Adversarial Attack and Defense on Graph Data:Survey

Lichao Sun, Yingtong Dou, Carl Yang, Kai Zhang, 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 about 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 data due to its representation structure. Given the importance of graph analysis, an increasing number of studies over the past few years have attempted to analyze the robustness of machine learning models on graph data. Nevertheless, existing research considering adversarial behaviors on graph data often focuses 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, this review is intended to provide an overall landscape of more than 100 papers on adversarial attack and defense strategies for graph data, and establish a unified formulation encompassing most graph adversarial learning models. Moreover, we also compare different graph attacks and defenses along with their contributions and limitations, as well as summarize the evaluation metrics, datasets and future trends. We hope this survey can help fill the gap in the literature and facilitate further development of this promising new field[1].

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

1 Introduction

Recent years have witnessed significant success achieved by deep neural networks (DNNs) in a variety of domains, ranging from image recognition [55], natural language processing [37], graph data applications [54], [69], [117], [118], [138], to healthcare analysis [90], brain circuit modeling [78], and gene mutation functionality [143]. With the superior performance, deep learning has been applied in several safety and security critical tasks such as self driving [9], malware detection [106], identification [107] and anomaly detection [42]. However, the lack of interpretability and robustness of DNNs makes them vulnerable to adversarial attacks. Szegedy et al. [111] have 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. [51] 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 [8], [19], [142]. For instance, black-box adversarial attacks are later explored based on transferability [81], [93] and query feedback from DNN models [5], [16]. Some defense and detection methods have also been followed to mitigate such adversarial behaviors [86], [102], while various adaptive attacks continue to be proposed showing that detection/defense is hard in general [3], [18].

Although there are an increasing number of studies on adversarial attack and defense, current research mainly focuses on image, natural language, and speech domains. The investigative effort 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 [33]. Compared with non-graph data, the adversarial analysis of graph data presents several unique challenges: 1) Unlike image data with continuous pixel values, the graph structure are discrete valued. It is difficult to design an efficient algorithm that can generate adversarial examples in the discrete space. 2) Adversarial perturbations are designed to be imperceptible to humans in the image domain, so one can force a particular distance

- *Lichao Sun, Kai Zhang, and Lifang He are with the Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA 18015 USA. E-mail: {lis221, kaz321, lih319}@lehigh.edu.*
- *Yingtong Dou and Philip S. Yu are with the University of Illinois at Chicago, Chicago, IL 60607 USA. E-mail: {ydou5, psyu}@uic.edu.*
- *Carl Yang is with the Emory University Atlanta, Atlanta, GA 30322 USA. E-mail: j.carlyang@emory.edu.*
- *Ji Wang is with the College of Systems Engineering, National University of Defense Technology, Changsha, Hunan 410073, China. E-mail: wangji@nudt.edu.cn.*
- *Bo Li is with the University of Illinois Urbana-Champaign at Champaign, Champaign, IL 61820 USA. E-mail: lbo@illinois.edu.*

Manuscript received 12 August 2020; revised 13 February 2022; accepted 7 August 2022. Date of publication 6 September 2022; date of current version 21 June 2023.
This work was supported by NSF under Grants III-1763325, III-1909323, III-2106758, and SaTC-1930941.
(Corresponding author: Lichao Sun.)
Recommended for acceptance by L. Chen.
This article has supplementary downloadable material available at https://doi.org/10.1109/TKDE.2022.3201243, provided by the authors.
Digital Object Identifier no. 10.1109/TKDE.2022.3201243

1. We also have created an online resource to keep track of relevant research on the basis of this survey at https://github.com/safe-graph/graph-adversarial-learning-literature

TABLE 1
Attack and Defense Works are Categorized by GNN or Non-GNN Oriented

Category	Type	Paper
Attack Model	GNN	[10], [14], [21], [25], [27], [33], [85], [108], [109], [121], [125], [146], [176], [178] [24], [41], [47], [56], [75], [112], [113], [123], [139], [141], [155], [165], [177]
	Non-GNN	[2], [22], [23], [28], [32], [38], [48], [57], [127], [128], [149], [150], [158], [172] [31], [39], [44]
Defense Model	GNN	[26], [36], [45], [63], [64], [88], [105], [115], [124], [139], [148], [163], [174], [179] [11], [41], [47], [59], [60], [61], [67], [89], [95], [126], [147], [156], [160], [161]
	Non-GNN	[2], [17], [34], [39], [48], [57], [62], [68], [82], [97], [153], [158], [171]

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 investigation.

Given the importance of graph applications in the context of big data and the successful use of graph neural networks (GNNs), the robustness of GNNs has attracted significant interests from both academia and industry. In recent years, many efforts have been made to explore adversarial attacks and defenses for a set of GNN models. The purpose of this paper is to present a comprehensive taxonomy of existing adversarial learning literature on graph data, to develop a framework to unify most existing approaches, and to explore the future tendencies. All relevant attack and defense studies are listed in Tables 3 and 4, primarily on the basis of tasks, strategies, baselines, evaluation metrics and datasets. Despite more than 100 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 follows.

Comprehensive Understanding. To the best of our knowledge, this survey is the first attempt to present an in-depth and comprehensive understanding of the literature about adversarial attack and defense on graph data. It has stimulated (and been cited by) various following-up research in this line [21], [29], [64], [66], [72], [151], [174]. This paper not only provides a broad perspective and guidance for key adversarial attack and defense technologies in the context of GNNs, but also explains many observations related to non-gradient and non-model-based approaches and gives an insight into future directions.

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[2]. This repository contains links to all relevant papers and corresponding codes, which makes it easier for researchers to use and track the latest developments, and could serve as a benchmark library in this area. However, many of these papers are preprints and reports which give a preview of research results, we will keep tracking them and weaving any updates into the repository accordingly. We hope this resource can foster further research on this important topic, and keep shedding light on all facets of future research and development.

Unified Problem Definition. Though there have been various attack and defense strategies on graph data, there is no unified approach to characterize their relationships and properties; each model seems to be a result of a unique approach. It is necessary to establish a good basis for easy understanding of existing models and efficient development of future technologies. In this review, we pioneer to establish a unified formulation and definition to systematically analyze all adversarial attack models on graph data. 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. So far there are over a hundred papers that study adversarial analysis on graph data. Compared with image data and text data, the analyses of graph data are more complex due to variations in the graph structure and task. Listing all papers could help but is not intuitive for readers to quickly understand the similarities and discrepancies between different studies. To this end, we summarize 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.

Datasets and Metrics. Due to different goals and data used in previous attack and defense works, it is difficult to compare the results of different studies. Currently, no one could directly answer the question about "what attack or defense is the best benchmark in this domain?". The only way to alleviate this is to build a benchmark like other areas [35], [120]. Toward this end, we not only develop taxonomies for previous approaches based on different criteria, but also summarize the corresponding datasets and metrics that are frequently used. We hope this study could pave the way for the community to establish a benchmark for future research and practical selection of models in this area.

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

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. The most frequently used notaions in the paper are summarized in Table 2.

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

TABLE 2
A Lookup Table of Commonly-Used Notations

Notation	Description	Notation	Description
G	original graph	$\mathcal{L}$	loss function
$\widehat{G}$	adversarial graph	f_θ	deep learning model
v	node	$\mathcal{Q}$	distance function
e	edge	ϵ	cost budget
c	target component	Φ	perturbation function
y	ground truth label	$\mathcal{D}$	dataset

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 $v_{j,2}^{(i)}$. Both nodes and edges can have arbitrarily associated data such as node features, edge weights and edge directions. According to these properties, graph data can be classified into different types as follows.

2.2 Types of Graph Data

Dynamic and Static Graphs. From a *temporal perspective*, graph data can be grouped into static graphs and dynamic graphs. A graph is dynamic, denoted as $G^{(t)}$, if any of its nodes, edges, node features, or edges features change over time. On the contrary, a static graph, denoted as G, consists of a fixed set of nodes and edges without changing over time.

A typical example of static graph is the molecular structure of drugs [40]. Once a drug is developed, its molecular structure does not change over time. Social network [96] is a good example of dynamic graphs. As people often add or remove friendship links in their social networks, the graph of relationships and interactions changes over time. In most existing attack works, the researchers study the attacks on dynamic graphs.

Directed and Undirected Graphs. The graphs can be divided into directed and undirected graphs according to whether the direction between the initial node and end node is unidirectional or bidirectional. A directed graph, denoted as $G^{(Dr)}$, has direction information associated with each edge, where any directed edge $e_1^{(i)} = (v_1^{(i)}, v_2^{(i)}) \neq (v_2^{(i)}, v_1^{(i)}) = e_2^{(i)}$, while an undirected graph has edges made up of unordered pairs of nodes.

Facebook is a classic undirected graph that A is B's friend means B is A's friend too. In contrast to friendships, links of many real networks such as the World Wide Web (WWW), food webs, neural networks, protein interaction networks and many online social networks are directed or asymmetrically weighted. Twitter is a typical example of directed graph, where the directed edge represents the following information from one user to another.

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 [76] 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 [43] 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 total number of 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 $f^{(ind)} \in F^{(ind)} : \mathcal{G} \rightarrow \mathcal{Y}$ 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, $c_1, c_2, \ldots,$ and 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 $f^{(tra)} \in F^{(tra)} : \mathcal{G} \rightarrow \mathcal{Y}$ 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 [10], [11], [33], [121], [139], [146], [174], [176], [178], [179] 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 [108], [172].

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 [33], 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 [141], [165].

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 [10], [21], [25], [32], [33], [41], [57], [75], [108], [121], [139], [146], [176], [178] 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,*

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

When G'_j equals to $\widehat{G}^{c_j}$, Eq. (2) represents the poisoning attack, whereas when G'_j is the original G without modification, Eq. (2) 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:

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

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 [70]. 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 Section 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.

