Artificial Intelligence Review (2024) 57:40
https://doi.org/10.1007/s10462-023-10656-4

Graph neural networks: a survey on the links between privacy and security

Faqian Guan[1] · **Tianqing Zhu**[2,3] · **Wanlei Zhou**[3] · **Kim-Kwang Raymond Choo**[4]

Accepted: 19 December 2023 / Published online: 8 February 2024

Abstract

Graph neural networks (GNNs) are models that capture the dependencies between graph data by passing messages between graph nodes and they have been widely used to process graph data that contains relational information. Example application areas include social networks, recommendation systems, and life sciences. However, like all neural networks, there are underpinning security and privacy concerns associated with GNN deployments in practice. For example, attackers can perturb a graph's data to undermine a model's effectiveness, or they can steal the model's data and/or parameters, thus threatening the privacy of the model. In this survey, we provide a comprehensive review of recent research efforts on security and/or privacy in GNNs. We also systematically describe the distinctions and relationships between security and privacy, as well as providing an outlook on future directions of research in this area.

Keywords Graph neural networks · Security · Privacy · Literature Survey

✉ Tianqing Zhu
tianqing.zhu@ieee.org

Faqian Guan
faqianguan@gmail.com

Wanlei Zhou
wlzhou@cityu.edu.mo

Kim-Kwang Raymond Choo
raymond.choo@fulbrightmail.org

[1] School of Computer Science, China University of Geosciences, No. 388 Lumo Road, Wuhan 430074, Hubei, People's Republic of China

[2] School of Computer Science, University of Technology Sydney, 15 Broadway, Sydney, NSW 2008, Australia

[3] School of Data Science, City University of Macau, Taipei, Macau

[4] Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249-0631, USA

1 Introduction

With the exponential growth of data and advancements in decentralized information sharing technology, machine learning has rapidly evolved in recent decades (Jordan et al. 2015). The core of machine learning involves learning patterns from large amounts of data, constructing a model, and iteratively refining its accuracy. However, many of the data resources available to researchers are non-Euclidean, which poses significant challenges for existing machine learning models (Wu et al. 2022). Hamilton (2017) defined Euclidean data as a class of data that exhibits good translation invariance, while non-Euclidean data is defined as data that fails to satisfy translation invariance. The lack of translation invariance in non-Euclidean data makes it incompatible with models derived from Euclidean data (Wu et al. 2022). To address this challenge, Graph Neural Networks (GNNs) (Scarselli et al. 2009) have been proposed as an effective approach for modeling typical types of non-Euclidean data, such as graph data. As such, GNNs have achieved outstanding results in various fields, such as computer vision (Shi et al. 2019, 2020), natural language understanding (Schlichtkrull et al. 2021; Wu et al. 2021), social networks (Wu et al. 2020; Hamilton et al. 2017), and recommendation systems (Wu et al. 2020; Ying et al. 2018; Fan et al. 2019).

However, while deep learning techniques bring convenience to people's lives, they also face challenges - security and privacy risks being key among them. For example, attackers can modify a deep learning model's data to make its predictions deviate from expectations (Goodfellow et al. 2015; Carlini and Wagner 2017; Papernot et al. 2016). Moreover, model extraction attacks (Tramèr et al. 2016), membership inference attacks (Shokri et al. 2017), and attribute inversion attacks (Fredrikson et al. 2014) can violate the privacy of sensitive data and can even result in a model's theft.

Compared to other deep learning techniques, security and privacy with GNNs is particularly challenging (Günnemann 2022). This is because graph data have a more expansive perturbation space that extends to the graph's structure and its attributes. And perturbing a graph's structure is often performed in discrete domains, which leads to complex optimization problems. Additionally, the core of a GNN lies in exploiting the interdependencies between graph data, which are identified through, for example, neighborhood aggregation. This means that perturbing one part of the graph often affects the other parts.

In addition, generating perturbation examples often incurs a cost, and attackers aim to generate effective adversarial examples at minimal cost to ensure their attacks remain imperceptible. Therefore, it is crucial to calculate the similarity between the original example and the perturbed adversarial example to determine if the perturbation is within the allowable cost. In the case of image data, the similarity can be calculated by comparing them pixel by pixel or using methods such as structural similarity index measurement (Wang et al. 2004). For instance, when an image model identifies an image of a cat as a cat, an attacker can create an adversarial example by perturbing the image pixels to mislead the model into predicting cat as a dog. In such a scenario, similarity calculation using structural similarity index measurement methods can be applied to determine whether the perturbation is within the cost.

However, in the case of graph data, adversarial examples can be generated by perturbing the topology of the graph and modifying node features. Attackers may add or remove edges to change the relationship between nodes, alter node attributes such as age, gender, and occupation, or even introduce new nodes. The impact of adjacency matrix and node feature modifications can vary depending on the situation, making it challenging to calculate the similarity between the original and perturbed examples. As a result, determining whether the perturbation is within the allowable cost becomes difficult.

In recent years, there has been a significant research focus on the security and privacy issues associated with GNNs. The concept of security in GNNs refers to the prevention of malicious attacks on the model or system, while privacy involves protecting individuals from unauthorized access to their sensitive data and personal information. Security concerns can have a negative impact on model performance, and attacks on GNN security can cause deviations from predictions, affecting the use of the models. Attackers generate adversarial samples by perturbing the adjacency matrix (Dai et al. 2018; Wang and Gong 2019; Wu et al. 2019) and/or node features (Wu et al. 2019; Bose et al. 2019; Ma et al. 2020) of the graph data, which can impact model predictions and usage. Defenders utilize methods such as graph data inspection or studying GNN messaging and aggregation methods to obtain a more robust GNN model (Zhang et al. 2022a; Zhu et al. 2019).

GNN privacy, on the other hand, involves protecting the confidentiality of data and model parameters. Attackers can access the model through available information and infer GNN model information, such as the adjacency matrix (He et al. 2021) and node features (Duddu et al. 2020) of the training graph, from the responded information to obtain privacy information, such as the age and gender of the user in the social network graph. Defenders of GNN privacy restrict response information to prevent such privacy attacks. Techniques such as differential privacy (Zhang et al. 2021a; He et al. 2021) can be used to ensure that only authorized parties can access sensitive data and model parameters.

The privacy and security of GNNs are strongly interconnected, as security breaches can lead to privacy violations, and security attacks can be employed to defend against privacy attacks (Jia et al. 2019; Wu et al. 2018). Additionally, privacy attacks can strengthen security attacks (Chang et al. 2020). Thus, it is critical to address both security and privacy issues in GNNs to ensure comprehensive protection.

Unfortunately, though, most existing surveys on GNNs are biased toward the application side of GNNs (Zhou et al. 2020; Waikhom and Patgiri 2021; Wu et al. 2020, 2021). Zhou et al. (2020) and Waikhom and Patgiri (2021) classified various GNN methods and discussed their applications and prospects. Wu et al. (2020) reviewed the application of GNNs to recommendation systems, while Wu et al. (2021) presented and organized research on GNNs for natural language processing. However, the privacy and security of GNNs are crucial research directions, and targeted reviews of security and privacy in this area tend to focus on only one of these areas. For instance, Sun et al. (2018) described some adversarial attacks on graph data in a review of studies on GNNs. On the other hand, Günnemann (2022) provided detailed descriptions of the robustness of GNNs. Moreover, numerous surveys have been conducted on the security and privacy of other deep learning techniques, such as social networks (Kayes 2017), federated learning (Mothukuri et al. 2021), and general adversarial networks (Cai et al. 2021). However, no survey has systematically discussed the security and privacy of GNNs in detail, including the correlations and distinctions between the two issues. The only exception is the work of Dai et al. (2022), which provides an introduction to privacy and security in the context of GNNs. However, they did not explain the relationship between privacy and security, and they did not distinguish between the background and attack methods in their specific classification of security. Their categorization of adversarial attacks includes four groups at the same level: white-box attacks, black-box attacks, evasion attacks, and poisoning attacks. Nevertheless, this classification may not be considered entirely reasonable, as evasion attacks or poisoning attacks can exhibit characteristics of both black-box and white-box attacks, depending on the knowledge possessed by the attacker (Cai et al. 2021). A summary of relevant surveys and the material covered is shown in Table 1.

Our contribution with this survey includes:

- Systematic description and analysis of recent research on GNN security and privacy, with a focus on the connections and differences between these two issues to fill gaps in previous literature.
- A more logical and comprehensive classification of adversarial attacks on GNNs, which separates the background knowledge required for attacks from the specific classification of adversarial attacks.
- A novel classification of adversarial defense for GNNs that takes into account the models' characteristics and the phases of defense occurrence.
- A review of recent research on GNN privacy, along with a reasonable classification and introduction to the topic.
- An outlook on future work in GNN security and privacy.

The rest of this survey is structured as follows. The graph base definition and GNN with its variations are introduced in Sect. 2. The security and privacy and their relationship are introduced in Sect. 3. The security adversarial attacks and defenses are introduced in Sect. 4. Privacy attacks and defenses are presented in Sect. 5. The future research work is described in Sect. 6. The last part is a conclusion of the survey.

2 Preliminaries

2.1 Notations

A graph dataset is defined as $D = \left\{G_i\right\}_{i=1}^{M}$, where M is the number of graphs. Here, the graph can be represented by $G = \{V, E\}$, where $V = \left\{V_i\right\}_{i=1}^{N}$ represents the set of nodes

Table 1 Summary of relevant surveys

Literature	Security	Privacy	Links between privacy and security	GNN
Zhou et al. (2020)	×	×	×	✓
Waikhom and Patgiri (2021)	×	×	×	✓
Wu et al. (2020)	×	×	×	✓
Wu et al. (2021)	×	×	×	✓
Sun et al. (2018)	✓	×	×	✓
Günnemann (2022)	✓	×	×	✓
Mothukuri et al. (2021)	✓	✓	×	×
Cai et al. (2021)	✓	✓	×	×
Kayes (2017)	✓	✓	✓	×
Dai et al. (2022)	✓	✓	×	✓
Our	✓	✓	✓	✓

Links between Privacy & Security indicates whether or not the review describes the relationship between privacy and security in the graph, N is the number of nodes, $E = \left\{e_j\right\}_{j=1}^{K}$ represents the set of edges, and K is the number of edges, $0 \leq K \leq N^2$.

In graph neural networks, the graph is typically represented as $G = \{A, X\}$, where $A \in \mathbb{R}^{N\times N}$ is the adjacency matrix of the graph. An entry of A with a value of 0 indicates the absence of an edge between two nodes, while non-zero entries represent the existence of an edge, with the specific value potentially indicating the edge weight. Most of the graph datasets used in privacy and security research are unweighted graphs, where each edge is represented as a binary connection between two nodes. Specifically, the value 1 is typically used to indicate the presence of an edge, while 0 indicates the absence of an edge. $\hat{A}$ represents the normalized adjacency matrix of the graph, which incorporates both the degree information of each vertex in the graph and their adjacency information. The normalized adjacency matrix is obtained by applying a normalization technique to the adjacency matrix of the graph. The specific calculation method is described in Sect. 2.4.1. $X\epsilon\mathbb{R}^{N\times S}$ is the attribute of the graph, and S is the size of the feature space for each attribute vector. Y denotes a different label for node-level, link-level, and graph-level tasks as introduced in the next section. The main notations used in this paper are listed in Table 2.

2.2 Tasks of graph neural networks

In recent years, GNNs have been very widely used. Hence, to study the security and privacy issues associated with GNNs, we first need to understand the different types of GNN tasks. As shown in Fig. 1, the tasks performed with GNNs can be divided into three categories based on the characteristics of the graph data: node-level tasks, graph-level tasks, and link-level tasks.

Table 2 Key notations in survey

Notation	Description
D, D*	Dataset, Auxiliary Dataset
G, G$'$, G$_p$	Graph, Perturbation Graph, Subgraph
A	Adjacency matrix representation of graph
$\hat{\text{A}}$	The normalized adjacency matrix
V, E	A set of graph nodes, graph edges
X	The node feature matrix of the graph
Y	The label matrix of the graph
H	Node representation in GNN
$\mathcal{N}(v)$	All neighboring nodes of node v
W	Parameters of GNN
Δ	Perturbation cost
σ	Activation function
$\odot$	Element-wise multiplication operation
$\mathcal{M}$	The prediction model
$\|\|$	Vector concatenation

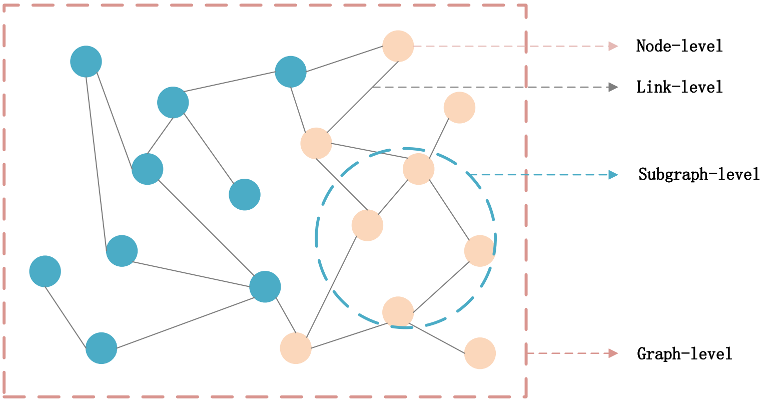

Fig. 1 Tasks of GNN

2.3 Basic structure of GNN

Graph Neural Networks (GNNs) (Scarselli et al. 2009) extend existing neural network methods to process data represented in the graph domain (Battaglia et al. 2018). By using a messaging mechanism, GNNs capture node features and topology information of the graph. More specifically, at each layer, GNNs update the node representation by aggregating the information of neighboring nodes. The updated node representation in the k-th layer of the GNN can be formulated as follows:

$$\mathrm{h}_v^{(k)} = \mathrm{COMBINE}^{(k)}\left(\mathrm{h}_v^{(k-1)},\ \mathrm{AGG}^{\,(k-1)}\left(\left\{\mathrm{h}_u^{(k-1)}\ :\ \forall u \in \mathcal{N}(v) \cup v\right\}\right)\right) \tag{1}$$

where $h_v^{(k)}$ is the representation of node $v \in \mathcal{N}(v)$ in the k-th layer, $h_v^{(0)}$ denotes the input feature of node V. $\mathcal{N}(v)$ denotes all neighboring nodes of node v.

2.4 Variant models of GNN

Inspired by GNN, a number of variants of GNN have been proposed in different application scenarios. In this section, we introduce GNN variants that are often applied in security and privacy domains.

2.4.1 Graph convolutional network (GCN)

GCN (Kipf and Welling 2016) is one of the most widely used GNN variants that employs a variant of convolutional neural network to learn graph-structured data. The convolutional network structure is determined by a local first-order approximation of spectral graph convolution. The hidden layer representation is achieved by encoding the local graph structure and node features. Specifically, each layer of GCN can be formulated as:

$$\begin{aligned} &\mathrm{H}^{(k)} = \sigma\left(\hat{\mathrm{A}}\mathrm{H}^{(k-1)}\mathrm{W}^{(k)}\right) \\ &\hat{\mathrm{A}} = \tilde{\mathrm{D}}^{-\frac{1}{2}}(\tilde{\mathrm{A}})\tilde{\mathrm{D}}^{-\frac{1}{2}}, \tilde{D}_{ii} = \sum_i \tilde{A}_{ij}, \tilde{\mathrm{A}} = \mathrm{A} + \mathrm{I} \end{aligned} \tag{2}$$

where $\mathrm{H}^{(k)}$ is the representation of all nodes in the k-th layer, σ is the activation function, and $\mathrm{W}^{(k)}$ is the parameter of the k-th layer. The matrix $\hat{A}$ obtained through the normalization technique is commonly referred to as the normalized adjacency matrix. It is computed by applying a specific normalization method to the original adjacency matrix of the graph, as shown in Eq. 2. This normalization process enhances the representation of the graph's connectivity patterns and facilitates effective computations in GNNs.

2.4.2 GraphSAGE (SAGE)

The training of GCN network requires the adjacency matrix of the entire graph, which is dependent on the structure of the graph. Moreover, GCNs can generally only be used in Transductive Learning. To address this issue, GraphSAGE (SAmple and aggreGatE) (Hamilton et al. 2017) uses a multi-layer aggregation function for training, where each layer aggregates the information of nodes and their neighbors to obtain the feature vector of the next layer. GraphSAGE utilizes the neighborhood information of nodes and does not rely on the global graph structure. The node feature aggregation of GraphSAGE can be expressed as:

$$\begin{aligned} \mathrm{h}_{\mathcal{N}(v)}^{k+1} &= \mathrm{AGG}_{k+1}\left(\left\{\mathrm{h}_u^k, \forall u \in \mathcal{N}(v)\right\}\right) \\ \mathrm{h}_v^{k+1} &= \sigma\left(\mathrm{W}^{k+1} \cdot \left[\mathrm{h}_v^k \| \mathrm{h}_{\mathcal{N}(v)}^{k+1}\right]\right) \end{aligned} \tag{3}$$

2.4.3 Graph attention networks (GAT)

GAT (Velickovic et al. 2018) is a more efficient variant of the GCN network, which addresses some of the challenges of GCN by introducing masked self-attention (Vaswani et al. 2017). The self-attention mechanism of GAT allows for assigning different weights to different nodes by attending to their neighbors, without incurring any costly matrix operations or relying on prior knowledge of the graph structure. The node representation can be formulated as follows:

$$\begin{aligned} \mathrm{h}_v^{k+1} &= \sigma\left(\sum_{u \in \mathcal{N}(v)} \alpha_{vu} \mathrm{W}\, \mathrm{h}_u^k\right) \\ \alpha_{vu} &= \frac{\exp\left(\mathrm{LeakyReLU}\left(\mathrm{a}^T\left[\mathrm{W}\,\mathrm{h}_v \| \mathrm{W}\,\mathrm{h}_u\right]\right)\right)}{\sum_{l \in \mathcal{N}(v)} \exp\left(\mathrm{LeakyReLU}\left(\mathrm{a}^T\left[\mathrm{W}\,\mathrm{h}_v \| \mathrm{W}\,\mathrm{h}_l\right]\right)\right)} \end{aligned} \tag{4}$$

where a represents the weight vector of a single-layer feedforward neural network, || denotes concatenation operator.

