ITC 4/53

Information Technology and Control
Vol. 53 / No. 4/ 2024
pp. 1251-1278
DOI 10.5755/j01.itc.53.4.37737

A Survey on Privacy Attacks and Defenses in Graph Neural Networks

Received 2024/06/24 | Accepted after revision 2024/11/19

HOW TO CITE: Luo, L., Ren, W., Huang, H., Wang, F. (2024). A Survey on Privacy Attacks and Defenses in Graph Neural Networks. *Information Technology and Control*, 53(4), 1251-1278. https://doi.org/10.5755/j01.itc.53.4.37737

A Survey on Privacy Attacks and Defenses in Graph Neural Networks

Lanhua Luo

School of Artificial Intelligence, Hezhou University, Hezhou, China; Faculty of Data Science, City University of Macau, Macau, China; e-mail: 200800025@hzxy.edu.cn

Wang Ren

Faculty of Data Science, City University of Macau, Macau, China; e-mail: r912124577@gmail.com

Huasheng Huang, Fengling Wang

School of Artificial Intelligence, Hezhou University, Hezhou, China;
e-mails: 200100002@hzxy.edu.cn, 201600060@hzxy.edu.cn

Corresponding author: 200100002@hzxy.edu.cn

Graph neural networks (GNNs) have emerged as a powerful tool in the field of graph machine learning, demonstrating by a various practical applications. However, the complex nature of graph structures and their expanding use across different scenarios present challenges for GNNs in terms of privacy protection. While there have been studies dedicated to addressing the privacy leakage problem of GNNs, many issues remain unresolved. This survey aims to provide a comprehensive understanding of the scientific challenges in the field of privacy-preserving GNNs. The survey begins with a succinct review of recent research on graph data privacy, followed by an analysis of the current methods for GNNs privacy attacks. Subsequently, the survey categorizes and explores the limitations, evaluation standards, and privacy defense technologies for GNNs, with a focus on data anonymization, differential privacy, graph-based federated learning, and methods based on adversarial learning. Additionally, the survey also summarizes some widely used datasets in GNNs privacy attacks and defenses. Finally, we identify several open challenges and possible directions for future research.

KEYWORDS: graph neural networks, privacy preserving, deep learning, differential privacy.

1. Introduction

Over the last decade, deep learning has achieved significant success in processing data formats like images, speech, text, and video. These domains share a commonality: their data are characterized by regular sizes and dimensions, known as Euclidean or grid-structured data [99]. However, a vast array of real-world applications involves non-Euclidean spatial data, such as protein structure prediction [27], knowledge graph completion [109], social network recommendation [73], text classification [112], and fact verification [130]. This non-euclidean spatial data can be effectively represented using graph data structures. Naturally, researchers have integrated deep learning techniques into graph-structured data, leading to the development of Graph Neural Networks (GNNs). Today, GNNs represent a significant and growing field within deep learning.

The privacy implications of large-scale, non-Euclidean structured data, typically composed of multiple interconnected roles, are playing vital roles in the real-world scenarios. These intricately interconnected roles, once abstracted into a graph-structured format, can potentially expose personal privacy information through nodes, edges, and subgraphs within the graph. Therefore, privacy research within the realm of GNNs must consider not only node attributes, but also their interrelationships. This complexity renders traditional privacy protection methods, based on euclidean spatial data in machine learning, inapplicable to graph-structured data. Hence, it is imperative to either adapt and enhance existing privacy protection methods designed for Euclidean spatial data or construct novel methods to safeguard privacy in the field of graph data. Currently, preliminary research in GNN privacy protection methods exists, such as the creation of a GNN learning framework independent of model architecture based on differential privacy [75], privacy research centered on graph federated learning [104], and the use of adversarial learning to enhance the quality of generated privacy protection data [93]. However, a comprehensive and in-depth systematic review of privacy protection approaches within GNNs is conspicuously absent, which is disadvantageous for aspiring researchers in this domain. Consequently, this paper presents a systematic exposition and analysis of the most recent research on GNN privacy attacks and defense mechanisms. It provides a classification and introduction to the topic, and delineates an outlook on future work in this domain, with the intention of assisting researchers embarking on this area of study.

Over the past few years, several surveys have summarized graph data privacy preservation methods and its applications, which can be categorized into the following categories:

1 Graph data release
- The review [60] summarized anonymization techniques for privacy-preserving data publishing.
- Jiang et al. [46] summarized the applications of differential privacy (DP) in social network analysis, including classification, challenges, adaptations, and new applications.
- Li et al. [54] discussed the private graph data release algorithms that aim to balance privacy and utility graphs. They focused on provably private mechanisms, including extensions of DP and other privacy formulations.

2 Graph adversarial learning
- The work [17] presented the vulnerability of deep learning models on graphs to adversarial attacks and the emerging field of graph adversarial learning.
- Sun et al. [83] studied adversarial attack and defense strategies for graph data. It highlighted the vulnerability of deep neural networks (DNNs) to adversarial attacks and the need for robust defense strategies.

3 Trustworthy graph learning
- In the survey [98], the authors focused on trustworthy graph learning, including reliability, explainability, and privacy protection. They emphasized the importance of ensuring that deep graph learning algorithms behave in a socially responsible manner and met regulatory compliance requirements.
- Dai et al. [20] summarized a comprehensive survey on trustworthy GNNs with a focus on privacy, robustness, fairness, and explainability. They discussed the challenges and offered a taxonomy of methods and frameworks for each aspect of trustworthiness.
- Zhang et al. [116] recently surveyed the importance of building trustworthy GNNs and proposed a roadmap to achieve this goal.

4 Heterogeneous data privacy-preserving

- Cunha et al. [19] reviewed privacy-preserving mechanisms (PPMs) for heterogeneous data types. They highlighted the importance of PPMs in protecting users' privacy and proposed a privacy taxonomy that establishes a relation between different types of data and suitable PPMs.

5 Unstructured data privacy protection

- Chen et al. [14] discussed the problem of data isolation in Knowledge Graphs (KGs) and the need for privacy-preserving techniques in KGs.
- Zhao et al. [127] focused on differential privacy for unstructured data content, including image, graph, audio, video, and text.

6 GNNs privacy and security

- The latest work [32] conducted an in-depth analysis of the security and privacy challenges faced by GNNs in practical applications, with a specific focus on adversarial attacks and their corresponding defensive strategies.

While previous survey papers focus on privacy protection theory and applications in unstructured data or graph data, the literature primarily focuses on specific aspects, such as privacy protection for data publishing, knowledge graphs, or social network analysis. Alternatively, some studies concentrate on the privacy, robustness, and interpretability of Trustworthy GNNs. There is no survey that specifically focuses on GNNs and systematically analyzes privacy attacks and defense technologies. Comprehensively understanding and systematically analyzing privacy attacks and defense technologies targeted at GNNs is a challenging task, which motivates our efforts in this paper.

The organization of this survey is listed as follows: Section 2 summarizes the attack principles and challenges of various privacy attack methods in GNNs. Section 3 and Section 4 analyze and summarize the classification, research progress, and evaluation criteria of privacy attacks and defense technologies in GNNs. Section 5 statistically analyzes commonly used datasets in existing research. Section 6 presents open challenges and future directions in GNNs privacy attack and defense technology.

2. Background

2.1. GNNs

2.1.1. Notations

Definition 2.1: Let $\mathrm{G} = \{\mathrm{V,E}\}$ represent a graph, where $V = \{\mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_\mathrm{n}\}$ constitutes a set of N nodes. Here, $e_{ij} = \langle \mathrm{v}_i, \mathrm{v}_j \rangle \in E$ indicates the presence of a connection between nodes v_i and v_j. Typically, $X \in \mathbb{R}^{n \times d}$ represents the node feature matrix, with $X_\mathrm{v} \in \mathbb{R}^d$ as individual node feature vectors. The edge set E is encapsulated by the adjacency matrix $A \in \mathbb{R}^{n \times n}$, where $A_{ij} \neq 0$ indicates the presence of an edge, and $A_{ij} = 0$ signifies its absence.

To better describe the privacy protection problem of GNNs, Table 1 lists the commonly used symbol definitions in this domain.

Table 1

Definition of common symbols

Notation	Description	Notation	Description
G	The original graph	H	Hidden layer feature information
G'	Perturbation graph	W	Weight matrix
E	Graph edges	L	Loss function
V	Graph nodes	σ	Nonlinear activation function
X	Feature matrix	Δ	Disturbance cost
X'	Perturbed Feature matrix	f	Target model
A	Adjacency matrix	f'	Alternative model
I	Identity matrix	ϵ	Privacy budget
$\tilde{A}$	$A + I$	δ	Slack variable
A'	Perturbed adjacency matrix	v_i	Node
D	Degree matrix of A	e_{ij}	Connected edges of v_i and v_j
$\tilde{D}$	Degree matrix of $\tilde{A}$	$\| \|_0$	L_0 norm

2.1.2. The Principles of GNNs

GNNs are a series of neural network architectures designed for modeling graph-structured [79, 78, 13, 34, 91]. For simplicity, this discussion will focus on two models: the basic GNN and the graph convolutional network (GCN) as representative examples.

1 Basic GNN

The foundational concept of GNNs was first introduced by Franco Scarselli et al. in 2009 [78, 79]. This pioneering work extended traditional neural network methodologies to graph data processing. It offered an efficient modeling approach to graph-structured data and contributed to the early efforts to adapt neural networks for graph data applications, and also played a crucial role in later research and development in the field of GNNs.

The primary aim of the basic GNN model is to learn a state embedding representation $h_{\mathrm{v}} \in \mathbb{R}^{s}$ for each node. The node state representation h_{v} is used to derive the model's output embedding representation o_{v}. The predictions are related to the distribution, clustering, and anomaly detection of node labels. The basic GNN uses a local transition function f to update the node state. This helps obtain the node embedding representatio h_{v} and output embedding representation o_{v}. Subsequently, a local output function g is introduced to determine the node input. The cumulative distribution is defined as follows [59]:

$$h_{\mathrm{v}} = f(x_{\mathrm{v}}, x_{co[\mathrm{v}]}, h_{ne[\mathrm{v}]}, x_{ne[\mathrm{v}]}) \tag{1}$$

$$o_{\mathrm{v}} = g(\mathrm{h}_{\mathrm{v}}, x_{\mathrm{v}})\,. \tag{2}$$

Among them, x and h represent the input features and hidden states of the node v, $co[\mathrm{v}]$ and $ne[\mathrm{v}]$ denote the set of edges and the set of nodes adjacent to v. Furthermore, x_{v} is the node's characteristics. $x_{co[\mathrm{v}]}$ is the edge's characteristics. $h_{ne[\mathrm{v}]}$ is the node's hidden state. $x_{ne[\mathrm{v}]}$ represents the adjacent nodes' characteristics.

The basic GNN is proficient in modeling structured data, but it has several limitations. These include low computational efficiency, a lack of hierarchical feature extraction capabilities, challenges in effectively modeling edge information features, and a focus on node representation rather than the graph as a whole. Additionally, it often lacks enough information to distinguish between nodes. This is due to other constraints [59].

2 GCN

Bruna et al. proposed the GCN model [13]. They were the first to generalize convolution operations to graph data from traditional data domains. GCN utilizes convolutional neural network (CNN) principles to achieve local perception of graph data. It incorporates features like translation invariance and weight sharing [12]. This groundbreaking work presented a new method and also provided guidance and a framework for improving other GNN models [99].