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 $\hat{G}$ 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 existing works, 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 [176], 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 [176], 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. (2), 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. (2), 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 various 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 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 [10], [33], [176].
- *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 [33], [85], [108]. Some works can also construct the adversarial samples with generative models [14], [23], [47]. 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 [141], [165].
- *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 [38], [127], [172]. 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 [127].

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 study 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 [21], [33], [121], [139], [146], [176], [178] and node embedding [10], [158]. 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. (2), we have a general formula for adversarial attack on node-relevant tasks.

Link-Relevant Task. Several other existing works study node embedding [10], [25], [108] or topological similarity [128], [172] 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. (2), 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 [33], [113], [141], [165]. By setting c_i as representation of graph target in Eq. (2), we have a general formula for adversarial attack on graph-relevant tasks.

3.8 Summary: Attack on Graph

In this subsection, we analyze 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 3, 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 3. 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. [33] used reinforcement learning approach to discover adversarial attack, which is the only approach that supports black-box attack compared to other works. [176] 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. [25], [108] mainly attacked the link predication task with a deep graph convolutional embedding model. [10] attacked multiple models by approximating the spectrum and using the gradient information. [121] attacked node classification through optimization approach and systematically discussed adversarial attacks on graph data. Previous works focused on edge or node modification, whereas [139] also modified the node features and proposed a hybrid attack on the graph convolutional neural networks (GCN) [69]. In addition to gradient check, [41], [146] attacked GCN by using the first-gradient optimization and low-rank approximation which makes an attack more efficient. [21] attacked general learning approaches by devising new loss and approximating the spectrum. [57] 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, [109] used reinforcement learning to attack GCN. However, it contains an obvious issue that it needs to break the graph structure by injecting new nodes. [75] 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 [141], [165]. 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. [32] is one of the first works to attack the graph data, and it also first proposed the attack approach in the unsupervised learning setting. [127] first attacked community detection though edge rewriting based on a heuristic approach. [128] attacked link prediction based on a heuristic approach which is based on the similarity measures. [172] used a greedy approach to attack link prediction based local and global similarity measure. In addition to traditional graph applications, [158] first attacked knowledge graph and destroyed the basic relational graph prediction model. [22] attacked community detection based on genetic algorithms. Unlike previous approaches, it chose to use rewiring instead of adding/removing edges while attacking the data. [47] used a generation approach to create a new isomorphism network to attack node classification. In addition to all previous works, [38] 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 [44], [95], [160], fraud detection [17], [39], opinion dynamic [31], [48], and graph classification [113], [141], [165] 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 [10], [25], [33] 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.

Most Recent Work. For the recently proposed attack methods, imperciptible pertubations such as added edges and modified node features are also the principle approaches like previous work. For the defense models, instead of commonly-used adversarial training techniques before, some researchers [30], [79], [80] first tried to propose new neiborhood aggregation schemas which guarantee the theoretical robustness under adversarial attack.

Future Directions. Adversarial attack on graph data is a new and hot area, and potential research opportunities are summarized below: 1) Most graphs are associated with attributes or more complex contents on nodes or edges in practice. However, few studies have well designed adversarial attack on attributed graphs, e.g., heterogeneous information networks and web graphs. 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 in their models. 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.

TABLE 3
Summary of Adversarial Attack Works on Graph Data (Time Ascending)

Task	Ref.	Year	Venue	Model	Strategy	Approach	Baseline	Metric	Dataset
Graph clustering	[32]	2017	CCS	SVD, Node2vec, Community detection algs	Noise injection, Small community attack	Add/Delete edges	-	ASR, FPR	NXDOMAIN, Reverse Engineered DGA Domains
Node classification	[176]	2018	KDD	GCN, CLN, DeepWalk	Incremental attack	Add/Delete edges, Modify node features	Random, FGSM	Accuracy, Classifcation margin	Cora-ML, Citeseer, PolBlogs
	[125]	2018	arXiv	GCN	Greedy, GAN	Add fake nodes with fake features	Random, Nettack	Accuracy, F1, ASR	Cora, Citeseer
	[121]	2019	CCS	LinBP, LBP, JW, DeepWalk, LINE, GCN, RW, Node2vec	Optimization	Add/Delete edges	Random, Nettack	FNR, FPR	Google+, Epinions, Twitter, Facebook, Enron
	[146]	2019	IJCAI	GCN	First-order optimization	Add/Delete edges	DICE, Greedy, Meta-self	Misclassification rate	Cora, Citeseer
	[21]	2019	AAAI	GCN, LINE, SGC, DeepWalk	Approximate spectrum, Devise new loss	Add/Delete edges	Random, Degree, RL-S2V,	Accuracy	Cora, Citeseer, Pubmed
	[178]	2019	ICLR	GCN, CLN DeepWalk	Meta learning	Add/Delete edges	DICE, Nettack, First-order attack	Accuracy, Misclassification rate	Cora, Pubmed, Citeseer, PolBlogs
	[85]	2019	arXiv	GCN	Reinforcement learning	Rewire edges	RL-S2V, Random	ASR	Reddit-Multi, IMDB-Multi
	[14]	2019	arXiv	GCN	Adversarial generation	Modify node features	Nettack	ASR	Cora, Citeseer
	[139]	2019	IJCAI	GCN	Check gradients	Add/Delete edges, Modify node features	Random, Nettack FGSM, JSMA	Accuracy, Classification margin	Cora, Citeseer, PolBlogs
	[112]	2020	BigData	GCN	Check gradients	Modify node features	Nettack	ASR	Cora-ML, Citeseer
	[41]	2020	WSDM	GCN, t-PINE	Low-rank approximation	Add/Delete edges	Nettack	Correct classification rate	Cora-ML, Citeseer, PolBlogs
	[177]	2020	TKDD	GCN, CLN, DeepWalk	Incremental attack	Add/Delete edges, Modify node features	Random, FGSM	Accuracy, Classifcation margin	Cora-ML, Citeseer, PolBlogs, Pubmed
	[109]	2020	WWW	GCN	Reinforcement learning	Inject new nodes	Random, FGA, Preferential attack	Accuracy, Graph statistics	Cora-ML, Pubmed, Citeseer
	[84]	2020	NIPS	GCN, JK-Net	Check gradients	Modify node features	Degree, Betweenness, PageRank, Random	Mis-classification rate	Cora, Citeseer Pubmed
	[49]	2021	NIPS	GCN family models, GDC, SGC	Check gradients	Add/delete edges	FGSM, PGD, Acc.	ASR	Cora ML, Citeseer Pubmed, arXiv Products, Paper 100M
	[116]	2021	CIKM	GCN, GAT APPNP	Optimization	Inject new nodes	Radnom, MostAttr PrefEdge, NIPA AFGSM, G-NIA	Misclassifcation rate	Reddit, Citeseer ogbn-products
	[175]	2021	KDD	GCN	Optimization	Inject new nodes/edges	GSM, AFGSM, SPEIT	Classification Accuracy	KDD-CUP, Reddit ogbn-arxiv
	[155]	2021	IJCAI	GCN, DeepWalk, Node2vec, GAT	Check gradients	Add/Delete edges	Random, FGA, Victim-class attack	ASR, AML	Cora, Citeseer, PolBlogs
Link prediction	[108]	2018	arXiv	GAE, DeepWalk, Node2vec, LINE	Project gradient descent	Add/Delete edges	Degree sum, Shortest path, Random, PageRank	AP, Similarity score	Cora, Citeseer, Facebook
	[172]	2019	AAMAS	Local&Global Similarity measures	Submodular	Hide edges	Random, Greedy	Similarity score	Random, Facebook
	[28]	2021	TKDE	Deep dynamic network embedding algs	Check gradients	Rewire edges	Random, Gradient, Common neighbor	ASR, AML	LKML, FB-WOSN, RADOSLAW
	[6]	2021	EMNLP	TransE, DistMult ConvE, ComplEx	Instance attribution	Add/Delete facts	Direct-Add/Del, CRIAGE, Random edits, Gradient Rollback	MRR Hits@K	WN18RR FB15k-237
	[7]	2021	ACL	TransE, DistMult ConvE, ComplEx	Exploit relation inference patterns	Create decoy facts	Random, CRIAGE Edits in the neiborhood	MRR Hits@K	WN18RR FB15k-237
Graph classification	[119]	2021	NIPS	GCN, GIN Cheby-GIN Graph U-net	Beyasian optimization	Add/delete edges Rewire edges Inject new nodes	Random, Genetic Gradient-based	ASR	IMDB-M, Proteins Collab, Twitter fake news Reddict-Multi-5k
	[92]	2021	CCS	GIN, SAG GUNet	Optimization	Add/delete edges	Random, RL-S2V	ASR, Average Purturbation, Average Queries, Average Time	COIL, IMDB, NCI1
	[157]	2021	CIKM	GCN	Project ranking of elements	Add edges	RandomSampling GradArgmax RL-S2V	Correct classification rate	BA-2Motifs, ENZYMES, Mutagenicity, PC-3, NCI109, NCI-H23H
Community detection	[22]	2019	TCSS	Community detection algs	Genetic algs	Rewire edges	Random, Degree, Community detection	NMI, Modularity	Karate, Dolphin, Football, Polbooks
	[75]	2020	WWW	Surrogate community detection model	Graph auto-encoder	Add/Delete edges	DICE, Random, Modularity based attack	Personalized metric	DBLP, Finance
Node classification, Link prediction	[10]	2019	ICML	Node2vec, GCN LP, DeepWalk	Check gradient, Approximate spectrum	Add/Delete edges	Random, Degree, Eigenvalue	F1 score, Misclassification rate	Cora, Citeseer, PolBlogs
	[53]	2021	PAKDD	DeepWalk, Node2Vec, LINE, GCN	Optimization	add/delete edges	Random, UNSUP	Micro F1, Precision	LFR, Cora, Citeseer ForestFire, PolBlogs
Graph classification, Node classification	[33]	2018	ICML	GNN family models	Reinforcement learning	Add/Delete edges	Rnd. sampling, Genetic algs.	Accuracy	Citeseer,Finance, Pubmed, Cora
Malware detection, Node classification	[57]	2019	CIKM	Metapath2vec	Greedy	Inject new nodes	Anonymous attack	%TPR, TP-FP curve	Private dataset
Knowledge graph fact plausibility prediction	[158]	2019	IJCAI	RESCAL, TransE, TransR	Check target entity embeddings	Add/Delete fact	Random	MRR, Hit Rate@K	FB15k, WN18
Vertex nomination	[2]	2019	arXiv	VN-GMM-ASE	Random	Add/Delete edges	-	Achieving rank	Bing entity transition graph
Manipulating opinion	[48]	2020	arXiv	Graph model	Adversarial optimization	Change initial opinion vector	-	-	-
Fraud detection	[39]	2020	KDD	Graph-based Fraud detectors	Reinforcement learning	Add/Delete edges	-	Practical effect	YelpChi, YelpNYC, YelpZip
Graph matching	[167]	2020	NIPS	SNNA, DGMC CrossMNA	Kernel density estimation, Meta learning	Inject new nodes	Random, RL-S2V Meta-Self, CW-PGD GF-Attack, CD-ATTACK	Accuracy, Precision@K	Autonomous systems Last.FM, DBLP LiveJournal
Knowledge graph alignment	[166]	2021	EMNLP	GCN	Kernel density estimation	Add/Delete relations	SWS, IWS, DPA GF-Attack, LowBlow CRIAGE, RL-RR	MRR His@K	DBP15K
Question answering, Item recommendation	[98]	2021	ICLR	RN, MHGRN KGCN, RippleNet	Reinforcement learning, Heuristic	Replace relations	Random	Accuracy, AUC, Aggregated triple score, Similarity in clustering coefficient /degree distribution	CSQA, OBQA Last.FM MovieLens-20M
Malware detection	[168]	2021	CCS	FCG	Heuristic optimization, Reinforcement learning	Add/Rewire edges Insert/Delete nodes	-	Initialization/Relative/ /Absolute ASR	Malscan
Node Similarity	[38]	2020	AAMAS	Similarity measures	Graph theory	Remove edges	Greedy, Random, High jaccard similarity	# Removed edges	Power,web-edu, hamsterster, euroroad