2.4.4 Graph isomorphism network (GIN)

GNN and its variants have achieved state-of-the-art results on node-level, link-level, and graph-level tasks. However, some previous models like GCN and GraphSAGE cannot learn to distinguish certain simple graph structures, such as graph isomorphism. To address this limitation, GIN (Xu et al. 2019) makes improvements in the neighborhood aggregation and graph readout functions, achieving a GNN as powerful as the Weisfeiler–Lehman (WL) graph isomorphism test (Weisfeiler and Leman 1968). GIN is capable of distinguishing graph structures and has a strong characterization ability to fit training data almost perfectly. The neighborhood aggregation and graph readout functions of GIN can be expressed as:

$$\begin{aligned} \mathrm{h}_v^{(k)} &= \mathrm{MLP}^{(k)}\left(\left(1+\epsilon^{(k)}\right)\cdot \mathrm{h}_v^{(k-1)} + \sum_{u\in\mathcal{N}(v)} \mathrm{h}_u^{(k-1)}\right) \\ \mathrm{h}_V &= \mathrm{CONCAT}\left(\mathrm{READOUT}\left(\left\{\mathrm{h}_v^{(k)} \mid v\in \mathrm{V}\right\}\right) \mid k=0,1,\dots,K\right) \end{aligned} \tag{5}$$

where MLP, which stands for Multilayer Perceptron (Rosenblatt 1958). The MLP is a widely employed model in the field of neural networks.

2.4.5 Simplifying graph convolutional (SGC)

The trend of complexity in traditional machine learning methods typically follows a trajectory from simple to complex. Graph Convolutional Networks (GCNs) are no exception and also exhibit this same complexity variation. However, GCNs may have unnecessary complexity and redundant computations. To address this issue, one approach is to iteratively eliminate the nonlinearity between GCN layers and collapse the resulting function into a linear variation. Specifically, SGC (Wu et al. 2019) is equivalent to removing the nonlinear activation processing step of a normal GCN, resulting in linear propagation for each layer. In other words, it is a pre-processing step for the final logistic regression, without passing through the nonlinear activation function. SGC can be expressed as a logistic regression, given by:

$$f(\mathrm{X},\mathrm{A}) = \mathrm{softmax}\left(\hat{\mathrm{A}}^K \mathrm{X}\mathrm{W}^K\right), \hat{\mathrm{A}}^K = \hat{\mathrm{A}}\hat{\mathrm{A}}\dots\hat{\mathrm{A}}, \mathrm{W}^K = \mathrm{W}^{(1)}\mathrm{W}^{(2)}\dots\mathrm{W}^{(K)} \tag{6}$$

where, in order to simplify the notation, we combine the repeated multiplication with the normalized adjacency matrix $\hat{A}$ into a single matrix operation by raising $\hat{A}$ to the power of K, denoted as $\hat{A}^K$. Furthermore, we consolidate the weights into a single matrix W^K obtained by multiplying the individual weight matrices $W^{(1)}, W^{(2)}, \dots, W^{(K)}$ together.

2.4.6 Robust graph convolutional networks (RGCN)

GCNs are known to be sensitive to perturbations, meaning that even small changes to the input can greatly affect their performance. To address this issue, the RobustGCN (RGCN) (Zhu et al. 2019) was proposed to enhance the robustness of GCNs. The key innovation of this method is the use of Gaussian distributions to represent the hidden layer feature vectors of nodes. When graph data is modified, such as changes to node features or topology, it affects the variance of the node Gaussian distribution. An attention mechanism (Vaswani et al. 2017) is also introduced to control the weight of the nodes. RGCN has been shown to be effective in defending against adversarial attacks in experiments. Its main idea can be expressed formally as follows:

$$\begin{aligned} \mathrm{M}^{(k+1)} &= \sigma\left(\tilde{\mathrm{D}}^{-\frac{1}{2}}\tilde{\mathrm{A}}\tilde{\mathrm{D}}^{-\frac{1}{2}}\left(\mathrm{M}^{(k)}\odot \mathcal{A}^{(k)}\right)\mathrm{W}_\mu^{(k)}\right) \\ \Sigma^{(k+1)} &= \sigma\left(\tilde{\mathrm{D}}^{-1}\tilde{\mathrm{A}}\tilde{\mathrm{D}}^{-1}\left(\Sigma^{(l)}\odot \mathcal{A}^{(k)}\odot \mathcal{A}^{(k)}\right)\mathrm{W}_\sigma^{(k)}\right) \end{aligned} \tag{7}$$

where $\mathrm{M}^{(k)}$ and $\Sigma^{(k)}$ are the mean and variance of the normal distribution in the k-th layer, respectively, and $\odot$ denotes the element-wise product. The attention mechanism

$\mathcal{A}^{(k)} = \exp\left(-\gamma\Sigma^{(k)}\right)$ is used to assign appropriate weights to each node, where γ is a hyperparameter controlling the strength of the attention.

3 The links between security and privacy in GNNs

3.1 The security of graph neural networks

Security in artificial intelligence (AI) is primarily concerned with safeguarding AI systems from unauthorized manipulation or corruption of data and models. An AI model's security is assessed by its ability to function correctly without interference. Specifically, an AI model is deemed secure if its training and prediction data remain undisturbed, and it can accurately predict outcomes based on the data.

GNNs, like all machine learning models, are vulnerable to adversarial attacks. GNNs take graph data as input to accomplish their node-, link-, and graph-level tasks. However, compared to image and text data, graph data contains node attribute features and dependencies between nodes, which opens up many new possibilities for attackers to breach the model. For example, an attacker might either change the nodes' features or the graph's topology, creating two different types of adversarial attacks. In addition, depending on the specificity of the graph data and the properties of the GNN model, perturbing a single node may affect the outcome of tasks that involve other related nodes.

Adversarial perturbation can either be performed on the nodes' features or the graph's structure. It is also possible to inject new nodes into a graph in a graph injection attack and perturb those nodes. Attackers can also generate adversarial examples by perturbing clean examples in the same way as is done in the text and image domains. These adversarial examples can be generated in gradient-based and non-gradient scenarios. Additionally, depending on the level of knowledge possessed by the attacker, the attacker can create adversarial examples with varying degrees of threat. These different levels of knowledge include white-box attacks, practical black-box attacks, and restricted black-box attacks.

The attacker generates adversarial examples in the scenarios mentioned above to launch security attacks on the GNN. These attacks can either be performed during the model's training stage or its inference stage.

In the training stage, the most common adversarial attacks include poisoning attacks and backdoor attacks. Poisoning attacks affect a model's performance by poisoning the training examples of the model, thus degrading its performance with a clean test set. Backdoor attacks embed triggers into the training data. When the model encounters a trigger example, the model produces a specific output.

In the inference phase, the most common adversarial attack is the evasion attack, where the adversary obtains an adversarial example by attacking the inference instance. When using a well-trained model, the accuracy of inference based on adversarial examples decreases, indicating that the evasion attack modifies the inferred example to evade the predictions of the model. It is worth mentioning that privacy attacks can also occur during the inference phase of the model, and privacy theft can be performed on the model that has been deployed for active inference.

Alongside the research on adversarial attacks, there is a great body of research on effective defenses against these attacks. These can also be classified by whether they take effect at the training stage or the inference stage.

In the training stage, the message passing method and the model's aggregation function can be changed to make the model more robust, thus improving defenses against adversarial attacks. In this stream of research, novel and more robust GNN variants, such as robust GCN (RGCN (Zhu et al. 2019)) have been proposed. In addition, in the same way as image and text data improve the robustness of the model, adversarial examples can also be added to the model's training set to defend against adversarial attacks.

In the inference phase of a GNN, it is possible to examine the inference data and, in so doing, determine whether the example is clean or an adversarial example. In addition, distillation learning is another possible defense. Some adversarial attack methods are rendered ineffective by distilling a trained model into a simple network. It is also possible to see whether some node or graph data is stable via a certificate. That is, perturbing the graph data will not affect prediction tasks with that data given an adversarial attack. Lastly, some researchers have investigated the stability of the model. In stable models, a small amount of data fine-tuning has little effect on the predictions.

Table 3 lists the main security attack and defense methods.

3.2 Privacy of graph neural networks

Privacy in AI pertains to safeguarding sensitive personal information, data, and models from unauthorized access, use, or disclosure by AI systems. Adversaries typically exploit model extraction attacks (Tramèr et al. 2016) to deduce the parameters and functions of the model. Data theft can be classified into membership inference attacks (Shokri et al. 2017) and attribute inversion attacks (Fredrikson et al. 2014). Membership inference attacks discern whether a given piece of data belongs to the training set, while attribute inversion attacks deduce specific information about the data.

GNNs are prone to privacy issues. A trained GNN model can contain a great deal of private information, such as the model's parameters and its training data, which may contain sensitive personal information. In addition to the privacy of the model, and data features similar to those in the image and text domains, GNNs also contain such unique information as link information indicating the connections between nodes.

The main privacy attack methods on GNN are model extraction, membership inference, and attribute inversion attacks.

In a model extraction attack, the user sends data cyclically through a provided interface and views the results. The results are used to infer the parameters and structure of the GNN in reverse.

Membership inference attacks, for our purposes, only apply to node- and graph-level tasks. Node-level membership inference attack mainly infer whether or not a node belongs to the training set. In a graph-level membership inference attack, in addition to guessing whether a graph exists in the training set, inferences can also be made about whether a subgraph is contained in the whole graph.

In attribute inversion attack, the attacker obtains node, link, and whole graph attribute information through the attack.

Most of the current research on the defense of privacy in GNNs has been conducted based on differential privacy. Differential privacy can also be categorized into node-level, link-level, and graph-level, depending on the privacy protection. Table 4 lists the main privacy attack and defense methods.

Table 3 Security attack and defense methods

	Stage	Method	Literature
Adversarial Attack	Training	Poisoning Attack	Zügner and Günnemann (2019), Bojchevski and Günnemann (2019), Sun et al. (2020), Zhang et al. (2022b), Zügner et al. (2018), Geisler et al. (2021)
		Backdoor Attack	Xi et al. (2021), Xu and Picek (2021), Zhao et al. (2021), Zhang et al. (2021b)
	Inference	Evasion Attack	Dai et al. (2018), Wang and Gong (2019), Wu et al. (2019), Xu et al. (2019), Bose et al. (2019), Chang et al. (2020), Ma et al. (2020), Mu et al. (2021), Zang et al. (2021), Zou et al. (2021), Ma et al. (2019), Wan et al. (2021), Chen et al. (2022), Zügner et al. (2018), Geisler et al. (2021)
Adversarial Defense	Training	Defense in Message Passing	Shanthamallu et al. (2021), Feng et al. (2021), Zhang et al. (2020), Liu et al. (2021), Zhang et al. (2022a), Zhang et al. (2022c)
		Defense in Aggregate Functions	Chen et al. (2021), Geisler et al. (2020), Geisler et al. (2021)
		Defense in Adversarial Training	Dai et al. (2018), Xu et al. (2019), Zügner and Günnemann (2019), Bojchevski et al. (2019), Jin et al. (2020), Feng et al. (2021)
	Inference	Defense by Detection	Mu et al. (2021), Wu et al. (2019), Xi et al. (2021)
		Defense by Distillation	Shanthamallu et al. (2021)
		Defense by Certificates	Schuchardt et al. (2021), Zügner and Günnemann (2019), Zügner and Günnemann (2020), Wang et al. (2021), Bojchevski et al. (2019), Jin et al. (2020)
		Defense against Fine-tuning	Zhao et al. (2021), Xu and Picek (2021)

3.3 The links between security and privacy

In the field of artificial intelligence (AI), security and privacy are distinct but closely related concepts. Security focuses on safeguarding AI models from malicious attacks that may manipulate or undermine the performance of the model. Examples of such attacks include poisoning attacks, backdoor attacks, and evasion attacks. Conversely, privacy aims to protect the sensitive information of individuals used to train AI models. Model extraction attacks, membership inference attacks, and attribute inversion attacks are common forms of privacy attacks.

It is important to note that security and privacy are closely intertwined concepts, and they are intricately linked in the study of artificial intelligence. Adversarial attacks on security can be used to perform privacy attacks. For example, Adversarial attacks on security have the potential to be leveraged for privacy attacks as well. For instance, Song et al. (2019) investigated the privacy concerns associated with adversarial attacks and demonstrated how adversarial samples, generated through data perturbation, can facilitate more effective membership inference attacks, consequently leading to increased leakage of sensitive membership information. Additionally, studies have demonstrated that the robust model achieved through adversarial training can significantly enhance the success rate of membership inference attacks (Liu et al. 2022). Furthermore, adversarial attacks can also be utilized to defend against privacy attacks. For instance, many researchers use backdoor attacks on graphs to verify the ownership of GNN models, i.e., to determine if GNN models have been stolen and modified (Zhao et al. 2021; Xu and Picek 2021). Marchant et al. studied the impact of adversarial attacks on privacy protection. In a separate study, Marchant et al. (2022) conducted a study investigating the impact of adversarial attacks on privacy defense strategies.

Similarly, privacy attacks can enhance the strength of adversarial attacks. The amount of information obtained in privacy attacks is closely linked to the classification of adversarial attacks, which includes white-box attacks, practical black-box attacks, and restricted black-box attacks (Chang et al. 2020). For instance, in the restricted black-box environment, if an attacker steals the GNN model and the training graph dataset through privacy attacks, the attacker's attack environment is equivalent to the white-box environment. In this scenario, the attacker can execute more harmful security attacks on the model.

In summary, security and privacy in GNNs are closely related but distinct concepts. Enhancing research in both security and privacy is crucial to ensuring trustworthy and reliable GNNs models.

3.4 A taxonomy of privacy and security

This survey systematically summarizes the state-of-the-art works on GNN security and privacy in recent years. Four areas are covered: adversarial attacks and defenses on GNNs and privacy attacks and defenses on GNNs. Figure 2 presents this breakdown as a taxonomy.

Adversarial attack on GNNs To perform an adversarial attack, some background knowledge is required in addition to knowing the tasks of GNN, such as the types of adversarial perturbations, the generation of adversarial perturbations, and knowledge of the adversarial attack. To perform adversarial attacks on GNNs, it is first necessary to generate adversarial examples. We have classified the adversarial perturbation methods into graph modification attacks and graph injection attacks. The generation of

Table 4 Privacy attack and defense methods

	Method	Literature
Privacy Attack	Model Extraction Attacks	Wu et al. (2022), Shen et al. (2021)
	Membership Inference Attacks	Olatunji et al. (2021), Wu et al. (2021), Duddu et al. (2020), He et al. (2021), Zhang et al. (2022d)
	Attribute Inversion Attacks	Duddu et al. (2020), Elinas et al. (2020), Zhang et al. (2021a), Shen et al. (2021), He et al. (2021), Zhang et al. (2022d)
Privacy Defense	Node-level Differential Privacy	Olatunji et al. (2021), Sajadmanesh and Gatica-Perez (2021), Du et al. (2021), Li et al. (2021), Sajadmanesh et al. (2023), Wu et al. (2021)
	Link-level Differential Privacy	Du et al. (2021), Li et al. (2021), Yang et al. (2021), Sajadmanesh et al. (2023), Hidano and Murakami (2022)
	Graph-level Differential Privacy	Du et al. (2021), Li et al. (2021), Yang et al. (2021), Mueller et al. (2022), Hidano and Murakami (2022)

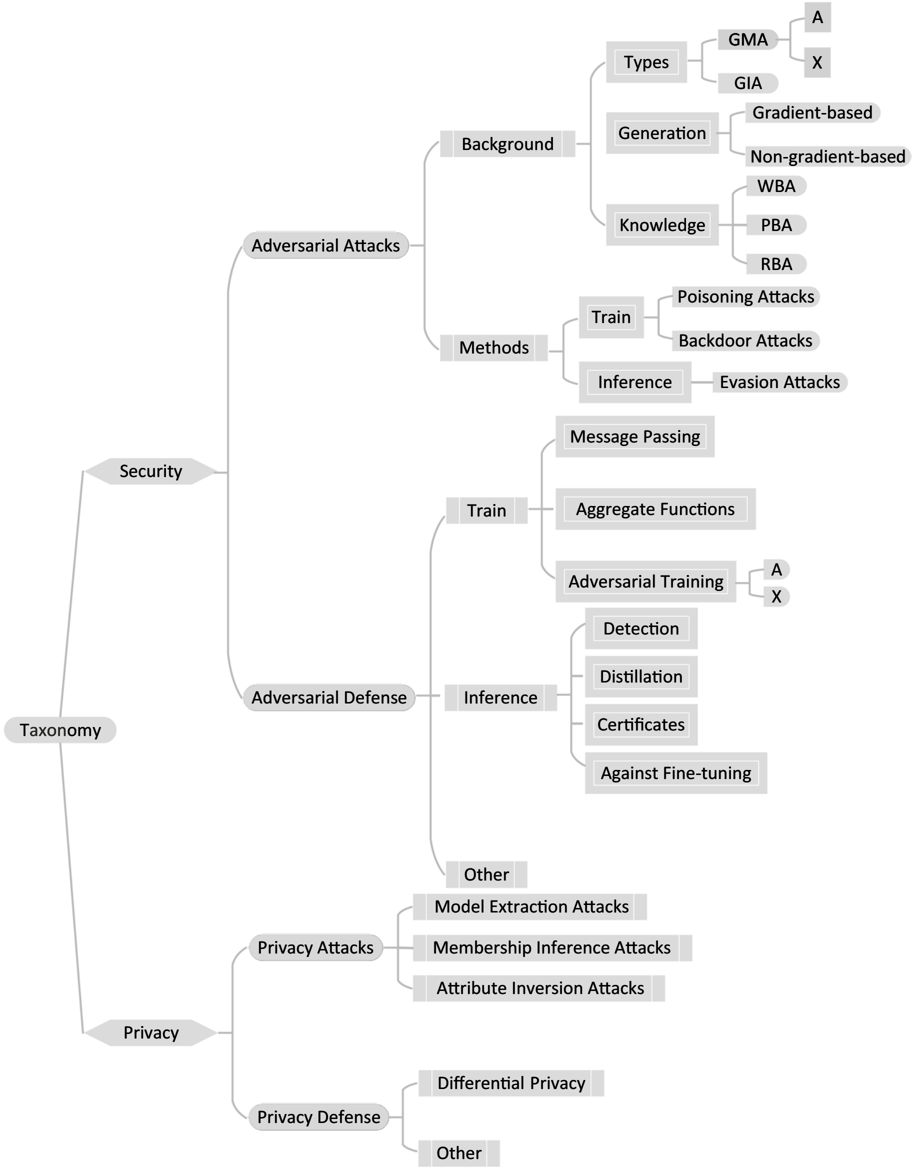

Fig. 2 Taxonomy of security and privacy on GNN

adversarial examples is mainly classified into gradient-based and non-gradient-based methods. Then, for the different levels of knowledge possessed by the attackers, the attack scenarios are classified explicitly into three scenarios, white-box attacks, practical black-box attacks, and restricted black-box attacks. Finally, the adversarial attack methods are specifically classified into poisoning attacks, backdoor attacks, and evasion attacks according to the stage in which the attack is performed.