GCN serves as the foundation for a variety of complex GNN models, including autoencoder-based models, generative models, and spatiotemporal networks. Its innovative aspect lies in the development of a method to extract features from graph data. The extracted features are utilized in a range of applications such as node classification [126], graph classification [96], edge prediction [64], and obtaining embedding representations of graphs [114], among others. Fundamentally, GCN aims to learn a function mapping, through which nodes in the graph can aggregate their own features and those of neighboring nodes to form a new node representation. The feature propagation function of GCN is [13]:

$$H^{\ell+1} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{\ell}W^{\ell})\,. \tag{3}$$

In short, the GNN architecture, a deep learning framework grounded in graph data structures, adeptly captures both high-order content and topological information within graphs. This architecture has undergone continuous enhancements in effectiveness, robustness, scalability, and universality. Such developments have propelled the application of GNNs in numerous downstream tasks, including node classification, graph classification, link prediction, and community detection, yielding significant social and economic benefits.

2.2. Privacy Risks in GNNs

Graph neural network models can encounter various degrees of privacy risks throughout their lifecycle, potentially leading to private information leaks or compromising the model's ability to provide normal services. These privacy risks specifically manifest in several ways.

2.2.1. Model Training Stage

1 An attacker may use acquired information to determine whether a node or subgraph is part of the training set, executing a membership inference attack (e.g., [123]); or they might infer global or group attributes of the training set, constituting an attribute inference attack (e.g., [95]).

2 Attackers can manipulate the training data distribution by introducing strategically crafted samples into the training set, altering model behavior and diminishing performance. This approach leads to poisoning attacks (e.g., [135]) or backdoor attacks (e.g., [124]).

3 In federated learning scenarios, an untrusted server may engage in adversarial attacks by continuously interacting with participant parameters, thereby extracting sensitive training data information (e.g., [16]).

2.2.2. Model Prediction Stage

1 Data leakage due to insufficient generalization ability of the trained GNN model, simple model, etc., such as member inference attacks (e.g., [129]).

2 If the model prediction results are sensitive, such as the probability of disease, the attacker extracts sensitive information related to the training data based on the model prediction results and auxiliary information, and can implements model inversion attacks (e.g., [37]).

3 To obtain free model services, the attack reconstructs the training graph or shadow graph based on the model's response results and related auxiliary information, trains alternative models with similar functions, and implements model extraction attacks (e.g., [102]).

4 The attacker injects a small number of malicious nodes or edges into the test data to cause the model to make wrong decisions, that is, to implement an escape attack (e.g., [21]).

2.3. Attack Strategies in GNNs

Attack strategy refers to the method employed by attackers. Based on the unique characteristics of graph data, these strategies can be categorized into the following five types.

2.3.1. Modification of Characteristics

Node features are crucial in GNNs. Even minor adjustments to the model's interpretation of node features can significantly influence the model's output, thereby facilitating attacks. The attack cost of modifying the characteristics can be expressed as [33]:

$$\left\|X' - X\right\|_0 \leq \Delta\ . \tag{4}$$

2.3.2. Modification of Connected Edges

The topology of graph data significantly influences the graph's characteristics. Attackers can alter this topology by adding or deleting a limited number of edges, thus executing attacks. The cost associated with modifying these edges can be quantified as indicated in the formula [33]:

$$\left\|A' - A\right\|_0 \leq \Delta\ . \tag{5}$$

2.3.3. Addition of False Nodes and Corresponding Edges

To preserve the information of the original nodes and edges in the graph, the attacker carries out the attack by introducing false nodes and corresponding edges. The cost of this attack can be quantified as follows [33]:

$$\left\|X' - X\right\|_0 + \left\|A' - A\right\|_0 \leq \Delta\ . \tag{6}$$

2.3.4. Modification of Subgraph

In executing a graph misclassification attack, the attacker modifies the existing subgraph, which involves altering node characteristics or the edges within the subgraph. The cost associated with this attack, when implemented through subgraph modification, can be quantified as follows:

$$\left\|G' - G\right\|_0 \leq \Delta\ . \tag{7}$$

2.3.5. Reconstruction of Graph Data

By integrating dataset distribution characteristics, model output results, and model parameters, along with other pertinent information, an attacker can reconstruct the graph data or create a shadow graph. This reconstructed data is then utilized to train an alternative model f', facilitating model extraction at-

Table 2

Privacy attack methods and common attack strategies in GNNs

Attack stage	Attack Method	Attack strategy
Model training	Membership inference attack	Modification of X or E, reconstruction of G
	Attribute inference attack	Modification of X or E, reconstruction of G
	Poisoning attack	Modification of X or E, add nodes and edges
	Backdoor attack	Modification of X or E, add nodes and edges
Model prediction	Membership inference attack	Modification of X or E, reconstruction of G
	Attribute inference attack	Modification of X or E, reconstruction of G
	Model inversion attack	Reconstruction of G
	Model extraction attacks	Modification of X or E, reconstruction of G
	Evasion attack	Modification of X or E, add nodes and edges

tacks, or to further implement other forms of attacks, such as inference attacks. The cost can be quantified as follows:

$$\left\| f' - f \right\|_0 \leq \Delta . \qquad (8)$$

Table 2 provides an overview of the various methods used for privacy attacks and the common strategies employed in GNNs.

3. Privacy Attack Methods in GNNs

Existing research in the field of privacy attacks in machine learning mainly focuses on Euclidean space. In contrast, there is a scarcity of studies exploring privacy attack methods in GNNs. Privacy violations on GNNs can fundamentally be viewed as extracting non-shareable information embedded in a model or training dataset. This non-shareable information can include member affiliations, node attribute knowledge, link relationships between nodes, and model parameters. Privacy attacks against GNNs, depending on the specific target, can broadly be classified into five distinct categories.

3.1. Membership Inference Attacks

The objective of a membership inference attack is to ascertain whether the target sample was used in the training of the model *f*, thereby exposing the privacy of the training dataset. Recently, given the surge in the application of GNNs, membership inference attacks based on graph machine learning have garnered increasing attention [24, 38, 70, 101, 58, 18]. In the realm of membership inference attack models, the most prevalent method involves the utilization of a shadow model or a shadow dataset. This method engages membership reasoning to determine whether the target sample was used as a training sample for the model. For instance, in a node classification task, the target sample might be a specific node [38], sensitive attributes or topological structures within the node [38, 70], or a subgraph from the target node's local graph [101, 123]. For graph classification tasks, the target might also be a graph awaiting classification [101]. It is important to note that targets with higher subgraph densities are more susceptible to membership inference attacks. Even when the adversary is unaware of the training data distribution or the architecture of the target model, the attack remains effective [38].

In existing research, Zhang et al. [123] were the first to explore the privacy concerns associated with graph embedding, introducing a subgraph inference-based attack method. This approach allows for the successful determination of whether a subgraph is part of the target graph, its effectiveness having been validated through experimental means. Liu et al. [58] have examined the interplay between graph adversarial attacks and privacy breaches, discovering that models trained via graph adversarial methods can greatly enhance the success rate of graph membership inference attacks. This enhancement can primarily be attributed to the distinct performance of the robust model's loss function on training and test datasets. Mauro Conti et al. [18] investigated a more complex attack scenario where the GNN model outputs only labels, by relaxing the assumptions of a similarly distributed shadow dataset and knowledge of the target model's architecture (similar to approaches in [37] and [38]). Obviously, this represents a more practical application scenario. The attacker leverages fixed attributes, 0-hop, and 1-hop queries to construct attack features, achieving better performance compared to probability-based membership inference attacks. Additionally, Zhong et al. [129] first studied the subgroup vulnerability differences in link-level membership inference attacks (LMIA) on GNNs. They identified a strong correlation between varying subgroup structural attributes (such as density, node similarity, and average edge betweenness centrality) and attack performance. Consequently, they designed a balanced fairness algorithm to counter LMIA, which employs fixed probability randomization on the original graph to perturb edge memberships. This method avoids iterative accumulation on the target model, effectively balancing LMIA defense performance and model utility while reducing vulnerability disparities among subgroups.

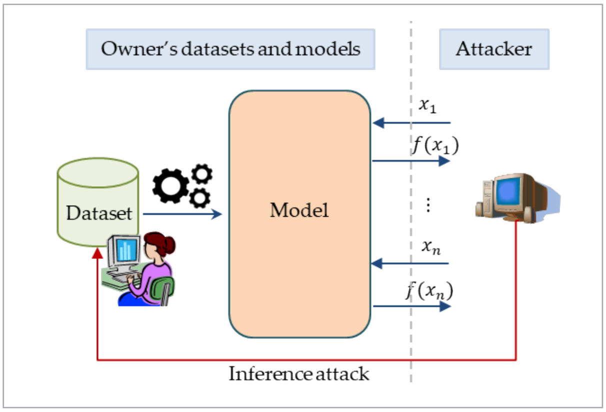
Generally, node-level and link-level membership inference attacks heavily rely on graph data attributes and model architecture. Notably, even when the fitting is normal, robustness against privacy attacks cannot be guaranteed. In contrast, graph-level membership inference attacks are primarily influenced by the target model's degree of overfitting, which is a critical determinant. This is in line with research on membership inference attacks on data with Euclidean-structured within the traditional machine learning domain.

3.2. Attribute Inference Attacks

Similar to membership inference attacks, attribute inference attacks also concern with the training data. However, the latter seeks to infer specific, potentially sensitive, attributes of the training data based on the model's outputs, non-sensitive attributes, and other information. Both membership and attribute inference attacks extract private information from training data through inference, and as such, they both fall under the umbrella of inference attacks. Consequently, they can be represented through the same inference model framework (refer to Figure 1). Consider the node classification task as an example: an attacker submits a query x_i to the shadow model, obtains the output's $f(x_i)$ probability value, and infers whether it is in the training data set: a typical scenario for a membership inference attack. In another scenario, the attacker may acquire the entire target graph G and a subgraph G' of interest as input [123], aggregate the embeddings of the target graph G and subgraph G', and derive attributes (such as degree distribution, number of nodes, etc.). This is a typical process of attribute inference attacks. The primary distinction between them is that member inference attacks are concerned with determining whether individual nodes, local nodes, or subgraphs were utilized in training the model. In contrast, attribute inference attacks aim to infer specific global attributes of the training dataset [42].

Figure 1

General model of inference attacks



At present, there is a limited body of research on attribute inference attacks within the realm of graph data. Notably, Zhang et al. [123] introduced a method to infer basic attributes of a target graph embedded within a given graph, which includes the count of nodes, edges, and the density of the graph. Their approach frames the attack as a multi-task classification problem, enabling the prediction of all graph attributes of interest simultaneously, thus achieving high attack accuracy.

For the first time, Wang et al. [95] engaged in a systematic exploration of group attribute inference for GNNs. They categorized threats from both white-box (where the model architecture and parameters are known) and black-box (where only the model output is accessible) scenarios based on different attack knowledge into six types, designing attack methods for each category. They discovered that with just 20% of the training graphs, high-precision inferences can be made for node and link attributes in the range of $[0.9, 1]$ and $[0.72, 0.92]$ respectively. Even when a shadow graph is utilized for the attack, the accuracy can reach as high as 0.66. Thus, the privacy leakage of group attributes for GNNs is a significant concern.

