y. Xiong, Babak Alipanahi, Leo J. Lee, Hannes Bretschneider, Daniele Merico, Ryan K. C. Yuen, Yimin Hua, Serge Gueroussov, Hamed S. Najafabadi, Timothy R. Hughes, Quaid Morris, Yoseph Barash, Adrian R. Krainer, Nebojsa Jojic, Stephen W. Scherer, Benjamin J. Blencowe, and Brendan J. Frey. The human splicing code reveals new insights into the genetic determinants of disease. *Science*, 347(6218), 2015.
[125] Kaidi Xu, Hongge Chen, Sijia Liu, Pin-Yu Chen, Tsui-Wei Weng, Mingyi Hong, and Xue Lin. Topology attack and defense for graph neural networks: an optimization perspective. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3961–3967. AAAI Press, 2019.
[126] X Xu, Y Yu, L Song, C Liu, B Kailkhura, C Gunter, and B Li. Edog: Adversarial edge detection for graph neural networks. Technical report, Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 2020.
[127] Xiaojun Xu, Yue Yu, Bo Li, Le Song, Chengfeng Liu, and Carl Gunter. Characterizing malicious edges targeting on graph neural networks. *Openreview*, 2018.
[128] Qi Xuan, Yalu Shan, Jinhuan Wang, Zhongyuan Ruan, and Guanrong Chen. Adversarial attacks to scale-free networks: Testing the robustness of physical criteria. *arXiv preprint arXiv:2002.01249*, 2020.
[129] Qi Xuan, Jun Zheng, Lihong Chen, Shanqing Yu, Jinyin Chen, Dan Zhang, and Qingpeng Zhang Member. Unsupervised euclidean distance attack on network embedding. *arXiv preprint arXiv:1905.11015*, 2019.
[130] Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. Hypergcn: A new method for training graph convolutional networks on hypergraphs. In *Advances in Neural Information Processing Systems*, pages 1509–1520, 2019.
[131] Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. *arXiv preprint arXiv:1603.08861*, 2016.
[132] Shanqing Yu, Minghao Zhao, Chenbo Fu, Jun Zheng, Huimin Huang, Xincheng Shu, Qi Xuan, and G Chen. Target defense against link-prediction-based attacks via evolutionary perturbations. *IEEE Transactions on Knowledge and Data Engineering*, 2019.
[133] Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473, 1977.
[134] Xiao Zang, Yi Xie, Jie Chen, and Bo Yuan. Graph universal adversarial attacks: A few bad actors ruin graph learning models. *arXiv preprint arXiv:2002.04784*, 2020.
[135] Ao Zhang and Jinwen Ma. Defensevgae: Defending against adversarial attacks on graph data via a variational graph autoencoder. *arXiv preprint arXiv:2006.08900*, 2020.
[136] Hengtong Zhang, Tianhang Zheng, Jing Gao, Chenglin Miao, Lu Su, Yaliang Li, and Kui Ren. Data poisoning attack against knowledge graph embedding. *arXiv preprint arXiv:1904.12052*, 2019.
[137] Shijie Zhang, Hongzhi Yin, Tong Chen, Quoc Viet Nguyen Hung, Zi Huang, and Lizhen Cui. Gcn-based user representation learning for unifying robust recommendation and fraudster detection. *arXiv preprint arXiv:2005.10150*, 2020.
[138] Xiang Zhang and Marinka Zitnik. Gnnguard: Defending graph neural networks against adversarial attacks. *arXiv preprint arXiv:2006.08149*, 2020.
[139] Yiming Zhang, Yujie Fan, Yanfang Ye, Liang Zhao, and Chuan Shi. Key player identification in underground forums over attributed heterogeneous information network embedding framework. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 549–558, 2019.
[140] Yingxue Zhang, S Khan, and Mark Coates. Comparing and detecting adversarial attacks for graph deep learning. In *Proc. Representation Learning on Graphs and Manifolds Workshop, Int. Conf. Learning Representations, New Orleans, LA, USA*, 2019.
[141] Zaixi Zhang, Jinyuan Jia, Binghui Wang, and Neil Zhenqiang Gong. Backdoor attacks to graph neural networks. *arXiv preprint arXiv:2006.11165*, 2020.
[142] Kai Zhou, Tomasz P Michalak, and Yevgeniy Vorobeychik. Adversarial robustness of similarity-based link prediction. *arXiv preprint arXiv:1909.01432*, 2019.
[143] Kai Zhou, Tomasz P Michalak, Marcin Waniek, Talal Rahwan, and Yevgeniy Vorobeychik. Attacking similarity-based link prediction in social networks. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 305–313. International Foundation for Autonomous Agents and Multiagent Systems, 2019.
[144] Dingyuan Zhu, Ziwei Zhang, Peng Cui, and Wenwu Zhu. Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1399–1407, 2019.
[145] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2847–2856, 2018.
[146] Daniel Zügner, Oliver Borchert, Amir Akbarnejad, and Stephan Guennemann. Adversarial attacks on graph neural networks: Perturbations and their patterns. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2020.
[147] Daniel Zügner and Stephan Günnemann. Adversarial attacks on graph neural networks via meta learning. *arXiv preprint arXiv:1902.08412*, 2019.
[148] Daniel Zügner and Stephan Günnemann. Certifiable robustness and robust training for graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 246–256, 2019.