Adversarial defenses on GNNs Studies on GNN adversarial defenses is also divided into the two stages where the defense might be implemented: the training stage and the inference stage. Many studies have sought to increase a model's robustness in the training stage by modifying the messaging method and the aggregation function. Thus, our taxonomy for model defenses follows this categorization: changing the messaging strategy, changing the aggregation function, and building a new model by changing the GNN's components (including messaging method, aggregation function, and other methods). In addition, adversarial training is also an effective defense in images and other domains, and many researchers have improved a model's robustness through adversarial training. In the model inference stage, detection and distillation are standard methods to defend against adversarial attacks. Model robustness certificates and defenses against fine-tuning are classified as defenses that operate in the validation stages. A robustness certificate proves that the nodes or graphs are robust, and their predictions do not deviate from expectations even given perturbation. Defenses against fine-tuning prevent a model from corrupting its predictions by fine-tuning the model with little data.

Privacy attack on GNNs The work on graph privacy is still in its infancy. Hence, these attacks are simply classified into model extraction attacks (Tramèr et al. 2016), membership inference attacks (Shokri et al. 2017), and attribute inversion attacks (Fredrikson et al. 2014). Membership inference attacks can be subdivided into node-level membership inference attacks and graph-level membership inference attacks, based on the characteristics of graph data being attacked. A node-level membership inference attack determines whether or not a node is in the training set. A graph-level membership inference attack not only determines whether the graph exists in the training set but can also infer whether subgraphs exist in a whole graph. Attribute inversion attacks are divided into inversion inference from the graph posterior and graph embedding inversion attacks according to the different starting points.

Privacy defenses on GNNs Most of the current defenses for GNN privacy are based on differential privacy (Dwork and Roth 2014). Hence, the privacy defenses have been divided into differentially-private approaches and other approaches. These are further classified into node-level, link-level, and graph-level protection based on the information being protected.

4 Security for graph neural networks

4.1 Adversarial attacks on graph neural networks

This section provides a discussion on adversarial attacks on GNNs. Firstly, we introduce the background knowledge of adversarial attacks, including the types of adversarial perturbations, the methods of generating adversarial examples, and the settings in which security attacks occur. Subsequently, we categorize and describe the most common adversarial attacks on security, and provide relevant literature according to the context.

4.1.1 Background of adversarial attack

Types of Adversarial Perturbations Most methods of generating adversarial examples for graphs involve adding, deleting, and modifying graphs. Zügner et al. (2018) define an attacker's changes to the clean graph as *perturbations*. Importantly, when generating adversarial examples of graphs, the disturbances must be *imperceptible*, and the process must be completed within a specific budget. We have subdivided graph perturbation into graph modification and injection attacks, as shown in Table 5.

Graph Modification Attack (GMA) Modifying a graph may involve modifying node attributes and adding or deleting edges but not modifying the number of nodes, as that would be classified as a graph injection attack. Given a graph G = (A, X), perturbing that graph with a graph modification attack might take the form:

$$\begin{aligned} &\min_{\mathrm{G}'} |\{\mathcal{M}(\mathrm{G}')_i = y_i, i \in \mathrm{V_t}\}| \\ &\text{s.t. } \mathrm{G}' = (\mathrm{A}', \mathrm{X}'), f_{\Delta_\mathrm{A}}(\mathrm{A}', \mathrm{A}) + f_{\Delta_\mathrm{X}}(\mathrm{X}', \mathrm{X}) \le \Delta \end{aligned} \tag{8}$$

where G′, A′, and X′ represent the modified G, A, and X of the graph data, respectively. The prediction model is represented by $\mathcal{M}$, and $\mathrm{V_t}$ represents the set of testing nodes. Predefined functions f_{Δ_A} and f_{Δ_F} are used to quantify the cost associated with the modifications made. Δ indicates the budget, with the formulation ensuring that any modifications to the graph's structure or properties are completed within this budget.

Specific graph modification attacks can be subdivided into structure-preserving perturbations and attribute-preserving perturbations.

- *Structure-preserving perturbation* In essence, modifying the properties of a graph's structure, such as changing the degree distribution or a node's centrality, involves changing the graph's edges, that is, *A*. Further, adversarial examples are generated by changing the original examples within a specific budget to ensure the perturbations are undetectable. Formally, this is expressed as:

$$f_{\Delta_\mathrm{A}}(\mathrm{A}', \mathrm{A}) \le \Delta \tag{9}$$

- *Attribute-preserving perturbation* Conversely, modifying the properties of a graph's nodes involves perturbing X, that is, the feature vectors of the nodes. Again, to ensure the changes are imperceptible, they must be completed within a budget. The formal expression is:

$$f_{\Delta_\mathrm{X}}(\mathrm{X}', \mathrm{X}) \le \Delta \tag{10}$$

Graph Injection Attack (GIA) There have been many studies on graph injection attacks. In these attacks, graphs are perturbed by adding nodes instead of perturbing the existing

Table 5 Types of adversarial perturbations

	Type	Perturbations	Literature
GMA	Modification	A	Dai et al. (2018), Wang and Gong (2019), Wu et al. (2019), Xu et al. (2019), Chang et al. (2020), Ma et al. (2020), Mu et al. (2021), Zang et al. (2021), Ma et al. (2019), Wan et al. (2021), Zügner and Günnemann (2019), Bojchevski and Günnemann (2019), Zhang et al. (2022b), Geisler et al. (2021), Zügner et al. (2018)
		X	Wu et al. (2019), Bose et al. (2019), Ma et al. (2020), Zügner et al. (2018)
GIA	Injection	A/X	Xi et al. (2021), Xu and Picek (2021), Zhao et al. (2021), Zhang et al. (2021b), Zou et al. (2021), Chen et al. (2022), Sun et al. (2020)

nodes as with a graph modification attack. Thus, new nodes N_I are injected into G, while the original graph structure and attributes remain unchanged. The formulation of the attack is as follows:

$$\mathrm{A}' = \begin{bmatrix} \mathrm{A} & \mathrm{P}_I \\ \mathrm{P}_I^T & \mathrm{A}_I \end{bmatrix}, \mathrm{A} \in \mathbb{R}^{N \times N}, \mathrm{P}_I \in \mathbb{R}^{N \times N_I}, \mathrm{A}_I \in \mathbb{R}^{N_I \times N_I} \tag{11}$$

$$\mathrm{X}' = \begin{bmatrix} \mathrm{X} \\ \mathrm{X}_I \end{bmatrix}, \mathrm{X} \in \mathbb{R}^{N \times M}, \mathrm{X}_I \in \mathbb{R}^{N_I \times M} \tag{12}$$

where M is the dimension of the node features, I represents the injected nodes, A_I is the adjacency matrix of the injected nodes, and P_I represents the adjacency matrix of the injected nodes and the original node edges of G. X_I represents the characteristics of the injected nodes, A' represents the adjacency matrix, and X' denotes the feature vectors after the nodes have been injected. Like graph modification attacks, graph injection attacks also need to be completed within a specific budget, formulated as:

$$\begin{aligned} &\min_{\mathrm{G}'} |\{\mathcal{M}(\mathrm{G}')_i = y_i, i \in \mathrm{V_t}\}| \\ &\text{s.t. } \mathrm{G}' = (\mathrm{A}', \mathrm{X}'), N_I \leq b, \deg(v)_{v \in I} \leq d, \|\mathrm{X}_I\| \leq \Delta_X \end{aligned} \tag{13}$$

where b, d and Δ_X are all positive constants, and the number of injected nodes N_I is less than the budget b. The degree of the injected nodes must less than the budget d, and the norm of the feature vector of the injected nodes must be less than Δ_X. These constraints guarantee that a graph injection attack will be imperceptible.

Generation of Adversarial Perturbations Most adversarial attacks involve modifying a graph's data, but each attacker has their own way of doing this. Hence, next, we divide the modification methods into two types, as shown in Table 6: data perturbation based on model gradient information and data perturbation based on non-gradient information.

Gradient-based Perturbation For the most part, gradient-based attacks are the easiest and most effective method of attack. Gradient-based perturbation refers to modifying the graph data using the model's gradient information. The gradient information contains the essential features of the data, which means the attacker can modify the adversarial examples according to each feature's importance. Any generated adversarial examples that are based on these modifications will make for inaccurate model predictions. Li et al. (2023) delved into the underlying factors contributing to the disruptive nature of gradient-based methods. They offered a comprehensive elucidation of the efficacy of gradient-based attack methods, offering insights from the vantage point of data distribution. This examination was undertaken in the context of both poisoning and evasion attacks, providing a comprehensive analysis of their effectiveness.

Non-gradient-based Perturbation In addition to gradient-based perturbation, attackers can destroy a model without using any gradient information. As we know, many researchers have turned to reinforcement learning to attack the model through long-term rewards (Ma et al. 2019; Dai et al. 2018). Others have used backdoor attacks, generating adversarial perturbations via random graphs (Zhao et al. 2021; Xu and Picek 2021). Still others have explored ways to construct adversarial examples with generative models (Bose et al. 2019; Chen et al. 2021). All the above approaches are attacks on the model that do not require any gradient information but still impact model performance quite effectively.

Table 6 Generation of adversarial perturbations

	Literature
Gradient-based	Xi et al. (2021), Dai et al. (2018), Wang and Gong (2019), Wu et al. (2019), Xu et al. (2019), Mu et al. (2021), Zang et al. (2021), Zou et al. (2021), Wan et al. (2021), Chen et al. (2022), Zügner and Günnemann (2019), Zhang et al. (2022b), Zügner et al. (2018) and Geisler et al. (2021)
Non-gradient-based	Xu and Picek (2021), Zhao et al. (2021), Zhang et al. (2021b), Dai et al. (2018), Bose et al. (2019), Chang et al. (2020), Ma et al. (2020), Ma et al. (2019), Wan et al. (2021), Bojchevski and Günnemann (2019) and Sun et al. (2020)

Knowledge of Adversarial Attacks Generating adversarial examples requires information about the target model and the target dataset. Depending on the knowledge available to the attacker, the threat level of the adversarial attack can be classified into three categories: white-box attacks, practical black-box attacks, and restricted black-box attacks, as shown in Table 7.

White-box Attacks (WBA) In this attack, the attacker has access to all information about the model, i.e., the training data (e.g., the adjacency matrix and the feature matrix), the labels, the model's parameters, its predictions, and so on. This attack is one of the most threatening attacks on a model's security. As such, it is also one of the most studied attacks. However, because white-box attacks only work when the attacker has full access to a model, they are not common in the real-world.

Practical Black-box Attacks (PBA) In these cases, the attacker has access to the labels and the predictions of the target classifier but no access to the model's parameters. In practical terms, these attacks are generally more harmful and common than white-box attacks because the attacker needs less knowledge.

Table 7 Knowledge of adversarial attacks

	Labels	Parameters	Predictions	Literature
WBA	✓	✓	✓	Xi et al. (2021), Zhang et al. (2021b), Dai et al. (2018), Wang and Gong (2019), Wu et al. (2019), Xu et al. (2019), Bose et al. (2019), Zang et al. (2021), Bojchevski and Günnemann (2019), Sun et al. (2020), Zhang et al. (2022b), Zügner et al. (2018) and Geisler et al. (2021)
PBA	✓	×	✓	Dai et al. (2018), Ma et al. (2019) and Zügner and Günnemann (2019)
RBA	×	×	×	Xu and Picek (2021), Zhao et al. (2021), Dai et al. (2018), Wang and Gong (2019), Chang et al. (2020), Ma et al. (2020), Mu et al. (2021), Zou et al. (2021), Wan et al. (2021) and Chen et al. (2022)

Restricted Black-box Attacks (RBA) Here, attackers can only access some features of the training dataset while having limited knowledge of the rest of the model. For example, attackers cannot access model parameters, labels, and predictions. This method is considered more practical and realistic than the other two attacks. However, it is also the most difficult and harmful as in reality, it is challenging for the attacker to acquire knowledge of the target data and model.

4.1.2 Methods of adversarial attacks

The two main stages in the life of a model are training and inference and, as mentioned, different types of attacks are perpetrated in each of these stages. Table 8 summarizes recent studies on adversarial attacks by stage, type, the method of generation, and the level of knowledge required to undertake the attack.

Adversarial attacks in the training stage The attacker affects the model's training by perturbing the training dataset, which means that the trained model performs worse than a model trained with a clean training dataset. The two main types of attacks in the training stage are poisoning attacks and backdoor attacks

- *Poisoning Attacks* The purpose of a poisoning attack is to interfere with the model's learning so that the model's output deviates from its expected performance. In a poisoning attack, adversarial examples are added to the training dataset, causing it to degrade on a clean test set.

  Studies have been undertaken on poisoning attacks that target both node-level tasks (Bojchevski and Günnemann 2019) and link-level tasks (Zhang et al. 2022b). Bojchevski and Günnemann (2019), for example, studied node embeddings, proposing a principled adversarial attack against an unsupervised node embedding strategy. They used eigenvalue ingestion theory (Stewart 1990) to solve a bi-layer optimization problem for poisoning attacks. By perturbing the structure of the poisoning model, they negatively affected the quality of the node embeddings so as to impair downstream tasks like node classification and link classification. Zhang et al. (2022b) proposed a poisoning attack based on unsupervised gradients that yields comparable performance to attacks on some supervised learning tasks, such as node classification and link classification. Specifically, they computed the gradient of the adjacency matrix of two views and flipped the edges with the rising gradient to maximize the contrast loss. This approach takes full advantage of the multiple views generated by graph contrast learning models and selects the most informative edges without labels, thus supporting the adaptation of the model to various downstream tasks.

  Zügner and Günnemann (2019) studied poisoning attacks within the constraints of a practical black-box attack. Here, they solved a challenging bi-level problem with poisoning attacks using meta-gradients. Essentially, they used a graph as a hyperparameter to perform the optimization, reversing the gradient-based optimization process of the model to obtain better adversarial examples. The method also yields good attack results in a restricted black-box attack setting where there is no access to the target classifier.

  Sun et al. (2020) proposed a new framework called NIPA based on a graph injection attack. NIPA uses a Markov decision process (Wei et al. 2017) to model the critical steps of a node injection attack. In addition, a new reinforcement method modifies the labels and links of the injected nodes without changing the nodes' connectivity.