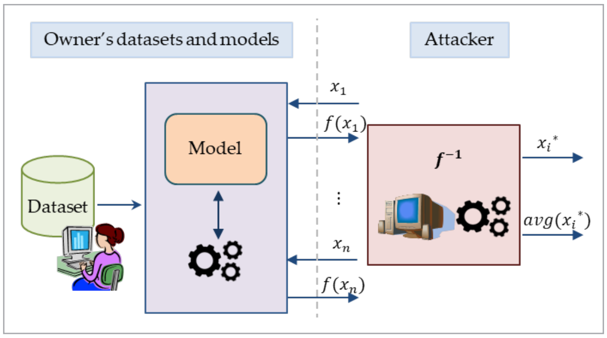
In a different approach, Olatunji et al. [69] proposed three attribute inference attack methods, including attribute inference attacks via repeated queries of the target model, those based on feature propagation solely, and shadow-based attribute inference attacks. While prior research predominantly focused on inferring single binary attributes, their work differs in that it can infer single or multiple binary attributes, as well as continuous attribute values. In conclusion, node attribute inference attacks are commonly employed in scenarios where attribute features and labels are strongly correlated. However, even when there is a weak correlation between node attribute features and task labels, the implementation of group attribute inference attacks can still lead to privacy leakage.

3.3. Model Inversion Attacks

Model inversion attacks constitute a form of privacy attack aimed at inferring sensitive information concealed within training data. Attackers extensively gather information through diverse channels, encompassing not only the labels of certain nodes but also auxiliary knowledge such as node attributes, node identifiers, and edge density. Initially, the attackers integrate and analyze this data, subsequently transforming the inversion task into a complex optimization problem. The core objective is to construct data that closely approximates the original target data, enabling the inference of node attributes, inter-node connections, and even the reconstruction of the entire graph's adjacency matrix. The attack framework is shown in Figure 2. Model inversion attacks treat the inversion task as an optimization problem. The optimal data x_i^* corresponding to the target class data x_i, is constructed through gradient methods, aiming to make $f(x_i)$ and $f(x_i^*)$ as close as possible, thereby inferring the sensitive features of the training data.

Figure 2

General model of model inversion attack



Wu et al. [105] discovered that features, particularly edges, play a significant role in privacy leakage during graph model inversion attacks. Taking inspiration from this work [105], Zhang et al. [125] proposed a method called GraphMI, which employs an optimized model inversion attack approach to reconstruct the adjacency matrix in a white-box setting with known training model parameters. The fundamental process of GraphMI involves three steps: initially, using the projected gradient descent method to identify the optimal network topology where the nodes are located; subsequently, forwarding the adjacency matrix and feature matrix to the graph autoencoder module, which gets parameters from the target model; finally, interpreting the optimized graph as an edge probability matrix, followed by sampling a binary adjacency matrix. Their research elucidates the correlation between edge influence and inversion risk, affirming that "the greater the edge influence, the greater the adversary's advantage".

He et al. [37] proposed a black-box method that steals links (based on link prediction). This approach assumes that the attacker has access to a dataset extracted from a distribution similar to the target data (shadow dataset). The core idea is to use a heuristic method to predict the attribute similarity (cosine similarity) of two nodes in the training dataset and determine whether two nodes are connected based on the degree of similarity. Experimental outcomes demonstrate that abundant graph structure information can be stolen through prediction.

Zhang et al. [123] proposed a method to initiate a graph reconstruction attack using the graph autoencoder paradigm, capable of reconstructing a graph with similar graph structure statistics (such as degree distribution, local clustering coefficient distribution) to the target graph. The cosine similarity of the local clustering coefficient distribution between the target graph and the reconstructed graph is as high as 0.99, sufficiently validating the effectiveness of the graph reconstruction attack method.

Zhang et al. [115] addressed the issue of variable degrees of link-stealing attacks from different groups, unveiling the theory of unequal vulnerability across different groups, and proposed a group-based attack paradigm. This paradigm allows customizing different attack strategies for different groups, achieving superior attack performance.

Zhou et al. [132] utilized the original Markov chain approximate attack chain to model and implement graph reconstruction attacks in a white-box attack scenario, enhancing attack fidelity through parameterization techniques and the introduction of randomness.

For the first time, Olatunji et al. [71] explored the possibility that feature explanations published in GNN may leak structural information of training nodes. Their study hypothesized that the attacker can obtain the model's feature explanations and designed attacks ranging from simple similarity-based to complex graph reconstruction attacks using graph structure learning technology. They quantified the information leakage of the graph structure via the attack success rate, and generated feature-based interpretations through three distinct methods based on gradient, perturbation, and agent model. The results indicate that gradient-based methods reveal the most information.

Research on adversarial reverse engineering methods for GNN models can enhance our understanding of their vulnerabilities and enable us to proactively mitigate privacy risks. However, there remains a notable lack of research on adversarial reverse engineering specifically targeting GNNs. Currently, implementing adversarial reverse attacks on GNNs faces three major challenges. Firstly, owing to the discrete nature of the graph, computing and optimizing the gradient on its binary edge is challenging, rendering existing model inversion attack methods inapplicable to the graph. Secondly, existing model inversion techniques do not sufficiently leverage the intrinsic characteristics of graph sparsity and feature smoothness. Lastly, the current model inversion attack methodologies fail to fully leverage on node attribute information and GNN model data.

3.4. Model Extraction Attacks

The principle of model extraction attacks is that the attacker submits queries to the target model, inferring model parameters or creating a machine learning model with similar functionality based on the responses. Successful extraction and misuse of the model can lead to significant privacy breaches, involving the disclosure of extensive private data. The typical procedure starts with the attacker issuing query requests to the model, collecting maximal information from the responses. Subsequently, the attacker uses the gathered input-output data to develop and train a knockoff model. This process, illustrated in Figure 3, involves the attacker owning data and a pre-trained model f. They send a query request $(x_1, x_2, \ldots, x_n)$ to the target, receive response $(f(x_1), f(x_2), \ldots, f(x_n))$, and create a query-response pair $((x_1, f(x_1)), (x_2, f(x_2)), \ldots, (x_n, f(x_n)))$. Using this pair and additional knowledge, they train and extract an alternate model f', epitomizing a standard model extraction attack.

Current research on model extraction attacks primarily focuses on traditional machine learning models in areas such as graphics and text, as outlined in references [90, 7, 30, 106]. However, there is limited investigation into potential attacks on GNNs.

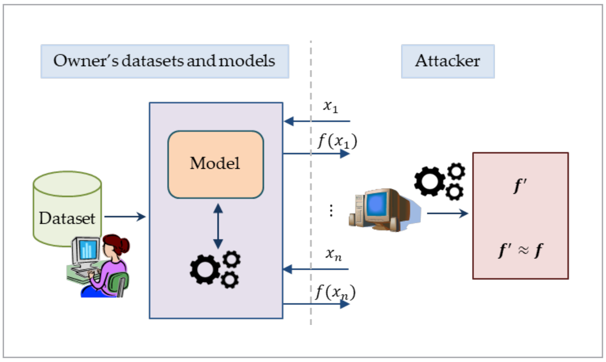
David et al. [22] explored a scenario where the adversary queries only the predicted labels via the target model's API. They trained the model on the labels of false sample graphs and learned the model by iteratively altering the sample subgraphs of the target's original graph, conduct model extraction training on various instances of these modified subgraphs. The requirement is that each class should have at least one sample in the sample graph data, and can reach an output fidelity of 80%.

Another study [100] involved generating legitimate appearing queries as ordinary nodes in the target graph, extracting GNN models through responses, graph structure information, and other available knowledge, and subsequently reconstructing models with similar functions.

Wu et al. [102] proposed a threat modeling framework based on black-box attack scenarios. This framework encompasses seven types of model extraction attacks with varying degrees of background knowledge, considering aspects like node attributes, graph structure, and shadow subgraphs. The core strategy involves using known background knowledge to create a substitute graph for model training and extracting a model that mirrors the target model.

Figure 3

General model of model extraction attack



Building on previous research [102, 22], another paper [82] introduced a model stealing attack method applicable to inductive GNNs. In this approach, suited for black-box attack scenarios, the adversary queries the target model through a remote access API, with the model's structure and training process remaining opaque. This study demonstrated that even without access to graph structure information, adversaries could still conduct effective model theft attacks.

3.5. Graph Injection Attacks

Graph injection attacks allow an attacker to introduce a limited number of nodes or edges, but prohibit the modification of the original graph's nodes or edges. This emerging attack method aligns with real-world scenarios. Depending on the attack target, stage, and method, graph injection attacks can be categorized into poisoning attacks [135, 11, 94, 23], evasion attacks [21, 87, 134, 48, 118], and backdoor attacks [124, 111, 128]. This section aims to provide a nuanced understanding of graph injection attacks by classifying and comparing them.

1. **Poisoning attack.** The objective of a poisoning attack is to alter the model's behavior during inference by modifying training data, such as label flipping or introducing malicious data. This compromises the model's accuracy or makes it susceptible to similarly modified samples. Zügner et al. [135] demonstrated an attack on a single node of a graph neural network, primarily focusing on poisoning attacks but also applicable to evasion attacks. This approach manipulates graph topology and node features while preserving key data characteristics (e.g., degree distribution, feature co-occurrence). To minimize detection, the authors developed Nettack, an algorithm based on linearization ideas, to calculate the subtlety of these attacks. Bojchevski et al. [11] altered a limited number of edges to degrade the embedding effect of the new graph, marking the first study on attacking node embedding. This research used spectral random walk algorithms and eigenvalue perturbation theory to effectively approximate spectral changes. In [94], an attack method without node injection limits was proposed, offering a linearized attack model with an optimized, lower time-cost strategy. Ding et al. [23] incorporated an attention mechanism in the link detection network to enhance the GNN model's focus on node connections. This resulted in more similar outputs for connected nodes and increased dissimilarity between unconnected nodes, improving node adjacency inference. The black-box setup and online learning in this study accommodate real-world application scenarios.
2. **Evasion attack.** Evasion attacks involve manipulating a model's output by modifying inputs in a manner imperceptible to humans. These attacks manifest during the model prediction phase, with the fundamental principle being the injection of nodes that propagate malicious attributes to pertinent nodes through feature aggregation. This process ultimately leads to erroneous predictions by the model. While research on evasion attacks targeting GNNs is relatively recent, it has garnered more attention from scholars compared to other forms of graph injection attacks. Among the pioneering works is Dai et al. [21], who proposed

an attack methodology based on reinforcement learning to maximize classification error rates by minimally altering edges. Tao et al. [87] addressed the issue of excessive node injection by introducing an evasion attack method based on single-node injection, utilizing Gumbel-Top-k technology for high-dimensional discrete attribute optimization. The edge injection budget Δ was limited during the attack process, and the enforced injection properties were kept consistent with the original graph. Zou et al. [134] suggested a gray-box-mode evasion attack, introducing a topological flaw edge selection strategy, selecting original nodes associated with injected nodes, and designing smooth feature optimization objectives to generate characteristics of the injected nodes. Such carefully designed perturbations are challenging to identify and possess strong concealment, enabling effective injection attacks. Ju et al. [48] used a black-box setting and employed node generators and edge samplers to create adversarial nodes, modeling node injection attacks through the Markov decision process, thus contributing new theories and methods for graph-structured data security. A recent work by Zhang et al. [118] considered that a fixed budget may lead to attack failure and proposed a topology attack method based on minimum budget. This method utilizes a dynamic projected gradient descent algorithm to alternately update perturbations and budget, achieving a minimum budget topology attack on the GNNs.