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 [51]. In the graph setting, we formulate the objective of adversarial defense by slightly modifying our unified formulation of adversarial attacks, *i.e.*, Eq. (2), 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 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 [33]. 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, [146] 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 [125], [146], [172] without sacrificing node classification accuracy on the original graph. In the meantime, [34] 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 [96] on node classification. Towards similarity-based link prediction, [171] formalized a Bayesian Stackelberg game to optimize the most robust links to preserve with an adversary deleting the remaining links.

Attribute Perturbations. Besides links, [36], [45], [105] also perturb node features to enable virtual adversarial training [91] that enforces the smoothness between original nodes and adversarial nodes. In particular, [45] designed a dynamic regularizer forcing GNN models to learn to prevent the propagation of perturbations on graphs, whereas [105] smoothed GCN in its most sensitive directions to improve generalization. [36] further conducted virtual adversarial training in batch to perceive the connectivity patterns between nodes in each sampled subsets. [124] leveraged adversarial contrastive learning [15] 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, [126] proposed to directly perturb the adjacency matrix and feature matrix by ignoring the discreteness, whereas [63] 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 [27] and Nettack [176], [26] 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. [39] 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. [148] proposed different approaches to detect potential malicious edges based on graph generation models, link prediction and outlier detection. Instead of edges, [59] proposed to filter out node sets contaminated by anomalous nodes based on graph-aware criteria computed on randomly drawn subsets of nodes; [163] proposed to detect nodes subject to topological perturbations (particularly by Nettack [176]) 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, [139] proposed to compute the Jaccard Similarity to remove suspicious edges between suspicious nodes. [147] 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. [174] assumed high prediction uncertainty for adversarial nodes and computed the attention weights based on the embedding variance in a Gaussian-based GCN. [115] suggested to train an attack-aware GCN based on ground-truth poisoned links generated by Nettack [176] and transfer the ability to assign small attention weights to poisoned links based on meta-learning.

Robustness Certification. On the contrary of detecting attacks, [11], [179] designed robustness certificates to measure the safety of individual nodes under adversarial

perturbation. In particular, [11] considered structural perturbation while [179] considered attribute perturbation. Training GNN models jointly with these certificates can lead to a rigorous safety guarantee of more nodes. From a different perspective, [62] derived the robustness certificate of community detection methods under structural perturbation. [68] proved polynomial spectral graph filters are stable under structural perturbation.

Complex Graphs. Beyond traditional homogeneous graphs, [97] studied the sensitivity of knowledge graph link prediction models towards adversarial facts (links) and the identification of facts. [57] 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 [59], [148], [163]. There exist methods that modify graph data without directly detecting attacks. Based on the insight that Nettack [176] only affects the high-rank singular components of the graph, [41] proposed to reduce the effect of attacks by computing the low-rank approximation of the graphs before training GNN models. [47] 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. [89] 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. [156] leveraged variational graph autoencoders to reconstruct graph structures from perturbed graphs where 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, [64] 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. [95] 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. [161] 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. [60] 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. [88] proposed a heuristic method to iteratively select training data based on the degrees and connection patterns of nodes. It 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, [67] 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 [33], [36], [45], [105], [146]. 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 [146] trained in this way even attain good defense performance against other specific attacks like Meta-self proposed in [178]. 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. [115] assumes that there are unpolluted graphs to aid the detection of attacks. [57], [61], [64], [174] propose new GNN architectures to enhance their robustness. [88], [89] 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, [41] first argued the weakness of Nettack [176] and leveraged SVD to defend against Nettack. [63] analyzed the strategies and approaches of Nettack [176] and RL-S2V [33] and proposed an adversarial training method. [139] inspected two gradient-based attacks (i.e., FGSM [51] and JSMA [94]) and applied 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 [139] could defend the Nettack as well. Along with the **Corresp. Attack** column of Table 4, 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 [48] or optimization [11], [62], [179] 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.

Most Recent Work. In addition to common tasks such as node classification and link prediction, attacks [167] and defenses [100], [169], [173] on graph alignment tasks (e.g., graph matching) were also proposed.

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 Metrics

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 Tables 3 and 4, 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 [131] for detailed explanations of those metrics. Note that the *False Negative Rate* (**FNR**) and *False Positive Rate* (**FPR**) used by [32], [121] 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**) [136] is an accuracy-based metric without label information. [23] uses it to measure the similarity between two clusters in a graph.

Besides the above metrics, Area-under-the-ROC-curve (**AUC**) [137] and *Average Precision* (**AP**) [130] are widely used, such as by [59], [108], [128], [148], [174]. 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**) [132] and **Hits@K** are two ranking metrics used by [97], [158] 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 [44]. **nDCG@K** [135] is another metric to evaluate the robustness of recommendation models [95].

5.1.3 Graph-Based Metric

The graph-based metrics indicate the specific properties of a graph. *Normalized Mutual Information* (**NMI**) [134] and **Modularity** [133] are two metrics used by [22], [23], [150] 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.

[109] 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 [12]

for more details about those metrics. Some more graph statistics metrics include *Degree Ranking*, *Closeness Ranking*, *Betweenness Ranking* used by [127] and *Clustering Coefficient*, *Shortest Path-length*, *Diagonal Distance* used by [149].

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)* [32] ([14], [24], [25], [27], [28], [63], [85], [112], [125], [141], [155], [165]). 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)* [176] ([88], [124], [139], [163], [177]). 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. [88] 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* [10] ([41], [113], [126], [146], [178]). 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* [88] ([38], [89]). 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. [38] takes the number of removed edges as the attacker budget. [88], [89] take the smallest number of perturbations for the attacker to successfully cause the target to be misclassified as the budget.
- *Average Modified Links (AML)* [27] ([25], [27], [28], [155]). 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* [127] ([75], [128], [149]). The concealment measures are used to evaluate the performance of hiding nodes or communities in a graph [75], [127], [128]. From another perspective, the structural changes introduced by an attack can be used to quantify the concealment of the attack as well [149].
- *Similarity Score* [108] ([172]). 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)* [11]. 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) [64].*RM is the difference between the post-attack accuracy of the proposed method and the post-attack accuracy of the vanilla GCN model. A greater value indicates a better defense performance.
- *Attack Deterioration (AD) [64].*AD is the ratio of decreased amount of accuracy after an attack to the accuracy without attack.
- *Average Defense Rate (ADR) [26].*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) [26].*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) [171].*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:

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

TABLE 4
Summary of Adversarial Defense Works on Graph Data (Time Ascending)