Table 8 Comparison of adversarial attacks on Graph Neural Networks

Lit.	Task	Type	Generation	Knowledge	Attack	Per.
Dai et al. (2018)	Node-level/Graph-level	GMA	Non-gradient-based/Gradient-based	WBA/PBA/RBA	Evasion Attack	A
Zügner et al. (2018)	Node-level	GMA	Gradient-based	WBA	Poisoning Attack/Evasion Attack	A/X
Wang and Gong (2019)	Node-level/Graph-level	GMA	Gradient-based	WBA/RBA	Evasion Attack	A
Wu et al. (2019)	Node-level	GMA	Gradient-based	WBA	Evasion Attack	A/X
Xu et al. (2019)	Node-level	GMA	Gradient-based	WBA	Evasion Attack	A
Bose et al. (2019)	Node-level	GMA	Non-gradient-based	WBA	Evasion Attack	X
Zügner and Günnemann (2019)	Node-level	GMA	Gradient-based	PBA	Poisoning Attack	A
Bojchevski and Günnemann (2019)	Node-level/Link-level	GMA	Non-gradient-based	WBA	Poisoning Attack	A
Chang et al. (2020)	Node-level	GMA	Non-gradient-based	RBA	Evasion Attack	A
Ma et al. (2020)	Node-level	GMA	Non-gradient-based	RBA	Evasion Attack	A/X
Sun et al. (2020)	Node-level	GIA	Non-gradient-based	WBA	Poisoning Attack	A/X
Xi et al. (2021)	Node-level/Graph-level	GIA	Gradient-based	WBA	Backdoor Attack	A/X
Xu and Picek (2021)	Node-level/Graph-level	GIA	Non-gradient-based	RBA	Backdoor Attack	A/X
Zhao et al. (2021)	Node-level	GIA	Non-gradient-based	RBA	Backdoor Attack	A/X
Zhang et al. (2021b)	Graph-level	GIA	Non-gradient-based	WBA	Backdoor Attack	A/X
Mu et al. (2021)	Graph-level	GMA	Gradient-based	RBA	Evasion Attack	A
Zang et al. (2021)	Node-level	GMA	Gradient-based	WBA	Evasion Attack	A
Zou et al. (2021)	Node-level	GIA	Gradient-based	RBA	Evasion Attack	A/X
Ma et al. (2019)	Graph-level	GMA	Non-gradient-based	PBA	Evasion Attack	A
Wan et al. (2021)	Graph-level	GMA	Non-gradient-based/Gradient-based	RBA	Evasion Attack	A
Geisler et al. (2021)	Node-level	GMA	Gradient-based	WBA	Poisoning Attack/Evasion Attack	A
Chen et al. (2022)	Node-level	GIA	Gradient-based	RBA	Evasion Attack	A/X
Zhang et al. (2022b)	Node-level/Link-level	GMA	Gradient-based	WBA	Poisoning Attack	A
Li et al. (2023)	Node-level	GMA	Gradient-based	WBA	Poisoning Attack/Evasion Attack	A
Xu et al. (2022)	Graph-level	GIA	Gradient-based	WBA	Backdoor Attack	A/X

Some adversarial attack methods are not only applicable to poisoning attacks but also achieve good results in evasion attacks (Zügner et al. 2018; Geisler et al. 2021). For instance, Zügner et al. (2018) published the very first study on adversarial attacks for GNNs. They focused on poisoning attacks, but their methods also achieve good results with evasion attacks. Adversarial perturbations against nodes and graph structures are generated which take instances and dependencies into account. Geisler et al. (2021) investigated the security of large-scale GNNs, solving an attack problem with surrogate loss.

- *Backdoor Attacks* A backdoor attack involves implanting a hidden trigger in a neural network during the training process that produces a specific output when the model encounters the trigger. Such triggers are not easy to detect because, in response to other queries, the model's output is no different from usual. The trigger can be implanted in either the data or the model.

  Some approaches add a watermark to the graph data (Zhao et al. 2021; Xu and Picek 2021). This is often used as a trigger to protect the copyright of the model, as ownership of the remote suspect model can be verified with the watermarked data. In Zhao et al. (2021), an Erdos-Renyi (ER) random graph (Gilbert 1959) with random features and labels was used as a trigger and added to the graph data for training. The trained model correctly predicts clean data and correctly classifies the randomly generated ER graph to judge the attribution of the model. However, only classification tasks at the node-level were studied; tasks at the graph-level were ignored.

  Xu and Picek (2021) also studied backdoor attacks based on ER graphs. These researchers designed two strategies for generating watermarked data for graph-level tasks, plus a data generation mechanism for node-level classification tasks.

  Zhang et al. (2021b) not only accomplished graph-level backdoor attacks through ER generation triggers, but they also tried generating triggers via both Small World (Watts and Strogatz 1998) and Preferential Attachment (Barabási and Albert 1999). They injected subgraphs into some training graphs and changed their labels to target labels chosen by the attacker. Specifically, they used the subgraphs as triggers to perform different backdoor attacks by controlling the trigger size, trigger density, trigger synthesis method, and poisoning strength.

  Xi et al. (2021) proposed a new approach to graph backdoor attacks that uses subgraphs containing topological structures as well as descriptive node and edge features as triggers. However, instead of providing a fixed trigger for all graphs as in Zhao et al. (2021), Xu and Picek (2021) and Zhang et al. (2021b), dynamic triggers are generated based on the features of the target graph. This optimizes the attack's effectiveness.

  In prior instances of backdoor attacks, even a relatively straightforward filtering procedure could identify samples containing backdoor triggers as anomalous. To address this limitation, Xu et al. (2022) introduced a novel approach to clean-label backdoor attacks in the context of GNNs. In this method, the adversary solely contaminates the input corresponding to the target class, all the while leaving the true label unchanged. Consequently, this strategy allows the attack to circumvent detection mechanisms and execute a more potent backdoor assault.

Adversarial attacks in the inference stage The attacker affects the inference of the model by perturbing the inference dataset. As a result, the model makes incorrect predictions based on these data. The most common type of attack to occur in the inference stage is the evasion attack.

- *Evasion Attacks* The evasion attack is a type of attack in which an attacker generates specific input examples to deceive the target machine learning system without modifying it. Here, the model is trained on a clean training dataset, and adversarial examples are added to the inference set, resulting in wrong predictions by the model. Several research teams have crafted evasion attacks based on generating adversarial examples through a gradient approach (Wu et al. 2019; Xu et al. 2019; Zang et al. 2021). Wu et al. (2019) proposed an adversarial attack on the graph's data based on the graph's unique topology information and the discrete features of the graph data, while Xu et al. (2019) simplified the difficulty of processing discrete graph data, proposing a new gradient-based attack method. These two approaches not only perpetrate attacks on predefined and retrainable GNNs, they also require less perturbation to do so than previous adversarial attacks. Zang et al. (2021) proposed a generalized adversarial attack method called GUA that identifies bad actors. GUA finds the anchor nodes of the dataset via gradient descent and corrupts the model by flipping a small number of these anchor nodes. White-box attacks require full access to the model and labels so as to construct an adversarial loss, which is typically unrealistic in the real world. Hence, most researchers focus on black-box attacks, and more specifically restricted black-box attacks (Dai et al. 2018; Wang and Gong 2019; Chang et al. 2020; Ma et al. 2020). For example, Dai et al. (2018) proposed an attack method based on reinforcement learning that learns generalizable attack strategies. The only feedback required is from the predicted labels of the target classifier, from which the attacker can learn to how to modify the graph's structure. In addition, these researchers proposed an attack method based on a genetic algorithm and gradient descent for when prediction confidence is high or the model's gradient is available. Wang and Gong (2019) presented the first systematic study of evasion attacks for the collective classification of graphs. They framed the attack as an optimization problem, proposing techniques to approximate the problem's solution. Ma et al. (2020) extended a common gradient-based white-box attack to a black-box setting by connecting the model's gradient with PageRank-like importance scores. Further, to improve attack performance, they put forward a greedy approach that corrects the importance score. The approach considers diminishing returns as well as the difference between the loss and the misclassification rate. Chang et al. (2020) studied graph evasion attacks in restricted black-box settings. They studied the relationship between graph signal processing and a graph embedding model. An attack was constructed using a graph filter and a GF-Attack feature matrix. This GF-Attack does not require knowledge of the graph embedding model and can attacks the graph filter given black-box access. Graph-level evasion attacks are studied in Ma et al. (2019), Mu et al. (2021) and Wan et al. (2021). Ma et al. (2019), for example, investigated graph-level escape attacks in a practical black-box setting. They proposed a graph rewiring operation for the attack. As part of a graph modification attack, this special perturbation approach affects the model with a more imperceptible perturbation than a general graph modification attack, which simply adds or removes edges. The rewiring operation preserves some basic properties of the graph data - in particular, the number of nodes and edges. Then, a specially-designed strategy called ReWatt performs the rewiring operation via reinforcement learning. Wan et al. (2021) and Mu et al. (2021) studied graph-level escape attacks in a restricted black-box setting. In Wan et al. (2021), the attack is formulated as an optimization problem with the objective of minimizing the number of edges to be perturbed in a graph while maintaining the high attack success rate. In addition, a coarse-grained search algorithm and a query-efficient gradient computation algorithm reduce the number of queries on the target GNN. Mu et al. (2021)

outline a novel attack method for classification models based on Bayesian optimization. The method is query-efficient so only a few queries are required to perform the attack. They also investigated the relationship between the vulnerability of graphs as machine learning models and the topological properties of perturbed graphs, which is an important step forward for graph-interpretable adversarial attacks. Graph injection-based evasion attacks are covered in Zou et al. (2021) and Chen et al. (2022). Here, Zou et al. (2021) analyzed the topological vulnerabilities of GNNs in graph injection attacks, proposing a topological defect graph injection attack that performs injection attacks quite effectively. First, a method called TDGIA introduces a topology-defective edge selection strategy that selects the original nodes to be connected to the injected nodes. Then, a smooth feature optimization performs feature generation for the injected nodes. This type of attack is highly flexible and proves to be more effective than most graph modification attacks. However, this flexibility can also cause serious damage to graph data with a homogeneous distribution, making the attack fairly easy to defend against. Chen et al. (2022) introduced a new constraint to complement the imperceptibility constraint, termed homogeneous imperceptibility. This new constraint forces a graph injection attack to retain the homogeneity of the data. These authors also introduced a novel objective function called Harmonious Adversarial Objective (HAO) that instantiates imperceptibility given homogeneity. With this approach, it is difficult for homogeneous defenders to identify adversarial examples while maintaining an effective attack. Notably, most graph modification attack methods change either the graphs adjacency matrix or the node features and the adjacency matrix. Lastly, Bose et al. (2019) proposed an evasion attack that only perturbs a graph's node features. Interestingly, this is a method that can be used on graphs, images, and text.

4.2 Defenses against adversarial attacks

Defenses can also be divided into those implemented in the training stage and those implemented in the inference stage. Defenses implemented in the training stage improve a model's robustness by modifying the model's structure and the training data. Defenses implemented in the inference stage protect a fully-trained model. Table 9 summarizes the literature on adversarial defenses discussed in this survey.

4.2.1 Training stage defenses

The message passing defense Graph neural networks have a unique ability to pass messages between nodes. That is, the features of nodes are propagated to neighboring nodes through the topology of the graph. In this way, GNNs can extract features from the whole of the graph by overlaying multiple message passing layers. The final information obtained via aggregation then includes feature information from neighboring nodes and some of the graph's structural information. GNNs have delivered excellent performance on many benchmark datasets using this messaging framework. However, real-world applications often contain abnormal graph data. For instance, in social networks, new users may not have fully completed their profiles, which can cause some tasks to fail. Worst of all, attackers can change a node attribute's characteristics to maliciously manipulate the predictions made by the model. Therefore, GNNs must be designed in a way that is robust to abnormal data.

Table 9 Comparison of defenses against adversarial attacks

Lit.	Task	Stage	Method	Code
Dai et al. (2018)	Node-level/Graph-level	Train	Adversarial Training	✓
Wu et al. (2019)	Node-level	Train/Inference	Detection	×
Xu et al. (2019)	Node-level	Train	Adversarial Training	✓
Zügner and Günnemann (2019)	Node-level	Train/Inference	Adversarial Training/Certificates	✓
Zhu et al. (2019)	Node-level	Train	Message Passing	✓
Bojchevski et al. (2019)	Node-level	Train/Inference	Adversarial Training/Certificates	✓
Elinas et al. (2020)	Node-level	Train	Message Passing	✓
Zügner and Günnemann (2020)	Node-level	Inference	Certificates	✓
Zhang et al. (2020)	Node-level	Train	Message Passing	✓
Jin et al. (2020)	Node-level	Train/Inference	Adversarial Training/Certificates	✓
Geisler et al. (2020)	Node-level	Train	Aggregate Functions	✓
Entezari et al. (2020)	Node-level	Train/Inference	Other/Low-Rank	×
Shanthamallu et al. (2021)	Node-level	Train/Inference	Message Passing/Distillation	×
Feng et al. (2021)	Node-level	Train	Message Passing	×
Mu et al. (2021)	Graph-level	Inference	Detection/Other-Low-Rank	✓
Schuchardt et al. (2021)	Node-level	Inference	Certificates	✓
Chen et al. (2021)	Node-level	Train	Aggregate Functions	✓
Zhao et al. (2021)	Node-level	Inference	Against Fine-tuning	×
Dai et al. (2021)	Node-level/Link-level	Train	Message Passing	✓
Wang et al. (2021)	Node-level/Graph-level	Inference	Certificates	✓
Liu et al. (2021)	Node-level	Train	Message Passing	✓
Geisler et al. (2021)	Node-level	Train	Aggregate Functions	✓
Xi et al. (2021)	Node-level/Graph-level	Inference	Detection	×
Feng et al. (2021)	Node-level	Train	Adversarial Training	✓
Xu and Picek (2021)	Node-level/Graph-level	Inference	Against Fine-tuning	×
Zhang et al. (2022a)	Node-level/Graph-level	Train	Message Passing	×
Zhuang and Hasan (2022)	Node-level	Train	Message Passing	✓
Zhang et al. (2022c)	Node-level	Train	Message Passing	✓
Zhang et al. (2021c)	Node-level	Inference	Detection	×
Wang et al. (2023)	Node-level	Train	Message Passing	✓
Wu et al. (2023)	Node-level	Inference	Distillation	✓
Tian et al. (2023)	Node-level	Inference	Distillation	✓

Looking to improve the robustness of models by changing the way models pass messages, Zhang et al. (2020) mitigated the adverse effects of adversarial attacks by modifying a GNN's neural message passing operator. More specifically, the message passing architecture was changed such that the modified model became robust to adversarial perturbations while still maintaining its ability to learn representations. Although residual connections in GNN message passing can help to improve performance, they also significantly amplify the vulnerability of GNNs to abnormal node features. To address this problem, Liu et al. (2021) derived a simple, efficient, interpretable, and adaptive message passing scheme.

Shanthamallu et al. (2021) looked to improve the robustness of GNN models by exploiting epistemic uncertainty in a message passing framework. The framework constructs a surrogate predictor that does not have direct access to the graph structure. Then reliable knowledge is systematically extracted from the GNN through a novel uncertainty matching strategy. Most importantly, this uncoupling means the GNN is significantly more robust to poisoning attacks by design and is completely immune to evasion attacks.

In most GNN networks, and particularly social networks, the proximity of the network and the fine local structures in the data are not accurately captured through node similarity. In addition, the node similarity metric can lead to non-optimal models with potential distribution bias. To address this problem, and to improve GNN robustness, Zhang et al. (2022a) proposed a new message passing mechanism. The approach learns the proximity of the local structure by collecting embedding sets that describe the nodes and their neighbors, i.e., subgraphs around the nodes of interest. In addition, the Wasserstein distance is calculated with the help of a differentiable optimization method, making the whole network trainable end-to-end.

Zhang et al. (2022c) investigated the robustness of heterogeneous graph neural networks (HGNNs) (Wang et al. 2019; Yun et al. 2019; Fu et al. 2020), proposing a robust HGNN framework called RoHE. In their study, they analyzed two key reasons for the vulnerability of HGNNs to attacks - one being the perturbation amplification effect, and the other being soft attention mechanisms. RoHE defends adversarial attacks by changing the message passing method and configuring an attention purifier. More specifically, RoHE introduces a meta path-based transfer probability as an a priori criterion for the cleaner, which reduces the confidence levels of malicious neighbors in a hostile center. Then, the purifier masks the neighbors with the lowest confidence via a learning scheme. This approach effectively mitigates the negative impact of malicious neighbors in soft attention mechanisms.

In a quest to mitigate the impact of undesirable perturbations, Wang et al. (2023) embark on an analysis of the intrinsic connectivity property, leading to the conception of the intrinsic connectivity graph. Furthermore, they discern the significance of the adjacency matrix rank in revealing a graph that yields embeddings identical to those of the intrinsic connectivity graph. To capture such a graph, the authors incorporate structural entropy into the objective function, thereby influencing GNN message passing. They also tackle the challenges posed by graph randomness and endeavor to learn precise node representations in the absence of label information.

Aggregate functions as defenses In the message passing step, GNNs update node embeddings by aggregating the embeddings of neighboring nodes. The aggregation function is a core part of GNNs, and is the reason why GNNs are able to support irregular graph data. Hence, graph-specific perturbations are highly effective at degrading the performance of GNNs, while traditional defenses seem unable to improve robustness. Many researchers have therefore focused on aggregation functions as a way to increase robustness in GNNs.

GNNs typically have stringent requirements for aggregation functions. Moreover, if the aggregation functions are sensitive to slight perturbations, then the entire model is often not robust. A couple of studies describe the effect of aggregation functions on GNNs in the face of structural attacks by introducing the theory of the breakdown point (Chen et al. 2021; Geisler et al. 2020). Chen et al. (2021) analysis takes advantage of the breakdown point, which quantitatively measures the robustness of aggregation schemes. They propose new aggregation functions - trimmed mean and median aggregation - with high breakdown points that strengthen a model's defense against adversarial attacks. Their experiments show that employing a robust aggregation function can result in good model robustness without sacrificing prediction accuracy. Inspired by the field of robust statistics, Geisler et al. (2020) proposed a robust aggregate function called soft medoid. A soft medoid shows that the maximum possible collapse point is 0.5, which means that, as long as fewer than 50% of the nodes are perturbed, the deviation of the aggregation will be bounded. However, in this paper, an adversary's vulnerability to soft medoid is only studied in the context of small graphs. GNNs are becoming increasingly important due to their popularity and the diversity of applications they are proving to be useful in. Thus, to make their approach more usable, Geisler et al. (2021) also studied how to protect large-scale GNNs with a new aggregation function called soft median.