3 **Backdoor attack.** Backdoor attacks are analogous to poisoning attacks but differ in their targets and processes. Like poisoning attacks, backdoor attacks also occur during the training phase. The attacker employs a data poisoning method, embedding hidden backdoors into the GNNs using training data laced with triggers. The backdoor is activated only under specific conditions. If the model processes benign data, it will produce correct predictions; however, it will exhibit abnormal behavior when processing data containing triggers. Zhang et al. [124] introduced a method involving randomly generated subgraphs. In this method, nodes are randomly selected and connected into subgraphs based on a probability matching the original graph's density. These subgraphs are then injected into the training set, and their labels are modified to create a dataset with a backdoor. The resulting model is a graph neural network compromised by a backdoor. Following a similar approach, Yang et al. [111] conducted a more comprehensive study on backdoor attacks in GNNs. Their study involved injecting predefined subgraphs into the test graph, causing the GNNs to favor the attacker's chosen target label during predictions. Previous studies often used randomly or gradient-based generated subgraphs as triggers for backdoor attacks, potentially overlooking the relationship between the trigger structure and the effectiveness of the attack. Acknowledging this, Zheng et al. [128] explored the impact of loops and statistically significant patterns in the graph on attack strategies. They discovered that triggers based on subgraphs with lower frequencies of occurrence yielded better attack performance. Consequently, the authors developed a method for generating triggers based on topic statistical information, which showed promising attack performance. However, this method requires extensive model access, which may increase the risk of detection.

Research on graph injection attacks encompasses studies on poisoning attacks and evasion attacks, with backdoor attacks receiving comparatively less attention. Poisoning attacks occur during the training phase, while evasion attacks take place in the testing phase. Backdoor attacks, which also primarily occur during training, differ in that they target model security rather than data security. Furthermore, attacks can be categorized based on the attacker's knowledge level into white-box, gray-box, and black-box approaches. Significant research has focused on adversarial example attacks [81, 55], which involve minor modifications to existing samples. These alterations, such as changing node attributes or the connections between nodes, can occur during either the training or testing phases. These attacks align with the principles of poisoning and evasion attacks and thus can be categorized accordingly. Table 3 provides a comprehensive summary of representative research on injection attacks in GNNs, encompassing six dimensions: tasks, attack types, background knowledge, modification content, and key technologies.

Table 3

Analysis of representative research work on graph injection attacks

Ref.	Attack type	Knowledge	Modification	Task	Methods
Ref. [135]	Poisoning attack, Evasion attack	Gray-box	Node & Edge	Node classification, Graph classification	Greedy algorithm, Linear model
Ref. [11]	Poisoning attack	White-box	Edge	Node classification, Link prediction	Random walk algorithm
Ref. [94]	Poisoning attack	Black-box	Node & Edge	Node classification	Fast Gradient-Sign, Linear model
Ref. [23]	Poisoning attack	Black-box	Edge	Node classification, Link prediction	Self-attention mechanism, Projected gradient descent
Ref. [21]	Evasion attack	White-box, Black-box	Edge	Node classification, Graph classification	Reinforcement learning, Genetic algorithm
Ref. [87]	Evasion attack	Black-box	Node & Edge	Node classification	Gumbel-Top-k technique, Reinforcement learning
Ref. [134]	Evasion attack	Gray-box	Node	Node classification	Defect edge selection, feature optimization
Ref. [48]	Evasion attack	Black-box	Node & Edge	Node classification	Markov decision, Reinforcement learning
Ref. [118]	Evasion attack	Black-box	Edge	Node classification	Dynamic PGD, Minimum budget control
Ref. [124]	Backdoor attack	Black-box	Subgraph	Graph classification	Randomly generate subgraphs
Ref. [111]	Backdoor attack	Black-box	Subgraph	Graph classification	Randomly generate subgraphs
Ref. [128]	Backdoor attack	Black-box	Subgraph	Graph classification	Topic-based backdoor attacks
Ref. [81]	Evasion attack	Black-box	Node	Node classification	Q-learning network, Jaccard distance
Ref. [55]	Poisoning attack	White-box, Gray-box	Edge	Node classification	Disturbance evaluation function

3.6. Summary

From the above analysis, it can be seen that GNNs predominantly confront five types of privacy attack methods. Notably, member inference attacks and graph injection attacks have received higher attention. Member inference attacks deduce the presence of specific records in a dataset, assessing their membership status. This area is a current research focus. Graph injection attacks compromise the graph model's output by introducing fictitious nodes, edges, or subgraphs, affecting data and model security. Model inversion attacks target the theft of private information from the training dataset, potentially undermining the dataset owner's commercial interests. Model extraction attacks involve developing alternative models as a research strategy, which could underpin other attacks such as member inference and attribute inference attacks. Table 4 summarizes the statistics and application areas of privacy attack methods on GNNs in key research works.

Table 4

Statistics of privacy attack methods for GNNs and their application areas

Privacy attack method	Ref.	Application Areas
Member inference attack	[123], [129], [24], [38], [70], [101], [58], [18]	Social networks, financial systems, recommendation systems, healthcare
Property inference attack	[123], [95], [42], [69]	Social networks, financial systems, recommendation systems
Model inversion attack	[123], [37], [24], [125], [115], [132], [71]	Social networks, recommendation systems, financial systems, healthcare, Natural language process
Model extraction attack	[102], [22], [100], [82]	All fields
Graph injection attack	[135], [124], [94], [11], [21], [23], [87], [134], [48], [118], [111], [128], [81], [55]	Social networks, financial systems, healthcare, cybersecurity

4. Privacy Defense Technology in GNNs

4.1. Data Anonymization

Data anonymization involves the application of techniques such as replacement, generalization, and clustering to process personal privacy information within a dataset. The process entails blending individual privacy information with other data to effectively conceal the true attribute characteristics with the aim of protecting individual private information. Anonymization technology serves as a commonly used method for privacy defense in various scenarios, including data release, transmission, and shared use.

Liu et al. [56] introduced $k-$anonymity technology [84] to the field of graph data, proposing the concept of $k-$ anonymity. A vector is considered k- anonymous when each element appears at least k times. For instance, vector $v=[5, 5, 1, 1, 1, 4, 4]$ is a 2-anonymous vector. They also introduced the notion of k- anonymous graphs, where the degree sequence vector satisfies the $k-$ anonymity property. The choice of $k-$anonymity for GNNs is suitable because it can effectively conceal the identity of individual nodes within the graph, thereby protecting privacy, while still allowing for meaningful analysis of the graph's structure and properties. This is particularly important in social network data where preserving the privacy of individuals is crucial.

Backstrom et al. [9] proposed a privacy defense technique that replaces identifiable attributes with synthetic identifiers prior to publishing real graph data on social networks. However, this method is susceptible to background knowledge attacks, enabling attackers to infer vertex identity from structural characteristics. To address this, Meden et al. [61] proposed a face image recognition method that hides personal identity information within the image. Their approach combines a GNN with an anonymity mechanism, offering a formal guarantee for privacy defense on closed identity datasets. Furthermore, Tian et al. [89] proposed a two-stage GNN privacy defense method in social networks. In the first stage, they designed an anonymization method, incorporating classic local differential privacy (LDP) and $k-DA$, to achieve both ϵ – local differential privacy and $k-$ anonymity. In the second stage, an adversarial training mechanism was developed to enhance the GNN model's resistance to ϵ – k anonymization interference. The experiments confirmed that the ϵ – k anonymization method effectively preserves the privacy of social network data while maintaining performance in tasks such as node classification, link prediction, and graph clustering. Researchers have also introduced clustering-based anonymization [88], random walk-based anonymization [63], and combined anonymization with other privacy defense technologies in graph data [26] to counter background knowledge attacks and homogeneity attacks. However, these methods still face challenges in achieving a balance between the availability and privacy of anonymous data.

To summarize, current research in graph data anonymization largely focuses on adaptive optimization of existing anonymization technology or proposes targeted protection for specific vulnerable feature dimensions. Given the complexity of graph data, relying solely on anonymization technology to protect the privacy of node features or labels proves inadequate. Attackers can infer private information from partial topological structures of graph data, inter-node links, non-anonymous node attributes, and other types of background knowledge. Moreover, the embedding representation of anonymized graph data often exhibits poor usability, diminishing the effectiveness of GNN training. Despite these challenges, the integration of anonymization techniques with GNNs remains a promising direction for privacy defense in graph data, as evidenced by the growing body of research in this area.

4.2. Differential Privacy

Differential privacy (DP) stands as a robust privacy protection standard initially proposed by Dwork et al. [25] in 2006. It has garnered widespread attention among researchers due to its stringent mathematical definition and quantifiable privacy protectionmodel. The formal definition of (ϵ, δ) – differential privacy is provided below.

Definition 4.1: D and D' are two adjacent datasets, M is a random query function, the parameter ϵ is the privacy budget, (ϵ, δ) – differential privacy is deemed satisfied if the following inequality holds [25]:

$$P(M(D) \in S) \leq e^{\epsilon} P(M(D)' \in S) \ . \tag{9}$$

the output results of adjacent datasets are influenced by the ϵ parameter value to be nearly identical, implying that $M(D)$ and $M(D')$ are approximately equal in a probabilistic sense [59]. This makes it exceedingly challenging for attackers to distinguish between adjacent datasets. A smaller value of ϵ indicates a stronger privacy defense capability. The variable δ is an optional slack term, equivalent to the probability of ϵ failure. In practical applications, δ is typically set to a very small value, such as 10^{-4} [59].

Differential privacy, initially applied to privacy defense technology in databases [10], has since witnessed significant research progress in the realm of machine learning security [1, 72, 62, 6]. In the context of graph data structures, differential privacy manifests in three distinct categories based on its application to different components: node differential privacy, edge differential privacy, and graph differential privacy. Furthermore, differential privacy can be categorized into two overarching types: centralized differential privacy (CDP) and local differential privacy (LDP). In CDP, the mechanism introduces noise to the data by defining global sensitivity, subsequently imposing statistical constraints on the quantitative boundary of privacy information leakage. This approach does not impose specific requirements on the volume of statistical data. Conversely, LDP adds noise to individual data points, where the introduced noises encompass both positive and negative perturbations. The aggregation of perturbation results offsets the positive and negative noise, requiring substantial datasets to ensure unbiasedness in the final statistical outcomes. The stochastic nature of the noise underscores the necessity for extensive datasets to meet the dual demands of data availability and statistical accuracy.

To safeguard the privacy of graph-structured data, Hay et al. [35] pioneered the application of differential privacy technology in 2009, aiming to securely release graph data. They introduced two variants of differential privacy for graph data release, namely Point Differential Privacy and Edge Differential Privacy. The algorithm employs a sophisticated two-stage random perturbation process to obtain the degree distribution of the graph. Olatunji et al. [70] extended the Private Aggregation of Teacher Ensembles (PATE) method from the literature [72]. They utilized random subgraph sampling of the teacher training set and a noise labeling mechanism for public data. Combining private graph knowledge with the "teacher" model's insights on the existing recommendation system, they trained the "student" model to achieve the release of graph-structured data with a guarantee of differential privacy. Zhang et al. [120] addressed potential attackers in existing recommendation systems who might leverage user behavior trajectories and recommendation results to infer user privacy attributes. They proposed a two-stage combination strategy involving user feature perturbation in the input stage and optimization stage perturbation to obtain a privacy-preserving GNN for recommendation. Experimental results demonstrated the method's effectiveness in defending against attribute inference attacks. Sajadmanesh et al. [75] considered scenarios where local node data may be anonymized due to potentially sensitive information. To address the reduction in node availability and its impact on model training caused by anonymization, they proposed a GNN learning framework, the Local Privacy Graph Neural Network (LPGNN). Based on LDP and independent of model architecture, this framework effectively protects the privacy of node data. It can be combined with any GNN model independently, thereby reducing communication overhead. Additionally, drawing inspiration from literature [2, 3, 4], they introduced a graph convolution layer, KProp, based on multi-hop aggregation of node features. This innovation enhances the information of the aggregated neighborhood set, thereby improving the denoising process's effectiveness and the efficiency of graph convolution estimation accuracy. Zhang et al. [119] proposed a framework for decoupled graph neural networks called DPAR, which achieves node-level differential privacy by leveraging a differentially private approximate personalized PageRank algorithm coupled with differentially private stochastic gradient descent, thereby optimizing the trade-off between privacy and utility.