Task	Ref.	Year	Venue	Model	Corresp. Attack	Strategy	Baseline	Metric	Dataset
Node classification	[45]	2019	TKDE	GCN	-	Adversarial training	DeepWalk, GCN, Planetoid, LP, GraphVAT, GraphSCAN	Accuracy	Cora, NELL, Citeseer
	[34]	2019	WWW	DeepWalk	-	Adversarial training	DeepWalk, LINE Node2vec, GraRep, Graph Factorization	Accuracy, AUC	Cora, Wiki Citeseer, CA-GrQc, CA-HepTh
	[126]	2019	arXiv	GCN, GraphSAGE	-	Adversarial training	Drop edges, Discrete adversarial training	Accuracy, Correct classification rate	Cora, Citeseer, Reddit
	[63]	2019	ICML Workshop	GCN	Nettack	Adversarial training	GCN, SGCN, FastGCN, SGC	ASR, Accuracy	Citeseer, Cora, Pubmed, Cora-ML, DBLP, PolBlogs
	[36]	2019	ICML Workshop	GCN	-	Adversarial training	GCN, GAT, LP, DeepWalk, Planetoid, Monet, GPNN	Accuracy	Citeseer, Cora, Pubmed, NELL
	[105]	2019	PRCV	GCN	-	Virtual adversarial training	GCN	Accuracy	Cora, Citeseer, Pubmed
	[11]	2019	NIPS	GCN	-	Robust training, MDP to get bound	GNN	Accuracy, Worst-case margin	Cora-ML, Pubmed, Citeseer
	[146]	2019	IJCAI	GCN	DICE, Meta-self	Check gradients, Adversarial training	GCN	Accuracy, Misclassification rate	Cora, Citeseer
	[139]	2019	IJCAI	GCN	Random, Nettack FGSM, JSMA	Drop edges	GCN	Accuracy, Classification margin	Cora, Citeseer, PolBlogs
	[179]	2019	KDD	GCN, GNN	-	Convex optimization	GNN	Accuracy, Average worst-case margin	Cora-ML, Pubmed, Citeseer
	[88]	2019	KDD Workshop	GCN, Node2vec	-	Change training set	GCN, Node2vec	Adversary budget, Classification margin	Cora, Citeseer
	[174]	2019	KDD	GCN	Nettack, RL-S2V, Random	Gaussian distribution layer, Variance-based attention	GCN, GAT	Accuracy	Cora, Citeseer, Pubmed
	[115]	2020	WSDM	GNN	Metattack	Meta learning, Transfer from clean graph	GCN, GAT, RGCN, VPN	Accuracy	Pubmed, Yelp, Reddit
	[41]	2020	WSDM	GCN, t-PINE	Nettack, LowBlow	Low-rank approximation	-	Correct classification rate	Cora-ML, Citeseer, PolBlogs
	[67]	2020	KDD	GNN	Nettack, Meta-self, Random	Graph structure learning	GCN, GCN-SVD, RGCN, GAT, GCN-Jaccard	Accuracy	Cora, Pubmed, Polblogs, Citeseer
	[20]	2021	NIPS	GCN	Nettack-One, Nettack-Multi, Metattack	Transfer robustness of low-frequency components by co-training	GCN-Jaccard/-SVD GNN GUARD, Pro-GNN	Accuracy	Cora, Citeseer, Pubmed, Coauthor CS, Amazon Photo
	[79]	2021	NIPS	GNN	Nettack	Adaptive message passing against abnormal node features	GCN, GAT APPNP, GCNII	Accuracy	Cora, Citeseer, Pubmed, Coauthor CS/Physics, Amazon Computers/Photo
	[30]	2021	IJCAI	GCN	Nettack	Aggregation with a high breakdown point	GCN, RGCN, GCN-Jaccard, SimPGCN	Accuracy	Cora, Cora-ML, Citeseer, Pubmed
	[65]	2021	WSDM	GNN	Metattack	Information aggregation based on feature similarity	GCN, GAT, Pro-GNN, GCN-Jaccard	Accuracy	Cora, Citeseer, Pubmed
	[80]	2021	ICML	GNN	Metattack	Message passing with graph smoothing	GCN, GAT, ChebNet, GraphSAGE, APPNP, SGC	Accuracy	Cora, Citeseer, Pubmed, Coauthor CS/Physics, Amazon Computers/Photo
	[164]	2021	AISTATS	GCN	DICE, Nettack, GF-Attack	Detect malicious nodes	GCN,SGCN,GAT, RGCN, GCN-Jaccard/-SVD	Accuracy, AUC	Cora, Citeseer, Polblogs, Pubmed
	[144]	2021	CIKM	GCN	Metattack, Nettack, Random	Graph structure learning with low-rank prior knowledge	RGCN, Pro-GNN GCN-Jaccard/-SVD	Accuracy	Cora, Citeseer Polblogs, Pubmed
Link prediction	[97]	2019	NAACL	Knowledge graph embeddings	-	Adversarial modification	-	Hits@K, MRR	Nations, WN18, Kinship, YAGO3-10
	[153]	2019	TKDE	Link prediction methods	Resource Allocation Index	Estimation of Distribution Algorithm	RLR, RLS HP, GA	Precision, AUC	Mexican, Dolphin Bomb, Lesmis, Throne, Jazz
	[171]	2019	ICDM	Similarity measures	-	Bayesian Stackelberg game and optimization	Protect Potential Neighbors	Damage prevention ratio	PA, TV Show, PLD, Gov
Graph classification	[165]	2020	arXiv	GIN	Graph generation	Randomized subsampling	-	ASR, Clean accuracy, Backdoor accuracy	Twitter, Bitcoin, COLLAB
Node embedding	[26]	2019	arXiv	GNN	Nettack, FGA	Smoothing gradients	Adversarial training	ADR, ACD	Cora, Citeseer, PolBlogs
Malware detection, Node classification	[57]	2019	CIKM	Heterogeneous graph, Metapath2vec	-	Attention mechanism	Other malware detection algs	Accuracy, F1, Precision, Recall	Private dataset
Community detection	[62]	2020	WWW	Community detection algs	-	Robust certification with optimization	-	Certified accuracy	Email, DBLP, Amazon
Fraud detection	[17]	2020	WWW	Graph-based Sybil detectors	Change label, Graph generation	Probability estimation	VoteTrust, SybilRank, SybilSCAR, SybilBelief	AUC	Facebook, Synthetic graphs
	[39]	2020	KDD	Graph-based Fraud detectors	IncBP, IncDS, IncPR, Random, Singleton	Minimax game, Reinforcement learning	SpEagle, GANG Fraudar, fBox	Practical effect	YelpChi, YelpNYC, YelpZip
Manipulating opinion	[48]	2020	arXiv	Graph model	-	Minimax game, Convex optimization	-	-	-
Recommendation system	[160]	2020	SIGIR	GCN	Mixed, Hate, Average, Random	Fraud detection	RCF, GCMC, GraphRec, MF, AutoRec, PMF	RMSE, MAE	Yelp, Moive&TV
	[159]	2021	WWW	GCN	Attribute Inference attack	Differential privacy	BPR, GCN, Blurm, DPAE, DPNE, DPMF, RAP	F1, NDCG@K, Hits@K	ML-100K
Node classification, Link prediction, Community detection	[145]	2022	AAAI	DeepWalk, GAE, DGI	-	Graph representation learning with optimization	Dwns_AdvT, RSC, DGI-EdgeDrop, -SVD, -Jaccard	Accuracy, AUC, NMI	Cora, Citeseer, Polblogs
Node classification, Graph classification	[122]	2021	KDD	GNN	Random	randomized smoothing	GCN, GAT	Certified accuracy	Cora, Citeseer, Pubmed, MUTAG, Proteins, IMDB
Graph classification Graph matching	[169]	2021	ICML	Graph classification /matching models	Radnom, NEA, GMA, RL-S2V	Constrain the norm of gradient	PAN, Pro-GNN, GRAND, GCN-SVD, RoboGraph, GraphCL, GroupSort, BCOP, FINAL REGAL, MOANA, DGMC, CONE-Align, G-CREWE	Accuracy	AS, CAIDA, DBLP, BZR, BZR_MD, MUTAG
Graph matching	[100]	2021	ICML	Graph matching algo	Random, NEA, GMA	Maximize distances between matched nodes; separate intra-graph nodes	FINAL, REGAL, MOANA, DGMC, CONE-Align, G-CREWE	Hits@K	AS, CAIDA, DBLP
Network alignment	[173]	2021	WWW	Network alignment algo	Random, Meta-self, GF, CD, GMA, LowBlow	Neutralize adversarial nodes to adversarial-free	GCN-Jaccard, GCN-SVD, Pro-GNN	Precision	AS, SNS, DBLP
Recommendation system, Knowledge graph, Quantum chemistry	[77]	2021	ICML	GNN	Neighborhood attack	Adversarial training	ChebNet, GraphSAGE	F1, AUC, RMSE	Movielens-1M, FB15k-237, WN18RR, Citeseer, Pubmed, QM9

TABLE 5
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	[103]	3,327	4,732	3,703	6	[176], [33], [27], [125], [108], [10], [178], [14], [150], [139], [146], [21], [148], [45], [163], [105], [26], [174], [124] , [64], [179], [11], [59], [60], [126], [41], [155], [112], [24], [61], [34], [63], [36], [88], [89], [177], [67], [56], [123], [156], [161], [147]
	Node/Link	Cora	[103]	2,708	5,429	1,433	7	[33], [27], [125], [108], [10], [178], [14], [150], [139], [146], [21], [148], [45], [163], [105], [26], [174], [124] , [64], [59], [60], [126], [155], [24], [61], [34], [63], [36], [88], [89], [67], [56], [123], [156], [161], [147]
	Node	Pubmed	[103]	19,717	44,338	500	3	[33], [178], [21], [109], [105], [174], [64], [179], [11], [59], [60], [115], [61], [63], [36], [89], [177], [177], [67], [56], [123]
	Node	Cora-ML	[87]	2,995	8,416	2,879	7	[176], [179], [11], [109], [41], [112], [63], [177]
	Node/Community	DBLP	[114]	-	-	-	-	[75], [63], [62], [123]
Social Network	Node/Link	PolBlogs	[1]	1,490	19,025	-	2	[176], [27], [10], [139], [23], [163], [26], [124] , [59], [60], [41], [155], [24], [61], [63], [89], [67], [156]
	Node/Link	Facebook	[73]	-	-	-	-	[128], [25], [108], [172], [121]
	Node/Community	Google+	[73]	107,614	13,673,453	-	-	[127], [128], [121]
	Node	Reddit	[54]	1,490	19,090	300	2	[126], [115], [123]
	Community	Dolphin	[83]	62	159	-	-	[22], [150], [153]
	Community	WTC 9/11	[71]	36	64	-	-	[127], [128]
	Community	Email	[73]	1,005	25,571	-	-	[23], [62]
	Community	Karate	[154]	34	78	-	-	[22], [150]
	Community	Football	[50]	115	613	-	-	[22], [23]
	Fraud Detection	Yelp	[99]	-	-	-	-	[39], [160]
	Recommendation	MovieLens	[52]	-	-	-	-	[95], [44]
Knowledge Graph	Fact/Link	WN18	[13]	-	-	-	-	[158], [97]
	Fact	FB15k	[13]	-	-	-	-	[158]
Others	Node	Scale-free	[4]	-	-	-	-	[127], [128], [149]
	Node	NELL	[152]	65,755	266,144	5,414	210	[45], [36]
	Graph	Bitcoin	[129]	-	-	-	-	[141], [165]
	Graph/Node	AIDS	[101]	-	-	-	-	[113], [141], [56]
	Graph/Node	DHFR	[110]	-	-	-	-	[113], [56]