Adversarial training as a defense Adversarial training is a meaningful way to enhance the robustness of neural networks. During the network training process, the adversarial examples obtained from the perturbation are added to the training set so the neural network can adapt to the perturbation, making the model robust.

In summary, adversarial training is a dynamic regularization technique that defends against the worst perturbations of input features. Many researchers have also studied adversarial training with GNNs to improve the robustness of models.

- *Structure perturbations* In 2018, researchers began studying adversarial training for graphs. Dai et al. (2018), for example, tried to increase the robustness of the model with an inexpensive adversarial training method. During training, the method randomly deletes the graph's edges and the drop. The experimental results show that the attack rate of various methods decreases by about 1%, while the accuracy of the target model remains unchanged. Although the attack success rate drops by only 1%, the adversarial training is having some effect, underlining the validity of this research. On this basis, researchers have continued to study the adversarial training of graphs based on structural perturbation. With the help of the proposed first-order attack generation method, Xu et al. (2019) offers an adversarial training method for GNNs, which improves the robustness of the model. The approach is robust to different gradient-based and greedy attack methods, but it does not sacrifice the original classification accuracy. Adversarial training has also been studied in parallel with the robust model certificate studies in Bojchevski et al. (2019) and Jin et al. (2020). When adversarial training is performed, the number of certified robust nodes increases. At the same time, the prediction accuracy of the clean data is not affected.
- *Attribute perturbations* In addition to the research on structural perturbations, many scholars have also conducted research on adversarial training based on attribute perturbation. Giving consideration to binary node attribute perturbations, Zügner and Günnemann (2019) proposed a robust semi-supervised training procedure. They improved the robustness of the GNN by jointly processing labeled and unlabeled nodes with little effect on the original prediction accuracy. Feng et al. (2021) designed a new novel GNN optimization method called graph adversarial training. The graph adversarial regularizer proposed in this paper can enhance the robustness of the model to the perturbation of node attribute, forcing the GNN model to learn to prevent the perturbation from propagating on the graph.

4.2.2 Inference stage defenses

Defense by Detection Detection is a standard defense method in the security field. Contaminated data is identified through detection, assuming that the data has been contaminated. Then, the contaminated data is removed or cleaned to reduce the attack's impact. Because of the particularity of graph data, in addition to graph feature detection, the topology of the graph can also be used to distinguish between clean and abnormal data.

Mu et al. (2021) outlines an adversarial graph detector. With clean and adversarial graphs as training sets, they trained a binary GNN classifier. The role of the classifier is to distinguish adversarial graphs from clean graphs. Using the classifier as a detector reveals whether the graph suffers from adversarial perturbations.

Wu et al. (2019) studied perturbations to GCNs caused by existing attack techniques, fining that, in existing attack methods, attackers connect edges of nodes with very different characteristics, and this connection method plays a key role in all attack methods. Hence, they subsequently proposed a new defense method that detects and recovers potential adversarial perturbations. The method is based on preprocessing, where a statistical analysis is performed on the node attributes and the similarity of nodes is calculated by introducing a similarity measure. Then, all edges connecting the nodes with low similarity scores are selected as candidates for deletion. The experimental results show that, even if these edges are removed from the clean graph, there is no harm to node-level predictions. In fact, it can even improve the model's predictions under certain circumstances.

Xi et al. (2021) extended NeuralCleanse (Wang et al. 2019) and RandomizedSmoothing (Zhang et al. 2021b) for detection defense. Given a suspicious GNN, they check for backdoors at the model inspection stage. Specifically, each class is searched for potential backdoors. If a class has a backdoor embedded, the minimal perturbation required to change all inputs in this class to the target class is abnormally smaller compared to other classes.

Zhang et al. (2021c) proposed a simple single node threshold test for detecting nodes that are subjected to targeted attacks. They also presented a kernel-based two-sample test to identify whether a given subset of nodes in the graph is maliciously compromised. Furthermore, they demonstrated the potential practical advantages of the proposed detection method as a mechanism to shield graph-based models from security threats in Bitcoin transaction data analysis.

Defense by Distillation The term 'distillation' as it pertains to model security was originally coined by Hinton et al. (2015). The core idea of distillation learning is to use the knowledge learned by a large model to guide the training of a small model. In this way, the performance of the small model is comparable to that of the large model. Still, the parameters are greatly reduced, thus reducing the model's size and speeding up its operation. More specifically, a complex network model is trained first, and then a smaller network is trained using the output of this complex network and the true labels of the data. Papernot and McDaniel (2017); Papernot et al. (2016) have also recently studied defense mechanisms through distillation.

In a GNN model, the graph attributes usually input into a model include a feature matrix and an adjacency matrix. Shanthamallu et al. (2021) proposed a defense method using distillation learning to defend against poisoning attacks. Using a novel uncertainty matching strategy, they jointly train a standard GNN model and a surrogate predictor that only inputs node features. By guiding the training of the surrogate predictor from the GNN model, the surrogate predictor significantly improves the GNN's robustness to adversarial attacks.

Wu et al. (2023) and Tian et al. (2023) delved into the exploration of model robustness by implementing a process of distilling knowledge from teacher GNNs into student MLP models. Wu et al. methodically dissected the knowledge acquired by GNNs into two components: low-frequency and high-frequency elements residing in both spectral and spatial domains. Their study further comprehensively examined the respective roles these components play in the GNN-MLP distillation process. In parallel, Tian et al. tackled three pivotal challenges within the GNN-MLP framework: the incongruity between content feature and label spaces, the stringent hard matching requirement to the teacher's outputs, and the susceptibility to noise within node features. They also presented a consolidated perspective on learning MLP that integrates effectiveness, robustness, and efficiency aspects.

Defense by Certificates Adversarial attacks aim to highlight the potential vulnerabilities of GNNs. Recent studies have shown that GNNs are highly non-robust given adversarial attacks on graph structures and node attributes, which makes their results unreliable. A successful attack will only provide non-robust results. However, an attack's failure does not necessarily imply that a GNN model is robust, and it does not guarantee that the method used is reliable. To be able to use GNNs safely, we need provable robustness principles.

Günnemann's team researched the perturbation of node attributes and the structure of graphs. In Zügner and Günnemann (2019), they propose a provably robust method against node attribute perturbations. They consider perturbations of node attributes under challenging L_0 perturbation budgets and deal with discrete data domains. A node is robust if it has been certified using this method. That is to say, in the case of various disturbances, the prediction result of this node is correct.

Robustness certificates against graph structure perturbations are investigated in Bojchevski et al. (2019); Zügner and Günnemann (2020). In Zügner and Günnemann (2019), robustness certificates are studied by linking PageRank and Markov decision processes. However, these certificates are seldom studied with GCNs. Zügner and Günnemann (2020) fills this gap by demonstrating robustness certificates for GCNs given structural perturbations. Schuchardt et al. (2021) proposes the first collective robustness certificate, which counts the number of predictions guaranteed to remain stable under perturbations, i.e., the number of predictions that cannot be attacked.

Notably, the robustness certificates above are all node-level. As yet, there are no robustness certificates at the graph-level that involve topological perturbations of local and global budgets. Jin et al. (2020), however, proposed a robustness certificate that operates at the graph-level based on Lagrangian dualization and convex envelopes. Given a well-trained GCN and a threat model with local and global budgets, this verification method effectively proves that any topological perturbation cannot alter graph predictions.

Wang et al. (2021) studied robustness certificates for graphs with perturbations like added and deleted edges. They demonstrate that any GNN has proven robustness certificates for both node-level and graph-level classification tasks. They extended a recently developed technique called randomized smoothing (Cao and Gong 2017; Cohen et al. 2019; Jia et al. 2020; Lécuyer et al. 2019; Li et al. 2019; Liu et al. 2018) to graph data. Randomized smoothing can transform any basic classifier into a robust classifier by adding random noise to the test examples. Experiments show that when an attacker arbitrarily adds and deletes edges to/from a graph, using a random smooth GCN can yield high certification accuracy.

Defense against fine-tuning Beyond training a neural network from scratch, fine-tuning is another way to obtain a new model. Many researchers have utilized fine-tuning as a means of attacking a deployed model. Fine-tuning has several advantages. First, training a model from scratch requires powerful computing resources and large datasets, whereas fine-tuning uses an existing model to continue training on the target task dataset (Yosinski et al. 2014). Hence, an attacker will likely use a model to train a new model from a stolen model with only a small amount of training data required.

Xu and Picek (2021) investigated whether the watermark generation method is robust to fine-tuning. They fine-tuned a watermarked GNN model using half of the test data. The other half of the test data was evaluated to see if the watermarks previously embedded in the GNN model remain valid in the new model. The experimental results show that both of the watermark generation methods they proposed are robust to fine-tuning, and the accuracy of one of the generation methods is not affected by fine-tuning

Zhao et al. (2021) investigated whether the size of a watermark affects fine-tuning. They used 30% of the training and test sets to fine-tune the model and evaluated the effect of fine-tuning on watermarking. What they found was that when a watermark is small - for example, when only ten nodes are used in the trigger - fine-tuning the model has little effect on the watermark's extraction.

4.2.3 Other methods

In addition to defending the model during the training and inference stages, some researched defense methods work outside these two stages.

Entezari et al. (2020) conducted defense research based on Nettack (Zügner et al. 2018). The main idea is that only high-rank or low-valued distinct components of the adjacency matrix of a graph are affected by adversarial attacks. Since these low-valued distinct components contain little graph structure information, they can be discarded to reduce the impact caused by adversarial attacks. The higher-order perturbations generated by Nettack can be discarded using lower-order approximations of the adjacency matrix and the eigenmatrix. At the same time, they show that a rank-10 approximation of the matrices can defend against an adversary's attack with high probability, and the model will yield almost the same performance as with a clean graph. Mu et al. (2021) extended a low-rank-based defense from node classification to graph classification. A singular value decomposition (SVD) is first performed on the adjacency matrix of each test graph. Then, the largest singular value is kept and the remaining singular values are discarded. A new adjacency matrix and the corresponding graph with the perturbations removed can then be derived from the largest singular value. Experiments show that this defense achieves a clear trade-off between accuracy and robustness.

5 Privacy for graph neural networks

Confidentiality in a machine learning system means that unauthorized users cannot access information related to the model, including the training data and information about the model itself, e.g., the model's parameters, its architecture, the training methods, etc.

5.1 Privacy attacks on graph neural networks

Attacks that break the confidentiality of a GNN can be classified into three types: model extraction attacks, membership inference attacks, and attribute inversion attacks. Figure 3 illustrates a simple flowchart that outlines these privacy attacks. In this figure, the attacker starts by querying the target model, and then trains the shadow model using the responses obtained and available data. The attacker can then conduct three types of privacy attacks using the shadow model. Table 10 summarizes the recent literature on privacy attacks.

5.1.1 Model extraction attacks (MEA)

In a model extraction attack, the adversary constructs an surrogate model, also known as a shadow model, that is identical or functionally very similar to the victim model. The

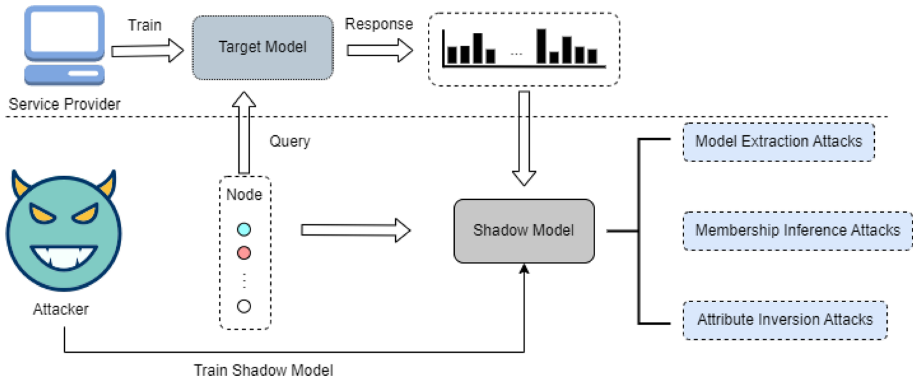

Fig. 3 A schematic diagram of the process and classification of privacy attacks

shadow model is constructed by repeatedly querying the victim model and using the results of the queries to reverse engineer the specific parameters and structure of the model. This attack method was first proposed by Tramèr et al. (2016) in 2016, who describes the method in detail using a linear regression model as an example. In this model, there are assumed to be n parameters. The attacker uses m examples, where $m > n$, to make predictions so as to obtain the corresponding predicted values. Then, a linear equation system consisting of m equations is constructed. The specific values of the n parameters are derived by solving the equations.

Notably, most previous model extraction attacks have targeted Euclidean models designed for images or text. These types of attacks on GNNs are rare and have seldom been studied. Since the special nature of graph data contains node features and topological structure information, the way the attacker categorizes knowledge will differs from data like pictures and text. Wu et al. (2022) were the first to comprehensively analyze and formulate a model extraction attack on a GNN. They divided the graph's knowledge into three dimensions: the attributes of the attack nodes, the structure of the attack nodes, and selected subgraphs (graphs with similar node attributes and topologies to the victim graph). Further, they described the model extraction attacks using seven categories according to whether the attacker possesses knowledge of the three dimensions. The experimental results show that they can effective shadow models can be trained in these seven categories using different strategies. Thus, each of the attacks successfully extracts the target model. Most of the shadow models achieve almost the same accuracy as the target with around 85% of the predictions being the same as the target.

Other studies have focused on transductive GNNs. These studies, however, assume that the attacker has knowledge of the victim model's training process, which is somewhat unrealistic in practice. Nevertheless, training and query graphs are used to train the shadow model.

Shen et al. (2021) focus on the more realistic and popular inductive GNN model, which generalizes well to invisible nodes. Attackers query the target model through a remotely accessible API, and they do not tamper with the training process of the target model. Moreover, their attack follows a generic framework that can be applied to all scenarios. The framework contains two main components. The first is used to learn the discrete graph structure if the query graph's structural information is unavailable. The second component then constructs the agent model by jointly learning the features of the nodes and the

Table 10 Comparison of privacy attacks on Graph Neural Networks

Lit.	Dataset	Model	Task	Denf.	Attacks	Code
Olatunji et al. (2021)	Flickr, Cora, CiteSeer, PubMed, Reddit	GCN, GAT, SGC, GraphSAGE	Node-level	×	MIA	×
Wu et al. (2021)	PROTEIN full, DD, ENZYMES, OGBG-PPA, CIFAR10, MNIST, NCI	GCN, GateGCN, GIN, GAT, GraphSAGE	Graph-level	×	MIA	✓
Duddu et al. (2020)	Pubmed, Citeseer, Cora, Facebook, LastFM	GCN, GAT, GraphSAGE, TAGCN, DeepWalk, Node2Vec	Node-level/Link-level/Graph-level	×	MIA/AIA	✓
Zhang et al. (2021a)	Cora, Citeseer, Polblogs, Air-traffic(USA, Brazil), AIDS, ENZYMES	GCN, GAT, GraphSAGE	Node-level/Graph-level	✓	AIA	✓
He et al. (2021)	Cora, Citeseer, Cora-full, LastFM	GraphSAGE, GAT, GIN	Node-level	✓	MIA	×
He et al. (2021)	Citeseer, Cora, Pubmed, AIDS, COX2, DHFR, ENZYMES, PROTEINS_full	GCN, GraphSAGE, GAT	Link-level	✓	AIA	×
Wu et al. (2022)	Cora, Citeseer, Pubmed	GCN	Node-level	×	MEA	✓
Shen et al. (2021)	DBLP, Pubmed, Citeseer, Coauthor, ACM, Amazon	GIN, GAT, GraphSAGE	Node-level/Graph-level	✓	MEA/AIA	✓
Zhang et al. (2022d)	DD, ENZYMES, AIDS, NCI1, OVCAR-8 H	GraphSAGE	Node-level/Graph-level	✓	AIA/MIA	✓
Li et al. (2020)	Google plus, Facebook, HepTh	GAE	Node-level/Link-level	×	AIA	×
Zhou et al. (2023)	Cora, Citeseer, Polblogs, USA, Brazil, AIDS	GCN, GraphSAGE, GAT	Node-level/Link-level	✓	AIA	✓
de Ocáriz Borde et al. (2023)	Cora, Citeseer, Pubmed, Physics, CS	GCN, GAT	Link-level	×	AIA	×
Zhou et al. (2022)	ogbl-ddi	GraphSAGE, GCN, GAT, GIN	Link-level	×	AIA	✓
Wang and Wang (2022)	Pokec, Facebook, Pubmed	GCN, GraphSAGE, GAT	Link-level	✓	AIA	✓

corresponding the target model. The article classifies the attacks into six different attacks according to the local knowledge known (e.g., the graph structure information, A_Q), the response results R (the node embedding matrix H, the predicted posterior probability matrix Θ, and the t-SNE projection matrix ϒ). Experiments show that this model stealing attack consistently achieves strong performance. In addition, the attacks remain effective even if the adversary does not know the graph structure information.