There is also literature available that discusses privacy leakage issues based on the similarity of graph data. Yang et al. [110] demonstrated that attackers can discern specific node details in a target network by examining the analogous attribute node degree distribution and triangle count between two social networks. This study enhanced the representation of the global graph structure using a GCN network, complemented by a generative adversarial network (GAN) [31] and an enhanced GVAE model [49]. Theoretical and empirical evaluations have confirmed that these frameworks robustly safeguard the privacy of the graph's overall structure and its edge connections. Imola et al. [44] focused on privacy preservation in scenarios involving k-star and triangle subgraph enumeration. They introduced noise addition techniques grounded in differential privacy, proposing both single-stage and multi-stage approaches. The single-stage method, which directly injects noise into graph edges for triangle detection, was shown to be flawed. Consequently, they advocate for a two-stage noise addition strategy, which includes a corrective mechanism to ensure accuracy in triangle counting.

In summary, differential privacy introduces noise to various components of the graph (such as nodes, edges, or the entire graph) or to the GNN model (parameters), thereby preventing attackers from inferring sensitive information about the graph data or the GNN model. While this technique is effective for privacy preservation, it can adversely affect the utility and accuracy of the model, resulting in decreased precision. Consequently, it remains unsuitable for applications requiring high accuracy. Table 5 encapsulates

Table 5

Comparison of representative research on differential privacy protection in GNNs

Ref.	Type	Optimize target	Noise mechanism
Ref. [75]	LDP, Node DP	Communication cost	Multi-bit
Ref. [70]	LDP, Node DP	Effectiveness	Laplace, Gaussian
Ref. [35]	CDP, Node DP, Edeg DP	Execution time, Effectiveness	The mechanism of Ref. [36]
Ref. [120]	LDP, Node DP	Effectiveness	Laplace, Combination mechanism
Ref. [15]	LDP, Node DP	Execution time, Computing costs	Gaussian, J-S estimator
Ref. [40]	LDP, Edge DP	Effectiveness	Laplace
Ref. [110]	LDP, Edge DP	Effectiveness	Gaussian
Ref. [44]	LDP, Edge DP	Relative error, Effectiveness	Laplace
Ref. [50]	LDP, Edge DP	Effectiveness	Laplace
Ref. [65]	LDP, Edge DP	Computing costs, Effectiveness	Gaussian
Ref. [8]	LDP, Edge DP	Computing costs, Effectiveness	Gaussian
Ref. [77]	CDP, Node DP, Edge DP	Computing costs, Effectiveness	Gaussian
Ref. [119]	Mix mode, Node DP	Effectiveness	Laplace, Gaussian

the salient works on differential privacy in GNNs, delineating types, fundamental concepts, optimization objectives, and mechanisms for noise integration.

4.3. Federated Learning

Since its inception, Federated Learning [51] has rapidly emerged as a focal point of research in both academia and industry, with practical applications in numerous domains including smart healthcare, the Internet of Things, edge systems, and autonomous driving [80]. This distributed data training and aggregation architecture ensures the consistency of a jointly trained model through a series of steps including local training, local updates upload, secure server-side aggregation, and global model download. Federated learning enables data to be trained at the terminal (data holder), thereby addressing the issue of single data features in the model training phase. Additionally, it offers a local privacy protection mechanism that effectively utilizes local computing resources for model training, thereby mitigating the risk of private information leakage during data transmission.

1 **Privacy risks in federated learning.** Several additional privacy risks persist within federated learning, encompassing three distinct facets. Firstly, attackers can potentially infer data from participants and misappropriate model parameters by reverse-engineering the aggregated gradient and weight information transmitted by the central server. Secondly, the federated learning framework inherently trusts all participants, thus it is susceptible to the risk of malicious entities exploiting this trust by contributing falsified data during training to extract private information. Finally, the base model supplied by a third-party platform could itself introduce privacy risks, such as the potential for the illicit embedding of viruses or unauthorized collection of parameter information.

2 **Mitigating data leakage in FedGNNs.** Zhou et al. [15] introduced a federated graph neural network (FedGNNs) learning approach, termed VFGNN. This research presumes the model's resilience against semi-honest attackers and bifurcates the computation into two segments. Calculations pertaining to private data are delegated to the data holder, whilst the server undertakes the remaining computations. The application of differential privacy safeguards the private information on the server side. The computational methodology employed by VFGNN not only preserves private data, but also enhances the model's accuracy and efficiency.

Additionally, Ni et al. [68] developed a vertical federated learning framework (Fed-VGCN) based on graph convolutional networks. FedVGCN incorporates a self-supervision mechanism and initially splits the graph data computation into two segments. Training occurs on two clients; each client employs additive homomorphic encryption to transfer intermediate results to the counterpart during the training process iterations, thereby ensuring privacy protection. Experimental validations were performed on the FedVGCN and GraphSage models. The findings indicate that FedVGCN surpasses GNN models trained on isolated data and is on par with conventional GNN models trained on combined plaintext data.

Wu et al. [103] proposed a federated GNN for privacy preservation in recommendation systems. Initially, the client uses local differential privacy to transmit the noise-added gradient to the server, and employs pseudo-interaction terms to ensure local user data items interaction anonymity. Subsequently, the client's embedded representation is uploaded to the server to offer personalized services. Experimental findings demonstrate that this model's recommendation accuracy rivals that of existing centralized GNN recommendation methods and effectively safeguards user privacy.

Following this, the authors executed an optimization [104] based on the work of [103] in three dimensions: firstly, further dividing the embedding and gradient modules of the FedGNN framework; secondly, excluding adjacent modules prior to model training commencement. User embedding is adjusted and then integrated into model training, addressing the issue of potential inaccurate user embedding due to inadequate model adjustment. The privacy budget ϵ definition method has been updated from $\epsilon = 2\delta/\lambda$ to $\epsilon = 2\delta_e/\lambda$, where e denotes the number of epochs. Furthermore, the RSA algorithm is designated as the homomorphic encryption algorithm for the user items involved in the interaction, and the average privacy protection ratio is adjusted. The research findings reveal a significant reduction in prediction error.

Rizk et al. [74] extended the client edge hierarchical federated learning architecture of [57] to graph-structured data, incorporating cryptography and differential privacy technology. The server employs the FedAvg mechanism for local training and aggregation of training results, adds noise to the updates during each round of client-server communication using differential privacy, and maps the updates to an encrypted version. The study explores the influence of privatization on algorithm performance under convexity and Lipschitz conditions.

Lastly, Gauthier et al. [29] proposed a personalized graph federated learning framework that enables distributed servers and their edge devices to learn collaboratively while maintaining each device's privacy and ensuring security. The framework primarily leverages differential privacy (especially zero-centralized differential privacy) for privacy protection, and mathematical analysis indicates linear time convergence and reasonable accuracy.

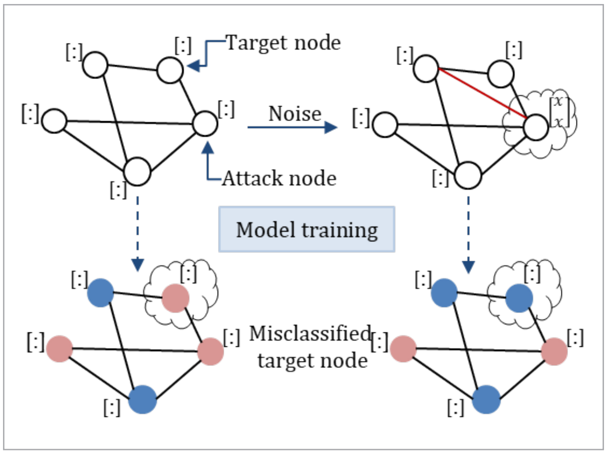
3 **Heterogeneous data handling in FedGNNs.** Fu et al. [28] introduced FedSpray, a novel federated graph learning framework, by incorporating global class structure proxies and a feature-structure encoder. This approach aims to enhance the classification performance of minority class nodes. Zhu et al. [133] innovatively applied topology-aware, data-free knowledge distillation techniques within FedTAD. By generating pseudo-graphs, they strengthened the reliable transfer of knowledge from local models to the global model, thus optimizing the performance of subgraph federated learning. Li et al. [53] proposed AdaFGL, a new paradigm that effectively addresses topological heterogeneity through structural non-IID splitting and a two-step personalized training approach. This method improved the accuracy of federated node classification. These three works are designed to tackle heterogeneous data challenges in federated graph learning scenarios.

As GNN models integrate with federated learning frameworks, the development and application of FedGNNs have progressed swiftly. However, FedGNNs continue to face challenges related to privacy leakage and deployment. Current privacy protection solutions in this domain are primarily categorized into three types: model aggregation, homomorphic encryption, and differential privacy. Model aggregation involves combining participants' model parameters to train a global model, thereby circumventing the transmission of original data during training. Homomorphic encryption allows participants to compute and transmit encrypted data without exposing the original data, including data, model parameters, gradients, and weights. Differential privacy aims to ensure that the outputs of GNN computations are not influenced by changes in specific records. Notably, graph federated learning often employs a combination of these privacy technologies for enhanced protection.

4.4. Adversarial Privacy Preserving

Recent studies [93, 41, 107] have demonstrated that attackers can exploit trained GNN models to extract private information from graph data. To counter such sensitive information leakage, adversarial learning [93] can be employed to create graph data fortified with privacy protection. The fundamental principle of adversarial privacy preservation involves identifying the most effective adversarial examples through sophisticated adversarial attacks (internal maximization optimization), incorporating these examples into the dataset for adversarial training. This enhances the model's expressive power and aids in discovering the optimal adversarial model through a training process focused on minimizing the loss function (external minimization optimization). This approach significantly bolsters the robustness and privacy defense mechanisms of the GNNs. Figure 4 [135] illustrates the overarching framework of graph adversarial attacks.

Figure 4
Graph neural network adversarial attack framework [135]



The adversarial nature of GNNs has been a subject of extensive research. Zhang et al. [121] developed a defense mechanism called GNNGuard, to protect GNNs from various attacks that disrupt graph-structured data during the training phase. The method can be directly incorporated into any GNN model. The primary methodology involves detecting and quantifying graph structures along with the relationship between node features. After quantification, higher weights are assigned to edges connecting similar nodes, while edges between irrelevant nodes are pruned. This process helps in mitigating the negative effects of adversarial attacks. Wang et al. [93] proposed a privacy-preserving graph representation learning framework. The framework tackles two problem scenarios: link prediction with node privacy protection and node classification with link privacy protection. The researchers express these two scenarios through two mutual information objectives and employ variational processing to resolve the issue of calculating the posterior distribution of mutual information items for practical applications. Hsieh et al. [41] proposed a graph perturbation method, NetFence, to protect the privacy of graph data nodes while also countering adversarial attacks on GNNs. This is achieved by modifying the graph structure, i.e., deleting an edge and adding a new one. The perturbed graph aims to reduce the prediction confidence of the private label while maintaining the target label's prediction confidence. This method strikes a balance between reducing privacy leakage risks and maintaining data utility. Xie et al. [107] first used a robust design in graph neural architecture search. The authors propose a method to automatically select the most appropriate defense strategy. They designed a metric for robustness and used evolutionary algorithms along with a single-path one-shot graph framework to search for the most robust architecture. Even under severe poisoning attacks, this approach can achieve state-of-the-art performance.