- *Certified Accuracy [62].* 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 [39].* Since the attacker may target at practical effect of attacks like boosting item revenue or reputation, [39] proposed a revenue-based metric to measure the performance of attacks and defenses from a practical angle.

6 Dataset and Application

Table 5 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 [69], [117], [118], [138]. [108] 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 [62]. Among the social network datasets, PolBlogs is another dataset used especially in adversarial settings where blogs are nodes and their 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 [44], [95] and review system [39], [160] based on the Yelp and MovieLens data. [97], [158] 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 [17], [31], [68], [127], [128], [165], [172]. Besides the node-level tasks, Bitcoin, AIDS, and DHFR datasets which contain multiple graphs are used to investiagte the robustness of graph classfication models [56], [113], [141], [165]. 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 5, it is worth noting some other datasets which get less attention but could be studied in future research. To the best of our knowledge, [57] is the first and only paper to examine the vulnerability of Heterogeneous Information Network (HIN) which is a graph model with heterogeneous node and edge types [104]. Though HIN has been applied to many security applications like malicious user detection [162], spam detection [74], and financial fraud detection [58], its robustness against adversarial attacks remain largely unexplored. A recent study [48] 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. [97], [158] 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 [28] 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 [140], adversarial attacks

TABLE 6
Summary of Open-Source Implementations of Algorithms

Type	Paper	Algorithm	Link
Graph Attack	[27]	FGA	https://github.com/DSE-MSU/DeepRobust
	[127]	DICE	https://github.com/DSE-MSU/DeepRobust
	[176]	Nettack	https://github.com/danielzuegner/nettack
	[33]	RL-S2V, GraArgmax	https://github.com/Hanjun-Dai/graph_adversarial_attack
	[139]	IG-Attack	https://github.com/DSE-MSU/DeepRobust
	[178]	Meta-self, Greedy	https://github.com/danielzuegner/gnn-meta-attack
	[10]	ICML-19	https://github.com/abojchevski/node_embedding_attack
	[146]	PGD, Min-max	https://github.com/KaidiXu/GCN_ADV_Train
	[21]	GF-Attack	https://github.com/SwiftieH/GFAttack
	[109]	NIPA	https://github.com/DSE-MSU/DeepRobust
	[155]	GUA	https://github.com/chisam0217/Graph-Universal-Attack
	[39]	IncBP, IncDS	https://github.com/YingtongDou/Nash-Detect
Graph Defense	[45]	GraphAT	https://github.com/fulifeng/GraphAT
	[34]	AdvT4NE	https://github.com/wonniu/AdvT4NE_WWW2019
	[174]	RGCN	https://github.com/DSE-MSU/DeepRobust
	[139]	GCN-Jaccard	https://github.com/DSE-MSU/DeepRobust
	[146]	Adversarial Training	https://github.com/KaidiXu/GCN_ADV_Train
	[179]	Robust-GCN	https://github.com/danielzuegner/robust-gcn
	[115]	PA-GNN	https://github.com/tangxianfeng/PA-GNN
	[64]	r-GCN, VPN	https://www.dropbox.com/sh/p36pzx1ock2iamo/AABEr7FtM5nqwC4i9nICLIsta?dl=0
	[11]	Graph-cert	https://github.com/abojchevski/graph_cert
	[41]	GCN-SVD	https://github.com/DSE-MSU/DeepRobust
	[67]	Pro-GNN	https://github.com/DSE-MSU/DeepRobust
	[156]	DefenseVGAE	https://github.com/zhangao520/defense-vgae
	[68]	SPGF	https://github.com/henrykenlay/spgf
	[39]	Nash-Detect	https://github.com/YingtongDou/Nash-Detect
Other Baseline	[51]	FGSM	https://github.com/1Konny/FGSM
	[94]	JSMA	https://github.com/tensorflow/cleverhans
	[8]	Gradient Attack (GA)	https://github.com/bethgelab/foolbox/blob/master/foolbox/attacks/gradient.py
	[46]	First-order	https://github.com/cbfinn/maml
Benchmark	[170]	GRB	https://github.com/THUDM/grb

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 them today. We first give a unified problem formulation for adversarial learning on graph data, and give definitions and taxonomies to categorize the literature on several levels. Next, we summarize most existing imperceptible perturbations evaluation metrics, datasets and discuss several principles about imperceptibility metric. Then, we analyze the contributions and limitations of existing works. Finally, we outline promising future research directions and opportunities that may come from this effort.

Acknowledgments

Thanks to Yixin Liu's (yila22@lehigh.edu) contributions to this work and the future maintenance of this work.

References

[1] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 us election: Divided they blog," in *Proc. 3rd Int. Workshop Link Discov.*, 2005, pp. 36–43.
[2] J. Agterberg, Y. Park, J. Larson, C. White, C. E. Priebe, and V. Lyzinski, "Vertex nomination, consistent estimation, and adversarial modification," 2019, *arXiv:1905.01776*.
[3] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," 2018, *arXiv:1802.00420*.
[4] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.
[5] A. N. Bhagoji, W. He, B. Li, and D. Song, "Exploring the space of black-box attacks on deep neural networks," 2017, *arXiv:1712.09491v1*.
[6] P. Bhardwaj, J. Kelleher, L. Costabello, and D. O'Sullivan, "Adversarial attacks on knowledge graph embeddings via instance attribution methods," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2021, pp. 8225–8239.
[7] P. Bhardwaj, J. Kelleher, L. Costabello, and D. O'Sullivan, "Poisoning knowledge graph embeddings via relation inference patterns," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 1875–1888.
[8] B. Biggio et al., "Evasion attacks against machine learning at test time," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2013, pp. 387–402.
[9] M. Bojarski et al., "End to end learning for self-driving cars," 2016, *arXiv:1604.07316*.
[10] A. Bojchevski and S. Günnemann, "Adversarial attacks on node embeddings via graph poisoning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 695–704.
[11] A. Bojchevski and S. Günnemann, "Certifiable robustness to graph perturbations," in *Proc. Adv. Neural Informat. Process. Syst.*, 2019, pp. 8317–8328.
[12] A. Bojchevski, O. Shchur, D. Zügner, and S. Günnemann, "NetGAN: Generating graphs via random walks," 2018, *arXiv:1803.00816*.
[13] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. Adv. Neural Informat. Process. Syst.*, 2013, pp. 2787–2795.
[14] A. J. Bose, A. Cianflone, and W. Hamiltion, "Generalizable adversarial attacks using generative models," 2019, *arXiv:1905.10864*.
[15] A. J. Bose, H. Ling, and Y. Cao, "Adversarial contrastive estimation," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 1021–1032.
[16] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," 2017, *arXiv:1712.04248*.
[17] A. Breuer, R. Eilat, and U. Weinsberg, "Friend or faux: Graph-based early detection of fake accounts on social networks," in *Proc. Web Conf.*, 2020, pp. 1287–1297.

[18] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, 2017, pp. 3–14.

[19] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy*, 2017, pp. 39–57.

[20] H. Chang et al., "Not all low-pass filters are robust in graph convolutional networks," in *Proc. Adv. Neural Informat. Process. Syst.*, 2021, pp. 25058–25071.

[21] H. Chang et al., "A restricted black-box adversarial framework towards attacking graph embedding models," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 3389–3396.

[22] J. Chen et al., "GA-based Q-attack on community detection," *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 3, pp. 491–503, Jun. 2019.

[23] J. Chen, Y. Chen, L. Chen, M. Zhao, and Q. Xuan, "Multiscale evolutionary perturbation attack on community detection," 2019, *arXiv:1910.09741*.

[24] J. Chen et al., "MGA: Momentum gradient attack on network," 2020, *arXiv:2002.11320*.

[25] J. Chen, Z. Shi, Y. Wu, X. Xu, and H. Zheng, "Link prediction adversarial attack," 2018, *arXiv:1810.01110*.

[26] J. Chen, Y. Wu, X. Lin, and Q. Xuan, "Can adversarial network attack be defended?" 2019, *arXiv:1903.05994*.

[27] J. Chen, Y. Wu, X. Xu, Y. Chen, H. Zheng, and Q. Xuan, "Fast gradient attack on network embedding," 2018, *arXiv:1809.02797*.

[28] J. Chen, J. Zhang, Z. Chen, M. Du, and Q. Xuan, "Time-aware gradient attack on dynamic network link prediction," *IEEE Trans. Knowl. Data Eng.*, early access, Sep. 08, 2021, doi: 10.1109/TKDE.2021.3110580.