5.1.2 Membership inference attacks (MIA)

Given access to a data record and a model, membership inference attacks involve determining whether that data record belongs to the training set of the model (Shokri et al. 2017). Membership inference attacks are also a very popular research topic in the security field. This is because the ability to infer whether some specific data exists in the training set of a machine learning model poses a great security risk to users. For example, a model trained based on cancer patient information can directly impart information about a patient's disease if it is inferred that the patient is a member of the training set. This and other such examples can lead to a series of discrimination problems (Backes et al. 2016). Current research shows that most membership inference attacks focus on models trained on images and text (Leino and Fredrikson 2020; Salem et al. 2019; Shokri et al. 2017; Song and Shmatikov 2019). However, the graph data for training GNNs can contain a great deal of sensitive information, such as healthcare analytics (Errica et al. 2020; Gilmer et al. 2017), flow trajectories (Backes et al. 2017; Cho et al. 2011), and so on. For this reason, some researchers have analyzed membership inference attacks based on GNN models. We review the literature on membership inference attacks in two parts: node-level and graph-level.

Most of the success of membership inference attacks in traditional machine learning models has been attributed to model overfitting or memorization of data sets (Zhang et al. 2017). Overfitting leads to high confidence scores for the data records seen during training compared to new data, which provides an easy avenue to distinguish training from testing data. Olatunji et al. (2021) investigated whether overfitting in GNNs could also be a significant success factor for membership inference attacks. They performed such an attack on some GNNs by introducing two induction settings; then they analyzed the properties of the GNNs. They found, first, that a lack of overfitting does not guarantee robustness against membership inference attacks and, second, that the attacks were successful even when the target model was well generalized. Instead, the connectivity among the instances (unlike in tabular data) increases the vulnerability of GNN models to privacy attacks. They show that, in GNNs, the additional structural information is the major contributing factor. They support their findings with extensive experiments on four representative GNN models.

Duddu et al. (2020) investigated membership inference attacks in both the black-box and white-box settings. In the black-box setting, the attacker exploits the output prediction scores. In the white-box environment, the attacker can access the published node embedding. The black-box setting considers the specifics of downstream node classification tasks using the graph embeddings of GNNs and performs the attack both with and without auxiliary knowledge. The white-box setting involves an unsupervised attack for the more general case, which is to use only graph embeddings to distinguish whether a given node is in the training graph or not. The experimental results show that the attacker can accurately predict the training data in the above case.

Although these two methods (Olatunji et al. 2021; Duddu et al. 2020) show success, Duddu et al. (2020) lacks an explicit attack method, and Olatunji et al. (2021) performs their attacks in restricted scenarios. To address these issues, He et al. (2021) provide the first comprehensive analysis of node-level membership inference attacks on GNNs. They systematically defined the threat model and proposed three node-level membership inference attacks based on the adversary's background knowledge: the 0-hop attack, the 2-hop attack, and the combination attack. In the 0-hop attack, the attacker uses only the characteristics of the target node itself to query the target model. In the 2-hop attack, the attacker uses the features of the target node and a 2-hop subgraph to query the target model. The experimental results show that GNNs are vulnerable to node-level membership inference attacks even if the attackers have little background knowledge. Meanwhile, graph density and feature similarity have a significant impact on membership inference attacks.

In the context of GNN models, studying MIA at the graph-level is just as important as studying node-level MIA. Wu et al. (2021) took the first step towards exploring graph-level MIA in GNNs. The goal of MIA is to determine if graph examples are present in the training set. The authors proposed two types of attacks, namely learning-based and threshold-based attacks. In the former, the attacker sends a query to the target model and receives a confidence score. They comprehensively measured the effectiveness of the attacks under various experimental settings for different GNN models and training datasets. Furthermore, they analyzed the factors affecting the attack performance and explored the impact of overfitting on MIA at both the graph- and node-levels. In contrast to node-level MIA, which has little correlation with overfitting, MIA at the graph-level is different and closely related to overfitting, as mentioned in Olatunji et al. (2021).

5.1.3 Attribute inversion attacks (AIA)

Attribute inversion attacks obtain information about private data through an API provided by system. Attribute inversion attacks were first proposed by Fredrikson et al. (2014), who demonstrated the method on a linear regression model. Later, Fredrikson et al. (2015) extended this attribute inversion attack to extract the information from images of faces through a shallow neural network.

Attribute inversion attacks have been successful in fields where grid-like data, such as images, are common. However, since graph structures are quite specific and GNNs have their unique message passing capabilities, attribute inversion attacks on grid-like data are generally not directly applied to graphs. For this reason, many researchers have also studied model reversal attacks on GNNs. We discuss these model reversal attacks in two groups: posterior-based inversion attacks and embedding-based inversion attacks.

$Posterior - based\ inversion\ attack$:In a posterior-based inversion attack, information about the training data is reverse engineered from the final output of the model (Fredrikson et al. 2015). Zhang et al. (2021a) proposed an attribute inversion attack method that works when the adversary has a trained GNN model and some auxiliary knowledge (e.g., node labels and attributes). With this, the attackers can reconstruct all the edges between the nodes in the training set. Specifically, they designed two important modules: a projective gradient module, and a graph self-encoder module. The former addresses the discrete nature of the graph edges while maintaining the sparsity and smoothness of the graph features. The latter takes information such as the node attributes, the graph's topology, and the target model parameters into account during graph reconstruction. Using this method, one can investigate the relationship between a model reversal attacks and edge influence. The experimental results show that edges with greater influence are more likely to be recovered. Moreover, they demonstrated that the method is effective on several state-of-the-art GNNs.

Notably, Shen et al. (2021) designed a similar component approach to learn the discrete graph structure if the structural information is not available in the query graph. Zhou et al. (2023) approached the GNN as a Markov chain and leveraged the flexible chain approximation to launch an attack on the GNN. Their investigation delved into the core concepts of graph reconstruction, wherein they systematically examined the intricacies of chain-based graph reconstruction attacks and the corresponding defense mechanisms.

In contrast to the approach of conducting a link inversion attack through the reconstruction of the entire adjacency matrix, He et al. (2021) proposed a new attack method that can predict whether a link exists between any pair of nodes in a graph and is used to train the target GNN model from the model's output. They describe three types of background knowledge the attacker may have: the node features of the target dataset, a partial graph of the target dataset, and an auxiliary dataset that also contains its graph and node features. The attacks are classified into eight types based on which types of knowledge the attacker has. Building upon this foundation, de Ocáriz Borde et al. (2023) enhance edge inference performance by incorporating Riemannian geometry into the model, resulting in a more intricate embedding space. Zhou et al. (2022) undertake edge prediction experiments across varying sizes of test graphs.

Wang and Wang (2022) conducted an extensive study on Group Property Inference Attacks (GPIA) within graph neural networks. Their research revolves around two distinct types of properties that adversaries aim to infer: node group properties, which encapsulate the collective information of specific node groups, and link group properties, which encapsulate the collective information of specific link groups. Importantly, they address both Posterior-based and Embedding-based approaches to group property inference, reflecting a comprehensive exploration of the attack landscape. To validate their attack methodology, Wang et al. conducted a series of experiments under varying settings and datasets. Their findings underscored the effectiveness of the proposed attack strategy. Moreover, the researchers proactively designed a set of defense mechanisms against GPIA attacks. Empirical results highlighted the efficiency of these countermeasures, showcasing their ability to substantially reduce attack accuracy while incurring only minor accuracy losses in the GNN model itself.

$Embedding - based\ inversion\ attack$:In addition to inferring private data information from the final output of the model, there are also some studies on graph embedding that examine information leaks. Duddu et al. (2020) studied the attribute inversion attacks through the lens of graph embeddings, finding a way to use them to launch a membership inference attack. The paper mentions two model reversal attacks: graph reconstruction and attribute inference. The attack target of the graph reconstruction attack is to reconstruct the target graph given a corresponding graph embedding. The attribute inference attack target aims to infer sensitive node attributes corresponding to a single user. In a graph reconstruction attack, the attacker has access to the node embeddings of a subgraph and trains a generative model to reconstruct the target graph from its published embeddings. This attack may reconstruct sensitive input graphs, causing severe privacy implications. Moreover, link information can be obtained from the reconstructed graph as with Zhang et al. (2021a).

In attribute inference attacks, attackers use published graph embeddings to infer sensitive information about user nodes (e.g., a user's gender and location). Inferring these model attributes is usually formulated as a supervised learning problem. With a given target embedding, a supervised attack model is trained to predict the sensitive attributes of the target user.

Zhang et al. (2022d) also studied attribute inference attacks and graph reconstruction attacks via graph embeddings - their aim being to infer the essential attributes of the target graph, such as the number of nodes, number of edges, and graph density. The attack is modeled as a multi-task classification problem, where all the attributes of interest are predicted simultaneously. As such, their method can reconstruct graphs with a similar structure and statistical information to the target graph.

Anonymizing graph data by removing identifying information and adding or removing edges is a popular strategy for privacy protection (Ji et al. 2017). Li et al. (2020) conducted a privacy attack study on de-anonymization and proposed a seed-free graph de-anonymization method that automatically extracts features and matches nodes globally without initial matching node pairs. Specifically, they used deep neural networks to learn features and an adversarial framework for node matching. The extensive experimental results show that the proposed method outperforms existing seed-free methods by a factor of one hundred in landmark identification.

5.2 Defenses against privacy attacks

Privacy attacks on GNNs pose such a significant threat that many researchers have studied how these models can be protected. Differential privacy was specifically designed to objectively quantify the privacy loss of individuals whose data are algorithmically processed (Dwork 2008; Dwork and Roth 2014). Today, differential privacy remains one of our best defenses against privacy attacks (Zhang et al. 2021a; He et al. 2021). Table 11 summarizes the literature on privacy defenses in recent years.

5.2.1 Differential privacy

Differential Privacy (*DP*) :Differential privacy uses randomization methods to make personal data unstealable while ensuring that the statistical features are accurate. Specifically, randomization is added to the data so that an attacker cannot infer private information from any differences in the query results. The formal definition of differential privacy is as follows:

$$\mathbb{P}[M(\mathrm{D}) \in S] \leq e^{\varepsilon}\mathbb{P}\left[M\left(\mathrm{D}'\right) \in S\right] + \delta \tag{14}$$

where M is a randomization algorithm, D and D′ are two different datasets, S is all events, ε is the privacy budget, and δ is a perturbation. This formula holds that a randomized algorithm acting over M on two neighboring data sets D and D′ should yield an output event S with about the same probability.

To use differentially-private algorithms for GNN data, the properties of the neighboring datasets need to be formally defined (Mueller et al. 2022). In this paper, these neighboring datasets are divided into three categories according to what is being protected: removing or adding a node and its adjacent edges (node-level DP), removing or adding an edge (edge-level DP), and removing or adding an entire graph (graph-level DP).

Olatunji et al. (2021) proposed a new protection framework for the differential privacy of graph data (PrivGNN) so as to publish GNN models with differential privacy guarantees. They assumed two graphs exist: a labeled private graph and an unlabeled public graph. PrivGNN uses distillation learning (Hinton et al. 2015), where knowledge from some teacher models trained on private graphs is transferred to some student models trained only on public graphs using differential privacy. By exploiting the teacher-student training paradigm, PrivGNN is robust to attacks on GNN models, including membership inference attacks and model stealing attacks. In addition, they derive tight privacy

Table 11 Comparison of defenses against privacy attacks

Lit.	Dataset	Model	Task	Attacks	Code
Olatunji et al. (2021)	Amazon, ArXiv, Reddit	PrivGNN (GraphSAGE)	Node-level	MIA/AIA	×
Sajadmanesh and Gatica-Perez (2021)	Cora, Pubmed, Facebook, LastFM	LPGNN(GCN, GAT, GraphSAGE)	Node-level	AIA	✓
Du et al. (2021)	LastFM, Facebook	GAE	Node-level/Link-level/Graph-level	AIA	×
Li et al. (2021)	Yale, Rochester	APGE	Node-level/Link-level/Graph-level	AIA	✓
Yang et al. (2021)	DBLP, IMDB	DPGGAN	Link-level/Graph-level	AIA	✓
Wu et al. (2021)	MovieLens, Flixster, Douban, Yahoo-Music	FedGNN(GAT)	Node-level	AIA	×
Zhou et al. (2020)	Cora, Pubmed, Citeseer	VFGNN	Node-level	AIA	×
Hu et al. (2022)	Pokec-z, Pokec-n, German credit, Recidivism, Credit defaulter	DP-GCN	Node-level	AIA	×
Mueller et al. (2022)	Synthetic, Fingerprints, Molbace, ECG	GraphSAGE, GAT, GCN	Graph-level	AIA	×
Sajadmanesh et al. (2023)	Facebook, Reddit, Amazon	GAP	Node-level/Link-level	AIA	×
Hidano and Murakami (2022)	REDDIT-BINARY, REDDIT-MULTI-5K	GIN	Link-level/Graph-level	AIA	×
Chen et al. (2022)	Cora, Citeseer, Pubmed, Physics	SAGE, GCN, GAT, GIN	Node-level	AIA	✓
Wang et al. (2023)	Cora, Citeseer, DBLP, CS, Elliptic	GraphSAGE, GIN, GAT, GATv2, SuperGAT, APPNP	Node-level	AIA	✓

guarantees using Rényi Differential Privacy (RDP) (Mironov 2017), the theoretical results of a Poisson subsampling mechanism, and the advanced combination theorem of RDP.

To ensure individual link privacy, Yang et al. (2021) formulated and enforced strict privacy constraints on deep graph generation models using a differential privacy framework that focuses on link-DP. Their framework, called differential privacy graph generation adversarial network (DPGGAN), performs differentially-private training on a graph generation model that has had its links reconstructed. Strict individual link privacy preservation is achieved and, further, a structure-oriented graph comparison for practical global graph structure preservation is ensured.

Sajadmanesh et al. (2023) proposed a novel GNN learning method called GAP that carries differential privacy guarantees based on a study of aggregation perturbation. Aggregation perturbation, in which a Gaussian mechanism is applied to the output of the GNN aggregation function, is used as the primary technique for implementing DP in the proposed method. The neighborhood aggregation step and learnable transformation are separated into different aggregation and classification modules to avoid spending a privacy budget in each training iteration. To further reduce costs from the budget, the node features are transformed into a low-dimensional space via an encoder that does not depend on the graph adjacency matrix.

Local differential privacy is another privacy defense that has been studied (Kasiviswanathan et al. 2011; Sajadmanesh and Gatica-Perez 2021; Du et al. 2021; Wu et al. 2021). For example, Sajadmanesh and Gatica-Perez (2021) investigated node-level privacy, proposing a GNN learning framework (Drop) that preserves privacy independent of the model's architecture. The framework is based on local differential privacy with provable privacy guarantees.

Wu et al. (2021) investigated local differential privacy for recommendation tasks. Existing GNN-based recommendation methods rely on the centralized storage of user-item graphs and centralized model learning. However, user privacy is privacy-sensitive, and centralized storage of user-item graphs may cause privacy issues and risks. Based on this, they propose a novel privacy-preserving recommendation-based federal framework for a GNN (FedGNN) that can collectively use highly decentralized user data to train GNN models.

Studies have also been undertaken on privacy protection at the link level based on local differential privacy. Hidano and Murakami (2022), for instance, proposed a new local differential privacy algorithm called degree preserving random response (DPRR). The method outperforms Warner's RR (Warner 1965) without destroying the graph structure and neighborhood aggregation for non-private users.

Differential privacy at the graph level is still in its infancy with Mueller et al. (2022) being the first to demonstrate the application of differentially-private GNNs to graph-level tasks. Their work essentially extends the application of differential private stochastic gradient descent (DP-SGD) (Abadi et al. 2016) to graph-level classification tasks.

5.2.2 Other privacy defense methods

In addition to differential privacy, there are other methods of protecting GNNs against privacy attacks. In this section, we introduce the non-differentially private methods (Shen et al. 2021; He et al. 2021; Zhang et al. 2022d; He et al. 2021) also mentioned some privacy defenses (Shen et al. 2021).

Shen et al. (2021) mentions adding random Gaussian noise to the node embeddings and t-SNE projections returned by the target model to defend against the privacy attacks they developed. He et al. (2021) proposes two defense mechanisms to mitigate the risk of privacy leaks with GNNs: random edge addition and label-only output. Zhang et al. (2022d) defends against problems with privacy leaks by adding a moderate level of noise to the embeddings in the target graph while still maintaining the performance of normal tasks. He et al. (2021) defends against attacks by limiting the number of GNN models out of the maximum number of posteriors.

Specialized work on privacy defenses has also been undertaken. For example, in the real-world, some data owners segregate information into their own private database, which is not accessible to other users. Protecting data segregated this way is known as 'the segregation problem', and Zhou et al. (2020) devised a federated learning strategy specifically to address it. Called VFGNN, the scheme revolves around vertically partitioning the datasets. Computing the graph is divided into two parts - a private part and a non-private part. The private part is reserved for the data owner, and the corresponding calculations are given to a semi-honest server. First, the data owner uses secure multi-party computation techniques to generate the local node embeddings. This is done via feature extraction on the private data. Then, a global node embedding is generated by combining the local node embeddings from different data holders through different combination strategies. Lastly, the server returns the final hidden layer to the party that owns the labels to compute the prediction and loss. The data holder and the server perform forward and backward propagation to complete the model training and prediction.

Existing GNN studies on privacy protection assume that all the users' sensitive attributes are known in advance. In real applications, this is not reasonable because different users have different privacy preferences. For example, male users are much less sensitive to age in social networks than female users (Hu et al. 2022). To address this problem, Hu et al. (2022) proposed a novel privacy-preserving GNN model called DP-GCN that protects sensitive information in GCNs. DP-GCN consists of two modules: a deconvolved representation learning (DRL) module and a node classification (NCL) module. The DRL decomposes a user's non-sensitive attributes into sensitive and non-sensitive representations orthogonal to each other in a potential space. The NCL trains the GCN to classify unlabeled nodes in the graph with insensitive potential representations. The aim is that these insensitive representations can be used to perform downstream tasks. Experimental results show that the proposed model has good privacy preserving capability and competitive performance at node classification.