In summary, a variety of adversarial attack algorithms are employed to identify privacy breaches in GNNs. The existing literature reveals that prevalent defense and protection strategies encompass:

1. Adversarial training [16, 41], which integrates adversarial and clean samples for hybrid training, thereby augmenting the model's resilience against known adversarial attacks;
2. Adversarial perturbation detection [33, 107, 121, 122], which discerns differences between adversarial and clean samples using statistical and geometric features, and subsequently modifies the graph structure based on these findings;
3. Leveraging the attention mechanism [16, 86], this approach trains the model and penalizes adversarial samples to yield a highly robust GNN model. Additionally, purifying the disturbance map before sample involvement in model training [52, 47] is an effective method to mitigate the effects of adversarial attacks. Overall, the primary goal of adversarial learning-based GNN privacy protection is to bolster the model's robustness, thereby safeguarding the privacy of graph data.

4.5. Privacy Defense Capability Evaluation Criteria

Similar to the challenges faced in evaluating privacy defense mechanisms in traditional machine learning, GNNs also suffer from the absence of unified or specialized standards for assessing privacy defense capabilities. Evaluation methodologies vary significantly across research efforts, largely depending on the spe-

Table 6

GNN privacy defense capability evaluation metric

Evaluation metric	Ref.
Accuracy	[75], [93], [135], [16], [33], [54], [35], [119], [110], [15], [40], [50], [65], [8], [77], [68], [103], [29], [28], [133], [53] [41], [107], [121], [122], [52], [47]
Area under curve	[93], [50], [65], [122], [52]
Mean-square error	[47]
Structure similarity	[89], [131]
Privacy-utility	[65], [103]
Mean absolute error	[47], [76]
Relative error	[44], [47]
Root mean square error	[104], [103]
Mean square displacement	[74], [29]
Average attack success rate	[50], [76]
Recall rate	[8], [77]
Macro F1	[120], [65], [8], [77], [52]

cific tasks and application contexts. For instance, in social recommendation systems, tasks such as social, cross-domain, and behavioral recommendations are common, requiring operations like node classification and prediction, link prediction, and clustering. Metrics such as classification accuracy, prediction accuracy, similarity assessment, and class coefficients are employed for evaluation. In the context of adversarial privacy protection for classification tasks, metrics like classification accuracy, average attack success rate, misclassification rate, and mean square error are used for measurement. Table 6 collates and summarizes the evaluation criteria from representative studies on GNN privacy defense.

5. Datasets for Research on Privacy Attacks and Defenses in GNNs

In the realm of GNN privacy defense research, except for a handful of studies that utilize simulated datasets (such as ref. [35, 65, 43]) and proprietary datasets (such as ref. [35]), the majority rely on public datasets (or their subsets). Consequently, this section focuses on the utilization of public datasets.

Table 7 organizes and enumerates the fundamental characteristics of public datasets commonly implemented in GNN privacy defense research, including a tally of their usage in literature. These datasets

Table 7

Commonly used public datasets for GNNs privacy attacks and defenses research.

Dataset	Graphs	Nodes	Edges	Features	Labels	Ref.
Cora	1	2708	5429	1433	7	[75], [93], [135], [16], [102], [33], [38], [89], [15], [50], [8], [68],[133], [53], [41], [107], [122], [47], [76], [97]
Citeseer	1	3327	4732	3703	6	[93], [135], [16], [33], [102], [38], [89], [15], [50], [8], [68], [133], [53], [41], [107], [122], [47], [76], [97]
PubMed	1	19717	44338	500	3	[75], [93], [16], [33], [15], [50], [8], [68], [28], [133], [53], [107], [47]
DBLP	1	4107340	36624464	—	—	[110]
Arxiv	1	169343	295319	128	40	[70], [15], [8]
Facebook	1	—	—	—	—	[75], [50], [77]
LastFM	1	7624	27806	7842	18	[75], [38]
Reddit	1	232965	11606919	602	41	[70], [119], [40], [8], [77], [53], [47]
Orkut	1	3072441	117185083	—	—	[35], [44]
Elliptic	1	203769	234355	166	2	[89], [47], [76]
YouTube	1	1138499	2990443	—	—	[35]
LiveJournal	1	4847571	68993773	—	—	[35]
Amazon	1	8500	48766	767	10	[70], [77]
MovieLens-100K	1	943	100000	1682	5	[104], [120], [103]
MovieLens-1M	1	6040	1000209	3592	5	[104], [103], [47]
MovieLens-10M	1	138493	20000263	27278	5	[104], [103]
Yahoo	1	3000	5335	3000	100	[104], [103]
Douban	1	129490	16830839	58541	5	[104], [103]
Polblogs	1	1490	19025	—	2	[102], [135], [16], [122], [47], [97]
RaFD	67	67	536	—	8	[61]
XM2VTS	295	295	2360	—	8	[61]
CK+	123	593	—	—	7	[61]
Flickr	1	89250	449878	500	7	[104], [89], [35], [8], [103], [28], [53], [76]
Twitch	1	7126	35324	2545	2	[89], [76]
IMDB	1	896308	57064358	428440	2	[110], [44]
Physics	1	495924	34493	8415	8	[119], [28], [133], [53]

originate from various application fields, predominantly encompassing citation network datasets, social network datasets, image datasets, medical datasets, and biochemistry datasets. For instance, Cora, Citeseer, PubMed, DBLP, Arxiv, and PIT are citation network datasets, frequently applied to node classification and link prediction tasks; Facebook, LastFM, Reddit, Orkut, Elliptic, LiveJournal, MovieLens, Yahoo, Douban, and Polblogs are social network datasets, typically used for social recommendation, product recommendation, community discovery, and similar tasks; RaFD, XM2VTS, CK+, Flickr, Twitch, IMDB, and Yale are image datasets; In addition, certain literature makes use of biochemical datasets, such as NCI [38, 87] and OVCAR [38, 101]. Considering space constraints, this section only includes datasets that are commonly used in GNN privacy defense research, but these selections remain representative, as the choice of datasets for attack and defense research primarily aligns with the specific scenarios and tasks of interest.

6. Future Directions

With the ongoing advancements in graph data representation, the availability of GNN model, and training algorithms, GNNs have emerged as a robust and pragmatic tool in graph machine learning. Their development has paved the way for expanded applications in graph data. On this solid foundation, the privacy issues and corresponding attack and defense technologies related to GNNs are increasingly garnering attention. Current research efforts are addressing the privacy concerns of GNNs in various application contexts, yielding promising outcomes and practical implementations. However, a thorough review and analysis of the research on privacy attacks and defenses in GNNs reveals numerous unresolved challenges in this domain. Specifically, future research should focus on the following specific directions.

6.1. Balancing Privacy and Utility of Privacy Protection Mechanisms

Implementing privacy protection in machine learning often comes at the cost of model or data utility. The relationship between privacy and utility indicates that as the degree of privacy defense increases, the utility of the model and/or data decreases. Consequently, striking a balance between privacy protection and utility is crucial.

The complex structure of graph data complicates the measurement of its privacy and utility. Some studies have assessed the privacy and utility of proposed privacy protection schemes [20]. For instance, privacy is measured through factors such as privacy budget and time overhead [67], while utility is evaluated using metrics like accuracy. Literature [67] measures the privacy of graphs by calculating the re-identification rate and assesses utility using statistical indicators such as the number of edges, the average degree of nodes, and the degree variance of nodes in the graph data. However, current privacy quantification methods for GNNs are relatively simple, lacking efficient approaches. Furthermore, the optimal trade-off strategy between privacy and utility remains a key challenge for future privacy protection research.

6.2. Research on GNN Privacy Attack Performance Optimization

There have been numerous studies investigating privacy attacks on GNN models, with the objective of enhancing their robustness and generalization capabilities. Existing research predominantly centers on the exploration of privacy attack methods tailored to specific GNN models under particular assumptions. However, a notable research gap exists in addressing the occurrence of multiple privacy attacks simultaneously. Specifically, research on the performance optimization of privacy attacks based on GNN models can be conducted from the following six aspects: (1) Optimize existing GNN model attack methods by fully leveraging the characteristics of graph sparsity and feature smoothness. (2) Extend privacy attack methods to white box settings: Current privacy attacks are primarily based on black box settings, highlighting the urgent need to explore privacy attack methods that extend to GNN models in white box settings. (3) Explore research on model reuse attacks based on GNNs. (4) Conduct joint attack performance and correlation analysis on the simultaneous occurrence of multiple privacy attacks. (5) Explore defense methods in the case of multiple privacy joint attacks, because existing research predominantly focuses on defense methods against single privacy attacks. (6) Design a comprehensive privacy protection framework [108, 5] to address privacy protection issues across all aspects of GNN model applications, also a prominent topic for future research.

6.3. Research on Dynamic Defense Mechanisms Against Large-Scale Graph Data

Studies on defending against privacy attacks have primarily focused on small-scale graphs. The adaptability of these methods to large-scale graphs, along with their effectiveness and reliability, remains an area that necessitates further exploration. Current research suggests that existing countermeasures are at a significant disadvantage, particularly in the context of adversarial games [45]. Specifically, the majority of GNN adversarial defense algorithms currently in use are passive, static, and empirical, and thus fail to adapt effectively to the dynamic nature of adversarial attack methods. In light of this, future research should prioritize defending against combinations of multiple adversarial privacy attacks, examining the effectiveness and reliability of defense mechanisms against privacy attacks on large-scale graphs in real-world settings, and developing dynamic defense mechanisms to counteract adversarial privacy attacks in GNNs. This will ensure that GNN learning models meet the security and reliability requirements across various application scenarios, even as adversarial attacks continue to evolve.

6.4. Privacy Defense for Personalized Graph Federated Learning

The primary challenges faced by privacy defense in graph federated learning are twofold: (1) The issue of preventing potential data privacy leaks that may occur when local client GNN models share model parameters during the synthesis of the global GNN model. (2) The inability to acquire high-order interaction information between clients due to privacy restrictions when training data participants only contain first-order interactions of local client user data [104]. However, the high-order interaction information is fundamental to the implementation of personalized graph federated learning. As strategies for personalized graph federated learning continue to emerge [16, 117, 85], the design of reasonable model updates that ensure privacy while simultaneously breaking through information isolation to fully utilize high-order interactions for enhancing GNN model learning in personalized scenarios remains a significant challenge.

6.5. Other Research Directions

The five research directions previously mentioned stand as promising areas for future exploration in the sphere of privacy attack and defense within GNN research. Beyond these areas, several other avenues warrant investigation:

1 Beginning at the data processing phase, the design and selection of appropriate preprocessing techniques (such as graph purification) is prevent malicious adversarial or toxic data from participating in model training. This approach could enhance the quality of training data and bolster both data and model privacy.

2 While current research primarily concentrates on preventative measures and controls during incidents, there is a dearth of studies addressing remedial actions following data privacy leaks. Data leaks could precipitate substantial economic losses and potentially severe legal repercussions. Hence, research into post-incident remedial measures is of particular importance.

3 The integration of technologies like secure multi-party computation and blockchain to construct neural network privacy protection in decentralized scenarios is another promising research area.

4 Advanced persistent threat (APT), a long-term and highly concealed attack mode, is prevalent in graph data application fields, including social networks and recommendation systems. Its objective is to steal or misappropriate data. Therefore, examining the APT attack and defense game mechanism in GNNs holds practical significance.

5 Establishing a game model based on the game theory [39], with the purpose of balancing data privacy and utility [113], and combine it with techniques such as reinforcement learning to apply it to GNN models or data privacy defense, is a worthwhile research direction.

6 The exploration of whether privacy defense technology itself may leak privacy or escalate the complexity of privacy protection is also a significant area of research.