[29] L. Chen et al., "A survey of adversarial learning on graphs," 2020, *arXiv:2003.05730*.

[30] L. Chen, J. Li, Q. Peng, Y. Liu, Z. Zheng, and C. Yang, "Understanding structural vulnerability in graph convolutional networks," 2021, *arXiv:2108.06280*.

[31] M. Chen and M. Z. Racz, "Network disruption: Maximizing disagreement and polarization in social networks," 2020, *arXiv:2003.08377*.

[32] Y. Chen et al., "Practical attacks against graph-based clustering," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 1125–1142.

[33] H. Dai et al., "Adversarial attack on graph structured data," 2018, *arXiv:1806.02371*.

[34] Q. Dai, X. Shen, L. Zhang, Q. Li, and D. Wang, "Adversarial training methods for network embedding," in *Proc. World Wide Web Conf.*, 2019, pp. 329–339.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[36] Z. Deng, Y. Dong, and J. Zhu, "Batch virtual adversarial training for graph convolutional networks," 2019, *arXiv:1902.09192*.

[37] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[38] P. Dey and S. Medya, "Manipulating node similarity measures in network," 2019, *arXiv:1910.11529*.

[39] Y. Dou, G. Ma, P. S. Yu, and S. Xie, "Robust spammer detection by nash reinforcement learning," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 924–933.

[40] D. Duvenaud et al., "Convolutional networks on graphs for learning molecular fingerprints," in *Proc. 28th Int. Conf. Neural Informat. Process. Syst.*, 2015, pp. 2224–2232.

[41] N. Entezari, S. A. Al-Sayouri, A. Darvishzadeh, and E. E. Papalexakis, "All you need is low (rank) defending against adversarial attacks on graphs," in *Proc. 13th Int. Conf. Web Search Data Mining*, 2020, pp. 169–177.

[42] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning," *Pattern Recognit.*, vol. 58, pp. 121–134, 2016.

[43] D. Eswaran, S. Günnemann, C. Faloutsos, D. Makhija, and M. Kumar, "Zoobp: Belief propagation for heterogeneous networks," *Proc. VLDB Endowment*, vol. 10, no. 5, pp. 625–636, 2017.

[44] M. Fang, G. Yang, N. Z. Gong, and J. Liu, "Poisoning attacks to graph-based recommender systems," in *Proc. 34th Annu. Comput. Secur. Appl. Conf.*, 2018, pp. 381–392.

[45] F. Feng, X. He, J. Tang, and T.-S. Chua, "Graph adversarial training: Dynamically regularizing based on graph structure," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 6, pp. 2493–2504, Jun. 2021.

[46] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.

[47] J. Fox and S. Rajamanickam, "How robust are graph neural networks to structural noise?" 2019, *arXiv:1912.10206*.

[48] J. Gaitonde, J. Kleinberg, and E. Tardos, "Adversarial perturbations of opinion dynamics in networks," 2020, *arXiv:2003.07010*.

[49] S. Geisler, T. Schmidt, H. Şirin, D. Zügner, A. Bojchevski, and S. Günnemann, "Robustness of graph neural networks at scale," in *Proc. Adv. Neural Informat. Process. Syst.*, 2021, pp. 7637–7649.

[50] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, 2002.

[51] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2015, *arXiv:1412.6572v3*.

[52] Grouplens, Movielens dataset. [Online]. Available: https://bit.ly/2YHzDnZ

[53] V. Gupta and T. Chakraborty, "Adversarial attack on network embeddings via supervised network poisoning," 2021, *arXiv:2102.07164*.

[54] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Informat. Process. Syst.*, 2017, pp. 1024–1034.

[55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[56] X. He, J. Jia, M. Backes, N. Z. Gong, and Y. Zhang, "Stealing links from graph neural networks," 2020, *arXiv:2005.02131*.

[57] S. Hou et al., "αcyber: Enhancing robustness of android malware detection system against adversarial attacks on heterogeneous graph based model," in *Proc. 28th ACM Int. Conf. Informat. Knowl. Manage.*, 2019, pp. 609–618.

[58] B. Hu, Z. Zhang, C. Shi, J. Zhou, X. Li, and Y. Qi, "Cash-out user detection based on attributed heterogeneous information network with a hierarchical attention mechanism," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 946–953.

[59] V. N. Ioannidis, D. Berberidis, and G. B. Giannakis, "Graphsac: Detecting anomalies in large-scale graphs," 2019, *arXiv:1910.09589*.

[60] V. N. Ioannidis and G. B. Giannakis, "Edge dithering for robust adaptive graph convolutional networks," 2019, *arXiv:1910.09590*.

[61] V. N. Ioannidis, A. G. Marques, and G. B. Giannakis, "Tensor graph convolutional networks for multi-relational and robust learning," 2020, *arXiv:2003.07729*.

[62] J. Jia, B. Wang, X. Cao, and N. Z. Gong, "Certified robustness of community detection against adversarial structural perturbation via randomized smoothing," 2020, *arXiv:2002.03421*.

[63] H. Jin and X. Zhang, "Latent adversarial training of graph convolution networks," in *Proc. Int. Conf. Mach. Learn. Workshop Learn. Reasoning Graph-Structured Representations*, 2019.

[64] M. Jin, H. Chang, W. Zhu, and S. Sojoudi, "Power up! robust graph convolutional network against evasion attacks based on graph powering," 2019, *arXiv:1905.10029*.

[65] W. Jin, T. Derr, Y. Wang, Y. Ma, Z. Liu, and J. Tang, "Node similarity preserving graph convolutional networks," in *Proc. 14th ACM Int. Conf. Web Search Data Mining*, 2021, pp. 148–156.

[66] W. Jin, Y. Li, H. Xu, Y. Wang, and J. Tang, "Adversarial attacks and defenses on graphs: A review and empirical study," 2020, *arXiv:2003.00653*.

[67] W. Jin, Y. Ma, X. Liu, X. Tang, S. Wang, and J. Tang, "Graph structure learning for robust graph neural networks," 2020, *arXiv:2005.10203*.

[68] H. Kenlay, D. Thanou, and X. Dong, "On the stability of polynomial spectral graph filters," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 5350–5354.

[69] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[70] D. Koutra, A. Parikh, A. Ramdas, and J. Xiang, "Algorithms for graph similarity and subgraph matching," in *Proc. Ecol. Inference Conf.*, 2011.

[71] Valdis E Krebs, "Mapping networks of terrorist cells," *Connections*, vol. 24, no. 3, pp. 43–52, 2002.

[72] C. Kumar, R. Ryan, and M. Shao, "Adversary for social good: Protecting familial privacy through joint adversarial attacks," in *Proc. Conf. Artif. Intell.*, 2020, pp. 11 304–11 311.

[73] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Trans. Knowl. Discov. Data*, vol. 1, no. 1, pp. 2–es, 2007.

[74] A. Li, Z. Qin, R. Liu, Y. Yang, and D. Li, "Spam review detection with graph convolutional networks," in *Proc. 28th ACM Int. Conf. Informat. Knowl. Manage.*, 2019, pp. 2703–2711.

[75] J. Li, H. Zhang, Z. Han, Y. Rong, H. Cheng, and J. Huang, "Adversarial attack on community detection by hiding individuals," in *Proc. Int. World Wide Web Conf.*, 2020, pp. 917–927.

[76] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," 2018, *arXiv:1707.01926v3*.

[77] P. Liao et al., "Information obfuscation of graph neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 6600–6610.

[78] G. Litjens et al., "A survey on deep learning in medical image analysis," *Med. Image Anal.*, vol. 42, pp. 60–88, 2017.

[79] X. Liu et al., "Graph neural networks with adaptive residual," in *Proc. Adv. Neural Informat. Process. Syst.*, 2021, pp. 9720–9733.

[80] X. Liu et al., "Elastic graph neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 6837–6849.

[81] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," 2016, *arXiv:1611.02770*.

[82] A. Logins, Y. Li, and P. Karras, "On the robustness of cascade diffusion under node attacks," in *Proc. Web Conf.*, 2020, pp. 2711–2717.

[83] D. Lusseau et al., "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behav. Ecol. Sociobiology*, vol. 54, no. 4, pp. 396–405, 2003.

[84] J. Ma, S. Ding, and Q. Mei, "Towards more practical adversarial attacks on graph neural networks," 2020, *arXiv:2006.05057*.

[85] Y. Ma, S. Wang, L. Wu, and J. Tang, "Attacking graph convolutional networks via rewiring," 2019, *arXiv:1906.03750*.

[86] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*.

[87] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore, "Automating the construction of internet portals with machine learning," *Informat. Retrieval*, vol. 3, no. 2, pp. 127–163, 2000.

[88] B. A. Miller, M. Çamurcu, A. J. Gomez, K. Chan, and T. Eliassi-Rad, "Improving robustness to attacks against vertex classification," in *Proc. MLG Workshop Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1–8.

[89] B. A. Miller, M. Çamurcu, A. J. Gomez, K. Chan, and T. Eliassi-Rad, "Topological effects on attacks against vertex classification," 2020, *arXiv:2003.05822*.

[90] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: Review, opportunities and challenges," *Brief. Bioinf.*, vol. 19, pp. 1236–1246, 2017.

[91] T. Miyato, S.-I. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: A regularization method for supervised and semi-supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1979–1993, Aug. 2019.

[92] J. Mu, B. Wang, Q. Li, K. Sun, M. Xu, and Z. Liu, "A hard label black-box adversarial attack against graph neural networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2021, pp. 108–125.

[93] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2017, pp. 506–519.