Machine unlearning (Bourtoule et al. 2021) is a novel privacy defense method that achieves privacy protection by forgetting private data. One of the most advanced solutions in this area is SISA. It randomly divides the training set into multiple fragments and trains a composition model for each fragment. However, directly applying SISA to graph data may lead to significant corruption of the graph's structural information, thus reducing the utility of the resulting ML model. To address this issue, Chen et al. (2022) propose a combination of the SISA approach with GNN, named GraphEraser. They propose two new graph partitioning algorithms and a learning-based aggregation method for graph unlearning and conduct extensive experiments on five real-world datasets to demonstrate the forgetting efficiency and model practicality of GraphEraser. However, it's important to note that GraphEraser has been explicitly tailored for the transductive setting. In contrast, Wang et al. (2023) introduced the GUIDE framework, focusing on the unlearning process within the inductive setting. GUIDE is structured around three fundamental components: guided graph partitioning emphasizing fairness and balance, efficient subgraph repair, and similarity-based aggregation. This approach demonstrates its efficacy in the context of unlearning for inductive scenarios.

6 Future directions

While GNN technology has achieved good results with node-level, link-level, and graph-level tasks, it also raises many security and privacy concerns. Additionally, GNNs still have shortcomings and improvements are needed in terms of their security and privacy. These present opportunities for future research, as discussed next.

6.1 The future of security for GNNs

To enhance GNN models' security, it is imperative to strengthen current research on GNN security. Along with developing better methods to defend against adversarial attacks, it is crucial to shift the focus towards adversarial defense. For example, most of the current research into adversarial training is based on perturbing a graph's node features - in which case, adversarial training works very well. Yet interestingly, adversarial training does not improve a model's robustness particularly well when the graph's structure is perturbed, and structural perturbations are very common. Hence, ways to enhance the robustness of a model to structural perturbations through adversarial training is a very important and urgent problem to be researched. In addition, robust optimization (Madry et al. 2018) and adversarial dropout (Park et al. 2018) are suitable methods of improving the robustness of deep models, but how well they can be integrated into GNN models remains to be studied.

6.2 The future of privacy for GNNs

Research into privacy on GNNs is still in its infancy. In fact, most of the work summarized in Table 10 has only been conducted in the last two years. Notably, model extraction attacks and graph-level membership inference attacks have received relatively little attention despite their critical implications for GNN privacy. More research can and should be done on these two types of privacy attacks in the future.

In the area of GNN privacy defense, most current defense methods rely on differential privacy. However, there is an interesting new defense method called Machine Unlearning (Bourtoule et al. 2021) that has gained attention for its ability to perform data forgetting in trained models for privacy protection purposes. Although it has only been studied in GNN by Chen et al. (2022), we believe that it is a novel and meaningful direction for GNN privacy protection. As it is mentioned in the Sect. 5.2.2 of this paper, it has not been classified separately due to the scarcity of articles. Nonetheless, the potential of Machine Unlearning to improve GNN privacy should not be overlooked.

6.3 The future of security and privacy for GNNs

Considering the connection between privacy and security, we believe that joint research on these topics will become a trend in the future. Privacy attacks can provide knowledge about models and data properties that can be used to undermine security. Current research on counterattacks typically involves experiments based on white-box attacks, practical black-box attacks, and restricted black-box attacks. These studies primarily focus on improving the sample of adversarial attacks in their respective environments. However, conducting research in conjunction with privacy attacks can convert practical and restricted black-box environments into white-box environments for attacks, which could pose a greater risk to the security of the model.

In addition, attackers can corrupt the performance of the model by generating adversarial examples. To make the attack imperceptible, the properties of the adversarial example should be as consistent as possible with the clean example. This consistency can also expose data information during the analysis of the adversarial example or can be used to perform better privacy attacks. Therefore, we believe that joint research on adversarial examples and privacy attacks is an interesting direction that deserves further exploration.

7 Conclusion

In this survey, we provided a comprehensive introduction to the latest research on security and privacy on GNNs. More specifically, we reviewed and summarized the literature on GNN adversarial attacks, adversarial defenses, privacy attacks, and privacy defenses. The survey began with a brief introduction on security and privacy on GNNs, focusing on their similarities, differences and the relationships between the two. We then classified the research work on attacks and defenses in GNN security and privacy, respectively. Finally, we provide an outlook on the future of this field and the possible research opportunities for GNN security and privacy going forward.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

References

Abadi M, Chu A, Goodfellow IJ, McMahan HB, Mironov I, Talwar K, Zhang L (2016) Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, pp 308–318

Backes M, Berrang P, Humbert M, Manoharan P (2016) Membership privacy in microrna-based studies. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, pp 319–330

Backes M, Humbert M, Pang J, Zhang Y (2017) walk2friends: inferring social links from mobility profiles. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security, CCS, pp 1943–1957

Barabási A-L, Albert R (1999) Emergence of scaling in random networks. Science 286(5439):509–512

Battaglia PW, Hamrick JB, Bapst V, Sanchez-Gonzalez A, Zambaldi VF, Malinowski M, Tacchetti A, Raposo D, Santoro A, Faulkner R, Gülçehre Ç, Song HF, Ballard AJ, Gilmer J, Dahl GE, Vaswani A, Allen KR, Nash C, Langston V, Dyer C, Heess N, Wierstra D, Kohli P, Botvinick MM, Vinyals O, Li Y, Pascanu R (2018) Relational inductive biases, deep learning, and graph networks. CoRR arXiv:1806.01261

Bojchevski A, Günnemann S (2019) Adversarial attacks on node embeddings via graph poisoning. In: Proceedings of the 36th international conference on machine learning, ICML, vol 97, pp 695–704

Bojchevski A, Günnemann S (2019) Certifiable robustness to graph perturbations. In: Advances in neural information processing systems 32: annual conference on neural information processing systems 2019, NeurIPS, pp 8317–8328

Bose AJ, Cianflone A, Hamilton WL (2019) Generalizable adversarial attacks using generative models. CoRR arXiv:1905.10864

Bourtoule L, Chandrasekaran V, Choquette-Choo C.A, Jia H, Travers A, Zhang B, Lie D, Papernot N (2021) Machine unlearning. In: 2021 IEEE symposium on security and privacy (SP). IEEE, pp 141–159

Cai Z, Xiong Z, Xu H, Wang P, Li W, Pan Y (2021) Generative adversarial networks: a survey towards private and secure applications. CoRR arXiv:2106.03785 (2021)

Cao X, Gong NZ (2017) Mitigating evasion attacks to deep neural networks via region-based classification. In: Proceedings of the 33rd annual computer security applications conference, pp 278–287

Carlini N, Wagner DA (2017) Towards evaluating the robustness of neural networks. In: 2017 IEEE symposium on security and privacy, SP, pp 39–57

Chang H, Rong Y, Xu T, Huang W, Zhang H, Cui P, Zhu W, Huang J (2020) A restricted black-box adversarial framework towards attacking graph embedding models. In: The thirty-fourth AAAI conference on artificial intelligence, AAAI, pp 3389–3396

Chen J, Chen Y, Chen L, Zhao M, Xuan Q (2021) Multiscale evolutionary perturbation attack on community detection. IEEE Trans Comput Soc Syst 8(1):62–75

Chen L, Li J, Peng Q, Liu Y, Zheng Z, Yang C (2021) Understanding structural vulnerability in graph convolutional networks. In: Proceedings of the thirtieth international joint conference on artificial intelligence, IJCAI, pp 2249–2255

Chen Y, Yang H, Zhang Y, Ma K, Liu T, Han B, Cheng J (2022) Understanding and improving graph injection attack by promoting unnoticeability. In: International conference on learning representations

Chen M, Zhang Z, Wang T, Backes M, Humbert M, Zhang Y (2022) Graph unlearning. In: Proceedings of the 2022 ACM SIGSAC conference on computer and communications security, pp 499–513

Cho E, Myers SA, Leskovec J (2011) Friendship and mobility: user movement in location-based social networks. In: Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining, pp 1082–1090

Cohen JM, Rosenfeld E, Kolter JZ (2019) Certified adversarial robustness via randomized smoothing. In: Proceedings of the 36th international conference on machine learning, ICML. Proceedings of machine learning research, vol 97, pp 1310–1320

Dai E, Aggarwal C, Wang S (2021) NRGNN: learning a label noise resistant graph neural network on sparsely and noisily labeled graphs. In: KDD ’21: the 27th ACM SIGKDD conference on knowledge discovery and data mining, virtual event, pp 227–236

Dai H, Li H, Tian T, Huang X, Wang L, Zhu J, Song L (2018) Adversarial attack on graph structured data. In: Proceedings of the 35th international conference on machine learning, ICML, vol 80, pp 1123–1132

Dai E, Zhao T, Zhu H, Xu J, Guo Z, Liu H, Tang J, Wang S (2022) A comprehensive survey on trustworthy graph neural networks: privacy, robustness, fairness, and explainability. CoRR arXiv:2204.08570 (2022)

de Ocáriz Borde HS, Kazi A, Barbero F, Liò P (2023) Latent graph inference using product manifolds. In: The eleventh international conference on learning representations, ICLR 2023, Kigali, Rwanda, May 1–5, 2023

Duddu V, Boutet A, Shejwalkar V (2020) Quantifying privacy leakage in graph embedding. In: MobiQuitous ’20: computing, networking and services, virtual event, pp 76–85

Du W, Ma X, Dong W, Zhang D, Zhang C, Sun Q (2021) Calibrating privacy budgets for locally private graph neural networks. In: 2021 international conference on networking and network applications, pp 23–29

Dwork C (2008) Differential privacy: a survey of results. In: Theory and applications of models of computation, 5th international conference, TAMC, vol 4978, pp 1–19

Dwork C, Roth A (2014) The algorithmic foundations of differential privacy. Found Trends Theor Comput Sci 9(3–4):211–407

Elinas P, Bonilla EV, Tiao LC (2020) Variational inference for graph convolutional networks in the absence of graph data and adversarial settings. In: Advances in neural information processing systems 33: annual conference on neural information processing systems 2020, NeurIPS

Entezari N, Al-Sayouri SA, Darvishzadeh A, Papalexakis EE (2020) All you need is low (rank): defending against adversarial attacks on graphs. In: WSDM ’20: the thirteenth ACM international conference on web search and data mining, pp 169–177

Errica F, Podda M, Bacciu D, Micheli A (2020) A fair comparison of graph neural networks for graph classification. In: 8th international conference on learning representations, ICLR

Fan W, Ma Y, Li Q, He Y, Zhao YE, Tang J, Yin D (2019) Graph neural networks for social recommendation. In: The world wide web conference, WWW, pp 417–426

Feng F, He X, Tang J, Chua T (2021) Graph adversarial training: dynamically regularizing based on graph structure. IEEE Trans Knowl Data Eng 33(6):2493–2504

Feng B, Wang Y, Ding Y (2021) UAG: uncertainty-aware attention graph neural network for defending adversarial attacks. In: Thirty-fifth AAAI conference on artificial intelligence, AAAI, pp 7404–7412

Fredrikson M, Jha S, Ristenpart T (2015) Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security, pp 1322–1333

Fredrikson M, Lantz E, Jha S, Lin SM, Page D, Ristenpart T (2014) Privacy in pharmacogenetics: an end-to-end case study of personalized warfarin dosing. In: Proceedings of the 23rd USENIX security symposium, pp 17–32

Fu X, Zhang J, Meng Z, King I (2020) MAGNN: metapath aggregated graph neural network for heterogeneous graph embedding. In: WWW '20: the web conference 2020, pp 2331–2341

Geisler S, Schmidt T, Şirin H, Zügner D, Bojchevski A, Günnemann S (2021) Robustness of graph neural networks at scale. Adv Neural Inf Process Syst 34

Geisler S, Zügner D, Günnemann S (2020) Reliable graph neural networks via robust aggregation. In: Advances in neural information processing systems 33: annual conference on neural information processing systems 2020, NeurIPS

Gilbert EN (1959) Random graphs. Ann Math Stat 30(4):1141–1144

Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE (2017) Neural message passing for quantum chemistry. In: Proceedings of the 34th international conference on machine learning, ICML. Proceedings of machine learning research, vol 70, pp 1263–1272

Goodfellow IJ, Shlens J, Szegedy C (2015) Explaining and harnessing adversarial examples. In: 3rd international conference on learning representations, ICLR

Günnemann S (2022) Graph neural networks: adversarial robustness. In: Graph neural networks: foundations, frontiers, and applications, pp 149–176

Hamilton WL, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. In: Advances in neural information processing systems 30: annual conference on neural information processing systems 2017, NeurIPS, pp 1024–1034

Hamilton WL, Ying R, Leskovec J (2017) Representation learning on graphs: methods and applications. IEEE Data Eng Bull 40(3):52–74

He X, Jia J, Backes M, Gong N.Z, Zhang Y (2021) Stealing links from graph neural networks. In: 30th USENIX security symposium, USENIX, pp 2669–2686

He X, Wen R, Wu Y, Backes M, Shen Y, Zhang Y (2021) Node-level membership inference attacks against graph neural networks. CoRR arXiv:2102.05429

Hidano S, Murakami T (2022) Degree-preserving randomized response for graph neural networks under local differential privacy. CoRR arXiv:2202.10209

Hinton GE, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. CoRR arXiv:1503.02531

Hu H, Cheng L, Vap J.P, Borowczak M (2022) Learning privacy-preserving graph convolutional network with partially observed sensitive attributes. In: WWW '22: the ACM web conference 2022, virtual event, pp 3552–3561

Ji S, Mittal P, Beyah RA (2017) Graph data anonymization, de-anonymization attacks, and de-anonymizability quantification: a survey. IEEE Commun Surv Tutor 19(2):1305–1326

Jia J, Cao X, Wang B, Gong NZ (2020) Certified robustness for top-k predictions against adversarial perturbations via randomized smoothing. In: 8th international conference on learning representations, ICLR

Jia J, Salem A, Backes M, Zhang Y, Gong NZ (2019) Memguard: defending against black-box membership inference attacks via adversarial examples. In: Proceedings of the 2019 ACM SIGSAC conference on computer and communications security, CCS, pp 259–274 (2019)

Jin H, Shi Z, Peruri VJSA, Zhang X (2020) Certified robustness of graph convolution networks for graph classification under topological attacks. In: Advances in neural information processing systems 33: annual conference on neural information processing systems 2020, NeurIPS

Jordan MI, Mitchell TM (2015) Machine learning: trends, perspectives, and prospects. Science 349(6245):255–260

Kasiviswanathan SP, Lee HK, Nissim K, Raskhodnikova S, Smith AD (2011) What can we learn privately? SIAM J Comput 40(3):793–826

Kayes MI, Iamnitchi A (2017) Privacy and security in online social networks: a survey. Online Soc Netw Media 3–4:1–21

Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. CoRR arXiv:1609.02907

Lécuyer M, Atlidakis V, Geambasu R, Hsu D, Jana S (2019) Certified robustness to adversarial examples with differential privacy. In: 2019 IEEE symposium on security and privacy, SP, pp 656–672

Leino K, Fredrikson M (2020) Stolen memories: Leveraging model memorization for calibrated white-box membership inference. In: 29th USENIX security symposium, USENIX, pp 1605–1622

Li K, Luo G, Ye Y, Li W, Ji S, Cai Z (2021) Adversarial privacy-preserving graph embedding against inference attack. IEEE Internet Things J 8(8):6904–6915

Li B, Chen C, Wang W, Carin L (2019) Certified adversarial robustness with additive noise. In: Advances in neural information processing systems 32: annual conference on neural information processing systems 2019, NeurIPS, pp 9459–9469

Li K, Liu Y, Ao X, He Q (2023) Revisiting graph adversarial attack and defense from a data distribution perspective. In: The eleventh international conference on learning representations, ICLR 2023, Kigali, Rwanda, May 1–5, 2023

Li K, Lu G, Luo G, Cai Z (2020) Seed-free graph de-anonymiztiation with adversarial learning. In: d'Aquin M, Dietze S, Hauff C, Curry E, Cudré-Mauroux P (eds) CIKM '20: the 29th ACM international conference on information and knowledge management, virtual event, Ireland, October 19–23, 2020, pp 745–754

Liu X, Cheng M, Zhang H, Hsieh C (2018) Towards robust neural networks via random self-ensemble. In: Computer Vision—ECCV 2018—15th European Conference, Munich. Lecture Notes in computer science, vol 11211, pp 381–397

Liu X, Ding J, Jin W, Xu H, Ma Y, Liu Z, Tang J (2021) Graph neural networks with adaptive residual. Adv Neural Inf Process Syst 34

Liu Z, Zhang X, Chen C, Lin S, Li J (2022) Membership inference attacks against robust graph neural network. In: Chen X, Shen J, Susilo W (eds) Cyberspace safety and security—14th international symposium, CSS 2022, Xi'an, China, October 16–18, 2022, Proceedings. Lecture notes in computer science, vol 13547, pp 259–273

Ma J, Ding S, Mei Q (2020) Towards more practical adversarial attacks on graph neural networks. In: Advances in neural information processing systems 33: annual conference on neural information processing systems 2020, NeurIPS

Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A (2018) Towards deep learning models resistant to adversarial attacks. In: 6th international conference on learning representations, ICLR