7. Conclusions

In this survey, we systematically describe and analyze the latest research achievements in the field of privacy attacks and defense mechanisms for GNNs, thereby filling a gap in the existing literature. Specifically, the survey commences with an introduction to GNNs and their variants, along with the privacy risks they encounter. Subsequently, it offers a comprehensive classification, analysis, and summary of research endeavors within the domain of privacy attacks and defenses in GNNs, encompassing the strengths, weaknesses, commonly utilized datasets, and evaluation methodologies of current studies. Finally, the survey anticipates potential future research directions in this field.

Acknowledgement

This paper is supported by National Natural Science Foundation of China (No. 61972366 and 62366013), and Project for Enhancing Young and Middle-aged Teacher's Research Ability in Colleges of Guangxi (No. 2022KY0702).

Conflicts of Interest

The authors declare that there is no conflict of interest.

References

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., Zhang, L. Deep Learning with Differential Privacy. In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, 2016, 308-318. https://doi.org/10.1145/2976749.2978318
2. Abu-El-Haija, S., Perozzi, B., Kapoor, A., Alipourfard, N., Lerman, K., Harutyunyan, H., Steeg, G. V., Galstyan, A. Mixhop: Higher-order Graph Convolutional Architectures via Sparsified Neighborhood Mixing. In International Conference on Machine Learning, PMLR, 2019, 21-29. https://doi.org/10.48550/arXiv.1905.00067
3. Acharya, J., Sun, Z., Zhang, H. Communication Efficient, Sample Optimal, Linear Time Locally Private Discrete Distribution Estimation. arxiv preprint arxiv:1802.04705, 2018.
4. Acharya, J., Sun, Z., Zhang, H. Hadamard Response: Estimating Distributions Privately, Efficiently, and with Little Communication. In The 22nd International Conference on Artificial Intelligence and Statistics, PMLR, 2019, 1120-1129. https://doi.org/10.48550/arXiv.1802.04705
5. Akter, M., Moustafa, N., Lynar, T., Razzak, I. Edge Intelligence: Federated Learning-Based Privacy Protection Framework for Smart Healthcare Systems. IEEE Journal of Biomedical and Health Informatics, 2022, 26(12), 5805-5816. https://doi.org/10.1109/JBHI.2022.3192648
6. Arachchige, P. C. M., Bertok, P., Khalil, I., Liu, D., Camtepe, S., Atiquzzaman, M. Local Differential Privacy for Deep Learning. IEEE Internet of Things Journal, 2019, 7(7), 5827-5842. https://doi.org/10.1109/JIOT.2019.2952146
7. Atli, B. G., Szyller, S., Juuti, M., Marchal, S., Asokan, N. Extraction of Complex DNN Models: Real Threat or Boogeyman? In Engineering Dependable and Secure Machine Learning Systems: Third International Workshop, EDSMLS 2020, New York City, NY, USA, February 7, 2020, Revised Selected Papers, 2020, 42-57. https://doi.org/10.1007/978-3-030-62144-5_4
8. Ayle, M., Schuchardt, J., Gosch, L., Zügner, D., Günnemann, S. Training differentially private graph neural networks with random walk sampling. arxiv preprint arxiv:2301.00738, 2023. https://doi.org/10.48550/arXiv.2301.00738
9. Backstrom, L., Dwork, C., Kleinberg, J. Wherefore Art Thou R3579X? Anonymized Social Networks, Hidden Patterns, and Structural Steganography. In Proceedings of the 16th International Conference on World Wide Web, 2007, 181-190. https://doi.org/10.1145/1242572.1242598
10. Barak, B., Chaudhuri, K., Dwork, C., Kale, S., McSherry, F., Talwar, K. Privacy, Accuracy, and Consistency Too: A Holistic Solution to Contingency Table Release. In Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, 2007, 273-282. https://doi.org/10.1145/1265530.1265569
11. Bojchevski, A., Günnemann, S. Adversarial Attacks on Node Embeddings via Graph Poisoning. In International Conference on Machine Learning, PMLR, 2019, 695-704. https://doi.org/10.48550/arXiv.1809.01093
12. Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P. Geometric Deep Learning: Going Beyond Euclidean Data. IEEE Signal Processing Mag-

azine, 2017, 34(4), 18-42. https://doi.org/10.1109/MSP.2017.2693418

13. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y. Spectral Networks and Locally Connected Networks on Graphs. arXiv Preprint arXiv:1312.6203, 2013. https://doi.org/10.48550/arXiv.1312.6203
14. Chen, C., Zheng, F., Cui, J., Cao, Y., Liu, G., Wu, J., Zhou, J. Survey and Open Problems in Privacy-Preserving Knowledge Graph: Merging, Query, Representation, Completion, and Applications. International Journal of Machine Learning and Cybernetics, 2024, 1-20. https://doi.org/10.1007/s13042-024-02106-6
15. Chen, C., Zhou, J., Zheng, L., Wu, H., Lyu, L., Wu, J., Liu, Z., Wang, W., Zheng, X. Vertically federated graph neural network for privacy-preserving node classification. arxiv preprint arxiv:2005.11903, 2020. https://doi.org/10.48550/arXiv.2005.11903
16. Chen, J., Huang, G., Zheng, H., Yu, S., Jiang, W., Cui, C. Graph-Fraudster: Adversarial Attacks on Graph Neural Network-Based Vertical Federated Learning. IEEE Transactions on Computational Social Systems, 2022, 10(2), 492-506. https://doi.org/10.1109/TCSS.2022.3161016
17. Chen, L., Li, J., Peng, J., Xie, T., Cao, Z., Xu, K., He, X., Zheng, Z., Wu, B. A Survey of Adversarial Learning on Graphs. arXiv Preprint arXiv:2003.05730, 2020. https://doi.org/10.48550/arXiv.2003.05730
18. Conti, M., Li, J., Picek, S., Xu, J. Label-Only Membership Inference Attack Against Node-Level Graph Neural Networks. In Proceedings of the 15th ACM Workshop on Artificial Intelligence and Security, 2022, 1-12. https://doi.org/10.1145/3560830.3563734
19. Cunha, M., Mendes, R., Vilela, J. P. A Survey of Privacy-Preserving Mechanisms for Heterogeneous Data Types. Computer Science Review, 2021, 41, 100403. https://doi.org/10.1016/j.cosrev.2021.100403
20. Dai, E., Zhao, T., Zhu, H., Xu, J., Guo, Z., Liu, H., Tang, J., Wang, S. A Comprehensive Survey on Trustworthy Graph Neural Networks: Privacy, Robustness, Fairness, and Explainability. arXiv Preprint arXiv:2204.08570, 2022. https://doi.org/10.48550/arXiv.2204.08570
21. Dai, H., Li, H., Tian, T., Huang, X., Wang, L., Zhu, J., Song, L. Adversarial Attack on Graph Structured Data. In International Conference on Machine Learning, PMLR, 2018, 1115-1124. https://doi.org/10.48550/arXiv.1806.02371
22. DeFazio, D., Ramesh, A. Adversarial Model Extraction on Graph Neural Networks. arXiv Preprint arXiv:1912.07721, 2019.
23. Ding, R., Duan, S., Xu, X., Fei, Y. VertexSerum: Poisoning Graph Neural Networks for Link Inference. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, 4532-4541. https://doi.org/10.1109/ICCV51070.2023.00418
24. Duddu, V., Boutet, A., Shejwalkar, V. Quantifying Privacy Leakage in Graph Embedding. In MobiQuitous 2020 - 17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, 2020, 76-85. https://doi.org/10.1145/3448891.3448939
25. Dwork, C., McSherry, F., Nissim, K., Smith, A. Calibrating Noise to Sensitivity in Private Data Analysis. In Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3, 2006, 265-284. https://doi.org/10.1007/11681878_14
26. Fedeli, S., Schain, F., Imtiaz, S., Abbas, Z., Vlassov, V. Privacy Preserving Survival Prediction. In 2021 IEEE International Conference on Big Data (Big Data), IEEE, 2021, 4600-4608. https://doi.org/10.1109/BigData52589.2021.9672036
27. Fout, A., Byrd, J., Shariat, B., Ben-Hur, A. Protein Interface Prediction Using Graph Convolutional Networks. Advances in Neural Information Processing Systems, 2017, 30.
28. Fu, X., Chen, Z., Zhang, B., Chen, C., Li, J. Federated Graph Learning with Structure Proxy Alignment. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, 827-838. https://doi.org/10.1145/3637528.3671717
29. Gauthier, F., Gogineni, V. C., Werner, S., Huang, Y. F., Kuh, A. Personalized Graph Federated Learning with Differential Privacy. IEEE Transactions on Signal and Information Processing over Networks, 2023, 9, 736-749. https://doi.org/10.1109/TSIPN.2023.3325963
30. Gong, X., Wang, Q., Chen, Y., Yang, W., Jiang, X. Model Extraction Attacks and Defenses on Cloud-Based Machine Learning Models. IEEE Communications Magazine, 2020, 58(12), 83-89. https://doi.org/10.1109/MCOM.001.2000196
31. Gu, X., Cho, K., Ha, J. W., Kim, S. Dialogwae: Multimodal Response Generation with Conditional Wasserstein Auto-encoder. arxiv preprint arxiv:1805.12352, 2018. https://doi.org/10.48550/arXiv.1805.12352
32. Guan, F., Zhu, T., Zhou, W., Choo, K. K. R. Graph Neural Networks: A Survey on the Links Between Privacy and Security. Artificial Intelligence Review, 2024, 57(2), 40. https://doi.org/10.1007/s10462-023-10656-4

33. Guohan, H., Zhang, D. GRD-GNN: Graph Reconstruction Defense for Graph Neural Network. Journal of Computer Research and Development, 2021, 58(5), 1075-1091.

34. Hamilton, W., Ying, Z., Leskovec, J. Inductive Representation Learning on Large Graphs. Advances in Neural Information Processing Systems, 2017, 30. https://doi.org/10.48550/arXiv.1706.02216

35. Hay, M., Li, C., Miklau, G., Jensen, D. Accurate Estimation of the Degree Distribution of Private Networks. In 2009 Ninth IEEE International Conference on Data Mining, IEEE, 2009, 169-178. https://doi.org/10.1109/ICDM.2009.11

36. Hay, M., Rastogi, V., Miklau, G., Suciu, D. Boosting the Accuracy of Differentially-Private Histograms through Consistency. arxiv preprint arxiv:0904.0942, 2009. https://doi.org/10.48550/arXiv.0904.0942

37. He, X., Jia, J., Backes, M., Gong, N. Z., Zhang, Y. Stealing Links from Graph Neural Networks. In 30th USENIX Security Symposium (USENIX Security 21), 2021, 2669-2686. https://doi.org/10.48550/arXiv.2005.02131

38. He, X., Wen, R., Wu, Y., Backes, M., Shen, Y., Zhang, Y. Node-Level Membership Inference Attacks Against Graph Neural Networks. arXiv Preprint arXiv:2102.05429, 2022. https://doi.org/10.48550/arXiv.2102.05429

39. Heng, X., Tianqiong, Z., Lefeng, Z. Machine Unlearning: A Survey. In: ACM Comput. Surv., 2023, 56(1), 1-36. https://doi.org/10.1145/3603620

40. Hidano, S., Murakami, T. Degree-preserving Randomized Response for Graph Neural Networks Under Local Differential Privacy. arxiv preprint arxiv:2202.10209, 2022. https://doi.org/10.48550/arXiv.2202.10209

41. Hsieh, I. C., Li, C. T. Netfense: Adversarial Defenses Against Privacy Attacks on Neural Networks for Graph Data. IEEE Transactions on Knowledge and Data Engineering, 2021, 35(1), 796-809. https://doi.org/10.1109/TKDE.2021.3087515

42. Hu, H., Salcic, Z., Sun, L., Dobbie, G., Yu, P. S., Zhang, X. Membership Inference Attacks on Machine Learning: A Survey. ACM Computing Surveys (CSUR), 2022, 54(11s), 1-37. https://doi.org/10.1145/3523273

43. Huang, H., Zhang, D., Wang, K., Zhu, Y., Wang, R. Weighted Large-scale Social Network Data Privacy Protection Method. Journal of Computer Research and Development, 2020, 57(2), 363.