[94] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Secur. Privacy (Eur.&P)*, 2016, pp. 372–387.

[95] S. Peng and T. Mine, "A robust hierarchical graph convolutional network model for collaborative filtering," 2020, *arXiv:2004.14734*.

[96] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," 2014, *arXiv:1403.6652v2*.

[97] P. Pezeshkpour, Y. Tian, and S. Singh, "Investigating robustness and interpretability of link prediction via adversarial modifications," 2019, *arXiv:1905.00563*.

[98] M. Raman et al., "Learning to deceive knowledge graph augmented models via targeted perturbation," in *Proc. Int. Conf. Learn. Representations*, 2020.

[99] S. Rayana and L. Akoglu, "Collective opinion spam detection: Bridging review networks and metadata," in *Proc. 21th ACM sigkdd Int. Conf. Knowl. Discov. Data Mining*, 2015, pp. 985–994.

[100] J. Ren et al., "Integrated defense for resilient graph matching," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8982–8997.

[101] K. Riesen and H. Bunke, "Iam graph database repository for graph based pattern recognition and machine learning," in *Proc. Joint IAPR Int. Workshops Statist. Techn. Pattern Recognit. Struct. Syntactic Pattern Recognit.*, 2008, pp. 287–297.

[102] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," 2018, *arXiv:1805.06605*.

[103] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI Mag.*, vol. 29, no. 3, pp. 93–93, 2008.

[104] C. Shi, Y. Li, J. Zhang, Y. Sun, and S. Y. Philip, "A survey of heterogeneous information network analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 1, pp. 17–37, Jan. 2017.

[105] K. Sun, Z. Lin, H. Guo, and Z. Zhu, "Virtual adversarial training on graph convolutional networks in node classification," in *Proc. Chin. Conf. Pattern Recognit. Comput. Vis.*, 2019, pp. 431–443.

[106] L. Sun, Z. Li, Q. Yan, W. Srisa-an, and Y. Pan, "Sigpid: Significant permission identification for android malware detection," in *Proc. IEEE 11th Int. Conf. Malicious Unwanted Softw.*, 2016, pp. 1–8.

[107] L. Sun, Y. Wang, B. Cao, S Y. Philip, W. Srisa-An, and A. D. Leow, "Sequential keystroke behavioral biometrics for mobile user identification via multi-view deep learning," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2017, pp. 228–240.

[108] M. Sun et al., "Data poisoning attack against unsupervised node embedding methods," 2018, *arXiv:1810.12881*.

[109] Y. Sun, S. Wang, X. Tang, T.-Y. Hsieh, and V. Honavar, "Non-target-specific node injection attacks on graph neural networks: A hierarchical reinforcement learning approach," in *Proc. Int. World Wide Web Conf.*, 2020.

[110] J. J. Sutherland, L. A. O'brien, and D. F. Weaver, "Spline-fitting with a genetic algorithm: A method for developing classification structure- activity relationships," *J. Chem. Informat. Comput. Sci.*, vol. 43, no. 6, pp. 1906–1915, 2003.

[111] C. Szegedy et al., "Intriguing properties of neural networks," 2014, *arXiv:1312.6199v4*.

[112] T. Takahashi, "Indirect adversarial attacks via poisoning neighbors for graph convolutional networks," in *Proc. IEEE Int. Conf. Big Data*, 2019, pp. 1395–1400.

[113] H. Tang et al., "Adversarial attack on hierarchical graph pooling neural networks," 2020, *arXiv:2005.11560*.

[114] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: Extraction and mining of academic social networks," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2008, pp. 990–998.

[115] X. Tang, Y. Li, Y. Sun, H. Yao, P. Mitra, and S. Wang, "Transferring robustness for graph neural network against poisoning attacks," in *Proc. 13th Int. Conf. Web Search Data Mining*, 2020, pp. 600–608.

[116] S. Tao, Q. Cao, H. Shen, J. Huang, Y. Wu, and X. Cheng, "Single node injection attack against graph neural networks," in *Proc. 30th ACM Int. Conf. Informat. Knowl. Manage.*, 2021, pp. 1794–1803.

[117] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

[118] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," 2018, *arXiv:1809.10341*.

[119] X. Wan, H. Kenlay, B. Ru, A. Blaas, M. A. Osborne, and X. Dong, "Adversarial attacks on graph classification via Bayesian optimisation," 2021, *arXiv:2111.02842*.

[120] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," 2018, *arXiv:1804.07461*.

[121] B. Wang and N. Z. Gong, "Attacking graph-based classification via manipulating the graph structure," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2019, pp. 2023–2040.

[122] B. Wang, J. Jia, X. Cao, and N. Z. Gong, "Certified robustness of graph neural networks against adversarial structural perturbation," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2021, pp. 1645–1653.

[123] J. Wang, M. Luo, F. Suya, J. Li, Z. Yang, and Q. Zheng, "Scalable attack on graph data by injecting vicious nodes," 2020, *arXiv:2004.13825*.

[124] S. Wang et al., "Adversarial defense framework for graph neural network," 2019, *arXiv:1905.03679*.

[125] X. Wang, J. Eaton, C.-J. Hsieh, and F. Wu, "Attack graph convolutional networks by adding fake nodes," 2018, *arXiv:1810.10751*.

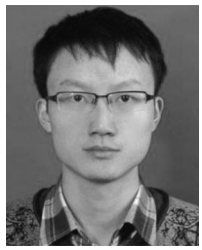
[126] X. Wang, X. Liu, and C.-J. Hsieh, "Graphdefense: Towards robust graph convolutional networks," 2019, *arXiv:1911.04429*.
[127] M. Waniek, T. P. Michalak, M. J. Wooldridge, and T. Rahwan, "Hiding individuals and communities in a social network," *Nat. Hum. Behav.*, vol. 2, no. 2, pp. 139–147, 2018.
[128] M. Waniek, K. Zhou, Y. Vorobeychik, E. Moro, T. P. Michalak, and T. Rahwan, "Attack tolerance of link prediction algorithms: How to hide your relations in a social network," 2018, *arXiv:1809.00152*.
[129] M. Weber et al., "Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics," 2019, *arXiv:1908.02591*.
[130] Wikipedia, Average precision. [Online]. Available: https://bit.ly/2Uz06lL
[131] Wikipedia, Confusion matrix. [Online]. Available: https://bit.ly/2wHUpcf
[132] Wikipedia, Mean reciprocal rank. [Online]. Available: https://bit.ly/3aBadMk
[133] Wikipedia, Modularity. [Online]. Available: https://bit.ly/3dMbsdB)
[134] Wikipedia, Mutual information. [Online]. Available: https://bit.ly/3bBeDCY
[135] Wikipedia, ndcg. [Online]. Available: https://bit.ly/3dKYqf6
[136] Wikipedia, Rand index. [Online]. Available: https://bit.ly/3azqoK6
[137] Wikipedia, Roc. [Online]. Available: https://bit.ly/341yHfa
[138] F. Wu, T. Zhang, A. H. D. S. Jr, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," 2019, *arXiv:1902.07153*.
[139] H. Wu, C. Wang, Y. Tyshetskiy, A. Docherty, K. Lu, and L. Zhu, "Adversarial examples for graph data: Deep insights into attack and defense," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 4816–4823.
[140] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," 2019, *arXiv:1901.00596*.
[141] Z. Xi, R. Pang, S. Ji, and T. Wang, "Graph backdoor," 2020, *arXiv:2006.11890*.
[142] C. Xiao, J.-Y. Zhu, B. Li, W. He, M. Liu, and D. Song, "Spatially transformed adversarial examples," 2018, *arXiv:1801.02612*.
[143] H. Y. Xiong et al., "The human splicing code reveals new insights into the genetic determinants of disease," *Science*, vol. 347, 2015, Art. no. 6218.
[144] H. Xu, L. Xiang, J. Yu, A. Cao, and X. Wang, "Speedup robust graph structure learning with low-rank information," in *Proc. 30th ACM Int. Conf. Informat. Knowl. Manage.*, 2021, pp. 2241–2250.
[145] J. Xu et al., "Unsupervised adversarially-robust representation learning on graphs," 2020, *arXiv:2012.02486*.
[146] K. Xu et al., "Topology attack and defense for graph neural networks: An optimization perspective," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 3961–3967.
[147] X. Xu et al., "Edog: Adversarial edge detection for graph neural networks," Lawrence Livermore National Lab.(LLNL), Livermore, CA, USA, Tech. Rep., 2020.
[148] X. Xu, Y. Yu, B. Li, L. Song, C. Liu, and C. Gunter, "Characterizing malicious edges targeting on graph neural networks," *Openreview*, 2018.
[149] Q. Xuan, Y. Shan, J. Wang, Z. Ruan, and G. Chen, "Adversarial attacks to scale-free networks: Testing the robustness of physical criteria," 2020, *arXiv:2002.01249*.
[150] Q. Xuan et al., "Unsupervised euclidean distance attack on network embedding," 2019, *arXiv:1905.11015*.
[151] N. Yadati, M. Nimishakavi, P. Yadav, V. Nitin, A. Louis, and P. Talukdar, "HyperGCN: A new method for training graph convolutional networks on hypergraphs," in *Proc. Adv. Neural Informat. Process. Syst.*, 2019, pp. 1509–1520.
[152] Z. Yang, W. W. Cohen, and R. Salakhutdinov, "Revisiting semi-supervised learning with graph embeddings," 2016, *arXiv:1603.08861*.
[153] S. Yu et al., "Target defense against link-prediction-based attacks via evolutionary perturbations," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 2, pp. 754–767, Feb. 2021.
[154] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthropological Res.*, vol. 33, no. 4, pp. 452–473, 1977.
[155] X. Zang, Y. Xie, J. Chen, and B. Yuan, "Graph universal adversarial attacks: A few bad actors ruin graph learning models," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021.
[156] A. Zhang and J. Ma, "Defensevgae: Defending against adversarial attacks on graph data via a variational graph autoencoder," 2020, *arXiv:2006.08900*.
[157] H. Zhang et al., "Projective ranking: A transferable evasion attack method on graph neural networks," in *Proc. 30th ACM Int. Conf. Informat. Knowl. Manage.*, 2021, pp. 3617–3621.
[158] H. Zhang et al., "Data poisoning attack against knowledge graph embedding," 2019, *arXiv:1904.12052*.
[159] S. Zhang, H. Yin, T. Chen, Z. Huang, L. Cui, and X. Zhang, "Graph embedding for recommendation against attribute inference attacks," in *Proc. Web Conf.*, 2021, pp. 3002–3014.
[160] S. Zhang, H. Yin, T. Chen, Q. V. N. Hung, Z. Huang, and L. Cui, "GCN-based user representation learning for unifying robust recommendation and fraudster detection," 2020, *arXiv:2005.10150*.
[161] X. Zhang and M. Zitnik, "Gnnguard: Defending graph neural networks against adversarial attacks," 2020, *arXiv:2006.08149*.
[162] Y. Zhang, Y. Fan, Y. Ye, L. Zhao, and C. Shi, "Key player identification in underground forums over attributed heterogeneous information network embedding framework," in *Proc. 28th ACM Int. Conf. Informat. Knowl. Manage.*, 2019, pp. 549–558.
[163] Y. Zhang, S. Khan, and M. Coates, "Comparing and detecting adversarial attacks for graph deep learning," in *Proc. Representation Learn. Graphs Manifolds Workshop*, 2019.
[164] Y. Zhang, F. Regol, S. Pal, S. Khan, L. Ma, and M. Coates, "Detection and defense of topological adversarial attacks on graphs," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 2989–2997.
[165] Z. Zhang, J. Jia, B. Wang, and N. Z. Gong, "Backdoor attacks to graph neural networks," 2020, *arXiv:2006.11165*.
[166] Z. Zhang et al., "Adversarial attack against cross-lingual knowledge graph alignment," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2021, pp. 5320–5337.
[167] Z. Zhang, Z. Zhang, Y. Zhou, Y. Shen, R. Jin, and D. Dou, "Adversarial attacks on deep graph matching," in *Proc. Adv. Neural Informat. Process. Syst.*, 2020, pp. 20 834–20 851.
[168] K. Zhao et al., "Structural attack against graph based android malware detection," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2021, pp. 3218–3235.
[169] X. Zhao et al., "Expressive 1-lipschitz neural networks for robust multiple graph learning against adversarial attacks," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 12 719–12 735.
[170] Q. Zheng et al., "Graph robustness benchmark: Benchmarking the adversarial robustness of graph machine learning," in *Proc. 35th Conf. Neural Informat. Process. Syst. Datasets Benchmarks Track*, 2021.
[171] K. Zhou, T. P. Michalak, and Y. Vorobeychik, "Adversarial robustness of similarity-based link prediction," 2019, *arXiv:1909.01432*.
[172] K. Zhou, T. P. Michalak, M. Waniek, T. Rahwan, and Y. Vorobeychik, "Attacking similarity-based link prediction in social networks," in *Proc. 18th Int. Conf. Auton. Agents MultiAgent Syst.*, 2019, pp. 305–313.
[173] Y. Zhou et al., "Robust network alignment via attack signal scaling and adversarial perturbation elimination," in *Proc. Web Conf.*, 2021, pp. 3884–3895.
[174] D. Zhu, Z. Zhang, P. Cui, and W. Zhu, "Robust graph convolutional networks against adversarial attacks," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1399–1407.
[175] X. Zou et al., "TDGIA: Effective injection attacks on graph neural networks," 2021, *arXiv:2106.06663*.
[176] D. Zügner, A. Akbarnejad, and S. Günnemann, "Adversarial attacks on neural networks for graph data," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2847–2856.
[177] D. Zügner, O. Borchert, A. Akbarnejad, and S. Guennemann, "Adversarial attacks on graph neural networks: Perturbations and their patterns," *ACM Trans. Knowl. Discov. Data*, vol. 14, pp. 1–31, 2020.
[178] D. Zügner and S. Günnemann, "Adversarial attacks on graph neural networks via meta learning," 2019, *arXiv:1902.08412*.
[179] D. Zügner and S. Günnemann, "Certifiable robustness and robust training for graph convolutional networks," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 246–256.