Marchant NG, Rubinstein BIP, Alfeld S (2022) Hard to forget: poisoning attacks on certified machine unlearning. In: Thirty-sixth AAAI conference on artificial intelligence, AAAI 2022, thirty-fourth conference on innovative applications of artificial intelligence, IAAI 2022, the twelveth symposium on educational advances in artificial intelligence, EAAI 2022 virtual event, February 22–March 1, 2022, pp 7691–7700

Ma Y, Wang S, Wu L, Tang J (2019) Attacking graph convolutional networks via rewiring. CoRR arXiv:1906.03750

Mironov I (2017) Rényi differential privacy. In: 30th IEEE computer security foundations symposium, CSF, pp 263–275

Mothukuri V, Parizi RM, Pouriyeh S, Huang Y, Dehghantanha A, Srivastava G (2021) A survey on security and privacy of federated learning. Future Gener Comput Syst 115:619–640

Mueller TT, Paetzold JC, Prabhakar C, Usynin D, Rueckert D, Kaissis G (2022) Differentially private graph classification with GNNs. CoRR arXiv:2202.02575

Mueller TT, Usynin D, Paetzold JC, Rueckert D, Kaissis G (2022) SoK: differential privacy on graph-structured data. CoRR arXiv:2203.09205 (2022)

Mu J, Wang B, Li Q, Sun K, Xu M, Liu Z (2021) A hard label black-box adversarial attack against graph neural networks. In: CCS '21: 2021 ACM SIGSAC conference on computer and communications security, virtual event, pp 108–125

Olatunji IE, Funke T, Khosla M (2021) Releasing graph neural networks with differential privacy guarantees. CoRR arXiv:2109.08907

Olatunji IE, Nejdl W, Khosla M (2021) Membership inference attack on graph neural networks. In: 3rd IEEE international conference on trust, privacy and security in intelligent systems and applications, TPS-ISA, pp 11–20

Papernot N, McDaniel PD (2017) Extending defensive distillation. CoRR arXiv:1705.05264

Papernot N, McDaniel PD, Goodfellow I.J, Jha S, Celik ZB, Swami A (2016) Practical black-box attacks against deep learning systems using adversarial examples. CoRR arXiv:1602.02697

Papernot N, McDaniel P.D, Wu X, Jha S, Swami A (2016) Distillation as a defense to adversarial perturbations against deep neural networks. In: IEEE symposium on security and privacy, SP, pp 582–597

Park S, Park J, Shin S, Moon I (2018) Adversarial dropout for supervised and semi-supervised learning. In: Proceedings of the thirty-second AAAI conference on artificial intelligence, AAAI, pp 3917–3924

Rosenblatt F (1958) The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev 65(6):386

Sajadmanesh S, Gatica-Perez D (2021) Locally private graph neural networks. In: CCS '21: 2021 ACM SIGSAC conference on computer and communications security, virtual event, pp 2130–2145

Sajadmanesh S, Shamsabadi AS, Bellet A, Gatica-Perez D (2023) GAP: differentially private graph neural networks with aggregation perturbation

Salem A, Zhang Y, Humbert M, Berrang P, Fritz M, Backes M (2019) Ml-leaks: model and data independent membership inference attacks and defenses on machine learning models. In: 26th annual network and distributed system security symposium, NDSS

Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G (2009) The graph neural network model. IEEE Trans Neural Netw 20(1):61–80

Schlichtkrull MS, Cao ND, Titov I (2021) Interpreting graph neural networks for NLP with differentiable edge masking. In: 9th international conference on learning representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021

Schuchardt J, Bojchevski A, Klicpera J, Günnemann S (2021) Collective robustness certificates: exploiting interdependence in graph neural networks. In: 9th international conference on learning representations, ICLR

Shanthamallu US, Thiagarajan JJ, Spanias A (2021) Uncertainty-matching graph neural networks to defend against poisoning attacks. In: Thirty-fifth AAAI conference on artificial intelligence, AAAI, pp 9524–9532

Shen Y, He X, Han Y, Zhang Y: Model stealing attacks against inductive graph neural networks. CoRR arXiv:2112.08331

Shi W, Rajkumar R (2020) Point-GNN: graph neural network for 3D object detection in a point cloud. In: 2020 IEEE/CVF conference on computer vision and pattern recognition, CVPR 2020, Seattle, WA, USA, June 13–19, 2020, pp 1708–1716

Shi L, Zhang Y, Cheng J, Lu H (2019) Skeleton-based action recognition with directed graph neural networks. In: IEEE conference on computer vision and pattern recognition, CVPR, pp 7912–7921

Shokri R, Stronati M, Song C, Shmatikov V (2017) Membership inference attacks against machine learning models. In: 2017 IEEE symposium on security and privacy, SP, pp 3–18

Song C, Shmatikov V (2019) Auditing data provenance in text-generation models. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, KDD, pp 196–206

Song L, Shokri R, Mittal P (2019) Privacy risks of securing machine learning models against adversarial examples. In: Cavallaro L, Kinder J, Wang X, Katz J (eds) Proceedings of the 2019 ACM SIGSAC conference on computer and communications security, CCS 2019, London, UK, November 11–15, 2019, pp 241–257

Stewart GW (1990) Matrix perturbation theory

Sun Y, Wang S, Tang X, Hsieh T, Honavar VG (2020) Adversarial attacks on graph neural networks via node injections: A hierarchical reinforcement learning approach. In: WWW '20: the web conference 2020, pp 673–683

Sun L, Wang J, Yu PS, Li B (2018) Adversarial attack and defense on graph data: a survey. CoRR arXiv:1812.10528

Tian Y, Zhang C, Guo Z, Zhang X, Chawla NV (2023) Learning MLPs on graphs: a unified view of effectiveness, robustness, and efficiency. In: The eleventh international conference on learning representations, ICLR 2023, Kigali, Rwanda, May 1–5, 2023

Tramèr F, Zhang F, Juels A, Reiter MK, Ristenpart T (2016) Stealing machine learning models via prediction apis. In 25th USENIX security symposium, USENIX, pp 601–618

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A.N, Kaiser L, Polosukhin I (2017) Attention is all you need

Velickovic P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y (2018) Graph attention networks. In: 6th international conference on learning representations, ICLR

Waikhom L, Patgiri R (2021) Graph neural networks: methods, applications, and opportunities. CoRR arXiv:2108.10733

Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612

Wang B, Gong NZ (2019) Attacking graph-based classification via manipulating the graph structure. In: Proceedings of the 2019 ACM SIGSAC conference on computer and communications security, CCS, pp 2023–2040

Wang C, Huai M, Wang D: Inductive graph unlearning. In: Calandrino JA, Troncoso C (eds) 32nd USENIX security symposium, USENIX security 2023, Anaheim, CA, USA, August 9–11, 2023

Wang B, Jia J, Cao X, Gong NZ (2021) Certified robustness of graph neural networks against adversarial structural perturbation. In: KDD '21: the 27th ACM SIGKDD conference on knowledge discovery and data mining, virtual event, pp 1645–1653

Wang X, Ji H, Shi C, Wang B, Ye Y, Cui P, Yu PS (2019) Heterogeneous graph attention network. In: The world wide web conference, WWW, pp 2022–2032

Wang X, Wang WH (2022) Group property inference attacks against graph neural networks. In: Yin H, Stavrou A, Cremers C, Shi E (eds) Proceedings of the 2022 ACM SIGSAC conference on computer and communications security, CCS 2022, Los Angeles, CA, USA, November 7–11, 2022, pp 2871–2884

Wang Y, Wang Y, Zhang Z, Yang S, Zhao K, Liu J (2023) USER: unsupervised structural entropy-based robust graph neural network. In: Williams B, Chen Y, Neville J (eds) Thirty-seventh AAAI conference on artificial intelligence, AAAI 2023, thirty-fifth conference on innovative applications of artificial intelligence, IAAI 2023, thirteenth symposium on educational advances in artificial intelligence, EAAI 2023, Washington, DC, USA, February 7–14, 2023, pp 10235–10243

Wang B, Yao Y, Shan S, Li H, Viswanath B, Zheng H, Zhao BY (2019) Neural cleanse: identifying and mitigating backdoor attacks in neural networks. In: 2019 IEEE symposium on security and privacy, SP, pp 707–723

Wan X, Kenlay H, Ru B, Blaas A, Osborne MA, Dong X (2021) Adversarial attacks on graph classification via Bayesian optimisation. CoRR arXiv:2111.02842

Warner SL (1965) Randomized response: a survey technique for eliminating evasive answer bias. J Am Stat Assoc 60(309):63–69

Watts DJ, Strogatz SH (1998) Collective dynamics of 'small-world' networks. Nature 393(6684):440–442

Weisfeiler B, Leman A (1968) The reduction of a graph to canonical form and the algebra which appears therein. NTI Ser 2(9):12–16

Wei Z, Xu J, Lan Y, Guo J, Cheng X (2017) Reinforcement learning to rank with Markov decision process. In: Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval, pp 945–948

Wu K, Wang C, Liu J (2022) Evolutionary multitasking multilayer network reconstruction. IEEE Trans Cybern 52(12):12854–12868

Wu K, Hao X, Liu J, Liu P, Shen F (2022) Online reconstruction of complex networks from streaming data. IEEE Trans Cybern 52(6):5136–5147

Wu L, Chen Y, Shen K, Guo X, Gao H, Li S, Pei J, Long B (2021) Graph neural networks for natural language processing: a survey. CoRR arXiv:2106.06090

Wu F, Jr. A.H.S, Zhang T, Fifty C, Yu T, Weinberger KQ (2019) Simplifying graph convolutional networks. In: Proceedings of the 36th international conference on machine learning, ICML. Proceedings of machine learning research, vol 97, pp 6861–6871

Wu Y, Lian D, Xu Y, Wu L, Chen E (2020) Graph convolutional networks with Markov random field reasoning for social spammer detection. In: The thirty-fourth AAAI conference on artificial intelligence, AAAI, pp 1054–1061

Wu L, Lin H, Huang Y, Fan T, Li SZ (2023) Extracting low-/high- frequency knowledge from graph neural networks and injecting it into MLPS: an effective GNN-to-MLP distillation framework. In: Williams B, Chen Y, Neville J (eds) Thirty-seventh AAAI conference on artificial intelligence, AAAI 2023, thirty-fifth conference on innovative applications of artificial intelligence, IAAI 2023, thirteenth symposium on educational advances in artificial intelligence, EAAI 2023, Washington, DC, USA, February 7–14, 2023, pp 10351–10360

Wu H, Wang C, Tyshetskiy Y, Docherty A, Lu K, Zhu L (2019) Adversarial examples for graph data: deep insights into attack and defense. In: Proceedings of the twenty-eighth international joint conference on artificial intelligence, IJCAI, pp 4816–4823

Wu Z, Wang Z, Wang Z, Jin H (2018) Towards privacy-preserving visual recognition via adversarial training: a pilot study. In: Computer vision—ECCV. Lecture notes in computer science, vol 11220, pp 627–645

Wu C, Wu F, Cao Y, Huang Y, Xie X: FEDGNN: federated graph neural network for privacy-preserving recommendation. CoRR arXiv:2102.04925

Wu B, Yang X, Pan S, Yuan X (2021) Adapting membership inference attacks to GNN for graph classification: approaches and implications. In: IEEE international conference on data mining, ICDM, pp 1421–1426

Wu B, Yang X, Pan S, Yuan X (2022) Model extraction attacks on graph neural networks: Taxonomy and realisation. In: ASIA CCS '22: ACM Asia conference on computer and communications security, pp 337–350

Wu S, Zhang W, Sun F, Cui B (2020) Graph neural networks in recommender systems: a survey. CoRR arXiv:2011.02260

Xi Z, Pang R, Ji S, Wang T (2021) Graph backdoor. In: 30th USENIX security symposium, USENIX, pp 1523–1540

Xu K, Chen H, Liu S, Chen P, Weng T, Hong M, Lin X (2019) Topology attack and defense for graph neural networks: an optimization perspective. In: Proceedings of the twenty-eighth international joint conference on artificial intelligence, IJCAI, pp 3961–3967

Xu K, Hu W, Leskovec J, Jegelka S (2019) How powerful are graph neural networks? In: 7th international conference on learning representations, ICLR

Xu J, Picek S (2021) Watermarking graph neural networks based on backdoor attacks. CoRR arXiv:2110.11024

Xu J, Picek S (2022) Poster: clean-label backdoor attack on graph neural networks. In: Yin H, Stavrou A, Cremers C, Shi E (eds) Proceedings of the 2022 ACM SIGSAC conference on computer and communications security, CCS 2022, Los Angeles, CA, USA, November 7–11, 2022, pp 3491–3493

Yang C, Wang H, Zhang K, Chen L, Sun L (2021) Secure deep graph generation with link differential privacy. In: Proceedings of the thirtieth international joint conference on artificial intelligence, IJCAI, pp 3271–3278

Ying R, He R, Chen K, Eksombatchai P, Hamilton WL, Leskovec J (2018) Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, KDD, pp 974–983

Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? In: Advances in neural information processing systems 27: annual conference on neural information processing systems 2014, NeurIPS, pp 3320–3328

Yun S, Jeong M, Kim R, Kang J, Kim HJ (2019) Graph transformer networks. In: Advances in neural information processing systems 32: annual conference on neural information processing systems 2019, NeurIPS, pp 11960–11970

Zang X, Xie Y, Chen J, Yuan B: Graph universal adversarial attacks: a few bad actors ruin graph learning models. In: Proceedings of the thirtieth international joint conference on artificial intelligence, IJCAI, pp 3328–3334

Zhang C, Bengio S, Hardt M, Recht B, Vinyals O (2017) Understanding deep learning requires rethinking generalization. In: 5th international conference on learning representations, ICLR

Zhang Z, Chen M, Backes M, Shen Y, Zhang Y (2022) Inference attacks against graph neural networks. In: Proceedings of the USENIX security

Zhang S, Chen H, Sun X, Li Y, Xu G (2022) Unsupervised graph poisoning attack via contrastive loss backpropagation. In: WWW '22: the ACM web conference 2022, virtual event, pp 1322–1330

Zhang Y, Gao H, Pei J, Huang H (2022) Robust self-supervised structural graph neural network for social network prediction. In: WWW '22: the ACM web conference 2022, virtual event, pp 1352–1361

Zhang Z, Jia J, Wang B, Gong NZ (2021b) Backdoor attacks to graph neural networks. In: SACMAT '21: The 26th ACM symposium on access control models and technologies, virtual event, pp 15–26

Zhang Z, Liu Q, Huang Z, Wang H, Lu C, Liu C, Chen E (2021) Graphmi: extracting private graph data from graph neural networks. In: Proceedings of the thirtieth international joint conference on artificial intelligence, IJCAI, pp 3749–3755 (2021)

Zhang Y, Regol F, Pal S, Khan S, Ma L, Coates M (2021) Detection and defense of topological adversarial attacks on graphs. In: Proceedings of The 24th international conference on artificial intelligence and statistics, pp 2989–2997

Zhang M, Wang X, Zhu M, Shi C, Zhang Z, Zhou J (2022) Robust heterogeneous graph neural networks against adversarial attacks

Zhang X, Zitnik M (2020) GNNGuard: defending graph neural networks against adversarial attacks. In: Advances in neural information processing systems 33: annual conference on neural information processing systems 2020, NeurIPS

Zhao X, Wu H, Zhang X (2021) Watermarking graph neural networks by random graphs. In: 9th international symposium on digital forensics and security, ISDFS, pp 1–6

Zhou J, Cui G, Hu S, Zhang Z, Yang C, Liu Z, Wang L, Li C, Sun M (2020) Graph neural networks: a review of methods and applications. AI Open 1:57–81

Zhou J, Chen C, Zheng L, Wu H, Wu J, Zheng X, Wu B, Liu Z, Wang L (2020) Vertically federated graph neural network for privacy-preserving node classification. arXiv preprint arXiv:2005.11903

Zhou Y, Kutyniok G, Ribeiro B (2022) OOD link prediction generalization capabilities of message-passing GNNs in larger test graphs. In: NeurIPS

Zhou Z, Zhou C, Li X, Yao J, Yao Q, Han B (2023) On strengthening and defending graph reconstruction attack with Markov chain approximation. In: Krause A, Brunskill E, Cho K, Engelhardt B, Sabato S, Scarlett J (eds) International conference on machine learning, ICML 2023, 23–29 July 2023, Honolulu, Hawaii, USA. Proceedings of machine learning research, vol 202, pp 42843–42877

Zhuang J, Hasan MA (2022) Defending graph convolutional networks against dynamic graph perturbations via Bayesian self-supervision. CoRR arXiv:2203.03762

Zhu D, Zhang Z, Cui P, Zhu W (2019) Robust graph convolutional networks against adversarial attacks. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, KDD, pp 1399–1407

Zou X, Zheng Q, Dong Y, Guan X, Kharlamov E, Lu J, Tang J (2021) TDGIA: effective injection attacks on graph neural networks. In: KDD ’21: The 27th ACM SIGKDD conference on knowledge discovery and data mining, virtual event, pp 2461–2471

Zügner D, Akbarnejad A, Günnemann S (2018) Adversarial attacks on neural networks for graph data. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, KDD, pp 2847–2856

Zügner D, Günnemann S (2019) Adversarial attacks on graph neural networks via meta learning. In: 7th international conference on learning representations, ICLR

Zügner D, Günnemann S (2020) Certifiable robustness of graph convolutional networks under structure perturbations. In: KDD ’20: the 26th ACM SIGKDD conference on knowledge discovery and data mining, virtual event, pp 1656–1665

Zügner D, Günnemann S: Certifiable robustness and robust training for graph convolutional networks. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, KDD, pp 246–256