Lichao Sun received the BS and MS degrees from the University of Nebraska Lincoln, and the PhD degree in computer science from the University of Illinois, Chicago in 2020, under the supervision of Prof. Philip S. Yu. He is currently an assistant professor with the Department of Computer Science and Engineering, Lehigh University. His research interests include security and privacy in deep learning and data mining. He mainly focuses on AI security and privacy, social networks, and natural language processing applications. He has published more than 45 research articles in top conferences and journals like CCS, USENIX-Security, NeurIPS, KDD, ICLR, AAAI, IJCAI, ACL, NAACL, *IEEE Transactions on Industrial Informatics*, *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Mobile Computing*.

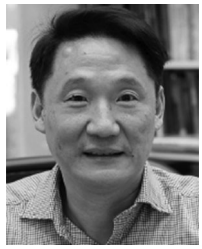
Yingtong Dou received the BE degree in IoT engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2017. He is currently working toward the PhD degree in computer science with the University of Illinois at Chicago, Chicago, IL, USA. He has published several papers in top-tier conferences including KDD, WWW, SIGIR, and CIKM. His research interest includes graph mining, fraud detection, social media analysis, and machine learning security.

Carl Yang received the BEng degree in computer science and engineering from Zhejiang University in 2014 and the PhD degree in computer science from the University of Illinois, Urbana-Champaign in 2020. He is an assistant professor with Emory University. His research interests span graph data mining, applied machine learning, knowledge graphs and federated learning, with applications in recommender systems, biomedical informatics, neuroscience, and healthcare. His research results have been published in top venues like *IEEE Transactions on Knowledge and Data Engineering*, KDD, WWW, NeurIPS, ICML, ICLR, ICDE, SIGIR and ICDM. He also received the Dissertation Completion Fellowship of UIUC in 2020, the Best Paper Award of ICDM in 2020, the Dissertation award finalist of KDD in 2021, and the Best Paper Award of KDD Health Day in 2022.

Kai Zhang is currently working toward the PhD degree with Lehigh University, US. His research interests include machine learning and data mining. He recently focuses on federated learning, security & data privacy in machine learning, and Edge AI. Before joining Lehigh, he was a reearch assistatn with Embry-Riddle Aeronautical University working on large-scale flight dispatching for disaster evacuation with big aviation data. The system design has been featured by the Annual National Mobility Summit and the National Transportation Library.

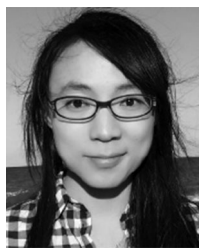


Ji Wang received the PhD degree in information system from the National University of Defense Technology, Changsha, China, in 2019. He is currently an assistant professor with the College of Systems Engineering, National University of Defense Technology. His research interests include deep learning and edge intelligence. He has published more than 20 research articles in refereed journals and conference proceedings such as *IEEE Transactions on Computers*, *IEEE Transactions on Parallel and Distributed Systems*, SIGKDD, and AAAI. He was a visiting PhD student with the University of Illinois at Chicago from March 2017 to September 2018 under the supervision of Prof. Philip S. Yu.



Philip S. Yu (Fellow, IEEE) received the BS Degree in E.E. from National Taiwan University, the MS and PhD degrees in E.E. from Stanford University, and the MBA degree from New York University. He is a distinguished professor in computer science with the University of Illinois at Chicago and also holds the Wexler Chair in Information Technology. Before joining UIC, He was with IBM, where he was manager with the Software Tools and Techniques department, Watson Research Center. His research interest is on Big Data, including data mining, data stream, database, and privacy. He has published more than 1,200 papers in refereed journals and conferences. He holds or has applied for more than 300 US patents. He is a fellow of the ACM. He is the recipient of ACM SIGKDD 2016 Innovation Award for his influential research and scientific contributions on mining, fusion and anonymization of Big Data. He also received the ICDM 2013 10-year Highest-Impact Paper Award, and the EDBT Test of Time Award (2014). He was the editor-in-chiefs of *ACM Transactions on Knowledge Discovery from Data* (2011-2017) and *IEEE Transactions on Knowledge and Data Engineering* (2001-2004).

Lifang He is currently an assistant professor with the Department of Computer Science and Engineering, Lehigh University. Before her current position, Dr. He worked as a postdoctoral researcher with the Department of Biostatistics and Epidemiology, the University of Pennsylvania. Her current research interests include machine learning, data mining, tensor analysis, with major applications in biomedical data, and neuroscience. She has published more than 100 papers in refereed journals and conferences, such as NIPS, ICML, KDD, CVPR, WWW, IJCAI, AAAI, *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Image Processing* and AMIA.



Bo Li is an assistant professor with the department of Computer Science, University of Illinois at Urbana–Champaign, and is a recipient of the Symantec Research Labs Fellowship, Rising Stars, MIT TR 35, and best paper awards in several machine learning and security conferences. Previously she was a postdoctoral researcher with UC Berkeley. Her research focuses on both theoretical and practical aspects of security, machine learning, privacy, game theory, and adversarial machine learning. She has designed several robust learning algorithms, a scalable framework for achieving robustness for a range of learning methods, and a privacy preserving data publishing system. Her work have been featured by major publications and media outlets such as Nature, Wired, Fortune, and IEEE Spectrum.