44. Imola, J., Murakami, T., Chaudhuri, K. Locally Differentially Private Analysis of Graph Statistics. In 30th USENIX Security Symposium (USENIX Security 21), 2021, 983-1000.

45. Ji, S. L., Du, T. Y., Li, J. F., Shen, C., Li, B. Security and Privacy of Machine Learning Models: A Survey. Ruan Jian Xue Bao/J Softw, 2021, 32(1), 41-67. https://doi.org/10.13328/j.cnki.jos.006131

46. Jiang, H., Pei, J., Yu, D., Yu, J., Gong, B., Cheng, X. Applications of Differential Privacy in Social Network Analysis: A Survey. IEEE Transactions on Knowledge and Data Engineering, 2021, 35(1), 108-127. https://doi.org/10.1109/TKDE.2021.3073062

47. Jin, W., Ma, Y., Liu, X., Tang, X., Wang, S., Tang, J. Graph Structure Learning for Robust Graph Neural Networks. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2020, 66-74. https://doi.org/10.1145/3394486.3403049

48. Ju, M., Fan, Y., Ye, Y., Zhao, L. Black-Box Node Injection Attack for Graph Neural Networks. arXiv Preprint arXiv:2202.09389, 2022. https://doi.org/10.48550/arXiv.2202.09389

49. Kipf, T. N., Welling, M. Variational Graph Auto-encoders. arxiv preprint arxiv:1611.07308, 2016. https://doi.org/10.48550/arXiv.1611.07308

50. Kolluri, A., Baluta, T., Hooi, B., Saxena, P. LPGNet: Link Private Graph Networks for Node Classification. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, 2022, 1813-1827. https://doi.org/10.1145/3548606.3560705

51. Konecný, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., Bacon, D. Federated Learning: Strategies for Improving Communication Efficiency. arxiv preprint arxiv:1610.05492, 2016, 8. https://doi.org/10.48550/arXiv.1610.05492

52. Li, K., Luo, G., Ye, Y., Li, W., Ji, S., Cai, Z. Adversarial Privacy-Preserving Graph Embedding Against Inference Attack. IEEE Internet of Things Journal, 2020, 8(8), 6904-6915. https://doi.org/10.1109/JIOT.2020.3036583

53. Li, X., Wu, Z., Zhang, W., Sun, H., Li, R. H., Wang, G. AdaFGL: A New Paradigm for Federated Node Classification with Topology Heterogeneity. arxiv preprint arxiv:2401.11750, 2024. https://doi.org/10.48550/arXiv.2401.11750

54. Li, Y., Purcell, M., Rakotoarivelo, T., Smith, D., Ranbaduge, T., Ng, K. S. Private Graph Data Release: A Survey. ACM Computing Surveys, 2023, 55(11), 1-39. https://doi.org/10.1145/3569085

55. Lin, X., Zhou, C., Wu, J., Yang, H., Wang, H., Cao, Y., Wang, B. Exploratory Adversarial Attacks on Graph

Neural Networks for Semi-Supervised Node Classification. Pattern Recognition, 133, 2023, 109042. https://doi.org/10.1016/j.patcog.2022.109042

56. Liu, K., Terzi, E. Towards Identity Anonymization on Graphs. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, 2008, 93-106. https://doi.org/10.1145/1376616.1376629

57. Liu, L., Zhang, J., Song, S. H., Letaief, K. B. Client-edge-cloud Hierarchical Federated Learning. In ICC 2020-2020 IEEE international conference on communications (ICC), 2020, 1-6. https://doi.org/10.1109/ICC40277.2020.9148862

58. Liu, Z., Zhang, X., Chen, C., Lin, S., Li, J. Membership Inference Attacks Against Robust Graph Neural Network. In International Symposium on Cyberspace Safety and Security, Springer, 2022, 259-273. https://doi.org/10.1007/978-3-031-18067-5_19

59. Liu, Z., Zhou, J. Introduction to Graph Neural Networks. Springer Nature, 2022. https://doi.org/10.1007/978-981-16-6054-2_7

60. Majeed, A., Lee, S. Anonymization Techniques for Privacy Preserving Data Publishing: A Comprehensive Survey. IEEE access, 2020,9, 8512-8545. https://doi.org/10.1109/ACCESS.2020.3045700

61. Meden, B., Emeršič, Ž., Štruc, V., Peer, P. k-Same-Net: k-Anonymity with Generative Deep Neural Networks for Face Deidentification. Entropy, 2018, 20(1), 60. https://doi.org/10.3390/e20010060

62. Mironov, I. Rényi Differential Privacy. In 2017 IEEE 30th computer security foundations symposium (CSF), IEEE, 2017, 263-275. https://doi.org/10.1109/CSF.2017.11

63. Mittal, P., Papamanthou, C., Song, D. Preserving Link Privacy in Social Network Based Systems. arXiv Preprint arXiv:1208.6189, 2012. https://doi.org/10.48550/arXiv.1208.618

64. Mudiyanselage, T. B., Lei, X., Senanayake, N., Zhang, Y., Pan, Y. Predicting CircRNA Disease Associations Using Novel Node Classification and Link Prediction Models on Graph Convolutional Networks. Methods, 2022, 198, 32-44. https://doi.org/10.1016/j.ymeth.2021.10.008

65. Mueller, T. T., Paetzold, J. C., Prabhakar, C., Usynin, D., Rueckert, D., Kaissis, G. Differentially Private Graph Classification with Gnns. arxiv preprint arxiv:2202.02575, 2022. https://doi.org/10.48550/arXiv.2202.02575

66. Mueller, T. T., Usynin, D., Paetzold, J. C., Rueckert, D., Kaissis, G. SoK: Differential Privacy on Graph-Structured Data. arxiv preprint arxiv:2203.09205, 2022. https://doi.org/10.48550/arXiv.2203.09205

67. Nguyen, H. H., Imine, A., Rusinowitch, M. Anonymizing Social Graphs via Uncertainty Semantics. In Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, 2015, 495-506. https://doi.org/10.1145/2714576.2714584

68. Ni, X., Xu, X., Lyu, L., Meng, C., Wang, W. A Vertical Federated Learning Framework for Graph Convolutional Network. arxiv preprint arxiv:2106.11593, 2021. https://doi.org/10.48550/arXiv.2106.11593

69. Olatunji, I. E., Hizber, A., Sihlovec, O., Khosla, M. Does Black-Box Attribute Inference Attacks on Graph Neural Networks Constitute Privacy Risk? arXiv Preprint arXiv:2306.00578, 2023. https://doi.org/10.48550/arXiv.2306.00578

70. Olatunji, I. E., Nejdl, W., Khosla, M. Membership Inference Attack on Graph Neural Networks. In 2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), 2021, 11-20. https://doi.org/10.1109/TPSISA52974.2021.00002

71. Olatunji, I. E., Rathee, M., Funke, T., Khosla, M. Private Graph Extraction via Feature Explanations. arXiv Preprint arXiv:2206.14724, 2022. https://doi.org/10.56553/popets-2023-0041

72. Papernot, N., Abadi, M., Erlingsson, U., Goodfellow, I., Talwar, K. Semi-supervised Knowledge Transfer for Deep Learning From Private Training Data. arxiv preprint arxiv:1610.05755, 2016. https://doi.org/10.48550/arXiv.1610.05755

73. Qiu, J., Tang, J., Ma, H., Dong, Y., Wang, K., Tang, J. DeepInf: Social Influence Prediction with Deep Learning. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining, 2018, 2110-2119. https://doi.org/10.1145/3219819.3220077

74. Rizk, E., Sayed, A. H. A Graph Federated Architecture with Privacy Preserving Learning. In 2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), IEEE, 2021, 131-135. https://doi.org/10.1109/SPAWC51858.2021.9593148

75. Sajadmanesh, S., Gatica-Perez, D. Locally Private Graph Neural Networks. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 2021, 2130-2145. https://doi.org/10.1145/3460120.3484565

76. Sajadmanesh, S., Gatica-Perez, D. When Differential Privacy Meets Graph Neural Networks. arxiv preprint

arxiv:2006.05535, 2020. https://doi.org/10.48550/arXiv.2006.05535

77. Sajadmanesh, S., Shamsabadi, A. S., Bellet, A., Gatica-Perez, D. {GAP}: Differentially Private Graph Neural Networks with Aggregation Perturbation. In 32nd USENIX Security Symposium (USENIX Security 23), 2023, 3223-3240. https://doi.org/10.48550/arXiv.2203.00949

78. Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., Monfardini, G. The Graph Neural Network Model. IEEE Transactions on Neural Networks, 2008, 20(1), 61-80. https://doi.org/10.1109/TNN.2008.2005605

79. Scarselli, F., Tsoi, A. C., Gori, M., Hagenbuchner, M. Graphical-Based Learning Environments for Pattern Recognition. In Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshops, SSPR 2004 and SPR 2004, Lisbon, Portugal, 2004, 42-56. https://doi.org/10.1007/978-3-540-27868-9_4

80. Shaheen, M., Farooq, M. S., Umer, T., Kim, B. S. Applications of Federated Learning; Taxonomy, challenges, and Research Trends. Electronics, 2022, 11(4), 670. https://doi.org/10.3390/electronics11040670

81. Sharma, K., Verma, S., Medya, S., Bhattacharya, A., Ranu, S. Task and Model Agnostic Adversarial Attack on Graph Neural Networks. In Proceedings of the AAAI Conference on Artificial Intelligence, 2023, 37(12), 15091-15099. https://doi.org/10.1609/aaai.v37i12.26761

82. Shen, Y., He, X., Han, Y., Zhang, Y. Model Stealing Attacks Against Inductive Graph Neural Networks. In 2022 IEEE Symposium on Security and Privacy (SP), IEEE, 2022, 1175-1192. https://doi.org/10.1109/SP46214.2022.9833607

83. Sun, L., Dou, Y., Yang, C., Zhang, K., Wang, J., Philip, S. Y., He, L., Li, B. Adversarial Attack and Defense on Graph Data: A Survey. IEEE Transactions on Knowledge and Data Engineering, 2022. https://doi.org/10.1109/TKDE.2022.3201243

84. Sweeney, L. k-Anonymity: A Model for Protecting Privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2002, 10(05), 557-570. https://doi.org/10.1142/S0218488502001648

85. Tan, Y., Liu, Y., Long, G., Jiang, J., Lu, Q., Zhang, C. Federated Learning on Non-iid Graphs Via Structural Knowledge Sharing. In Proceedings of the AAAI Conference on Artificial Intelligence, 2023, 37(8), 9953-9961. https://doi.org/10.1609/aaai.v37i8.26187

86. Tang, X., Li, Y., Sun, Y., Yao, H., Mitra, P., Wang, S. Transferring Robustness for Graph Neural Network Against Poisoning Attacks. In Proceedings of the 13th International Conference on Web Search and Data Mining, 2020, 600-608. https://doi.org/10.1145/3336191.3371851

87. Tao, S., Cao, Q., Shen, H., Huang, J., Wu, Y., Cheng, X. Single Node Injection Attack Against Graph Neural Networks. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, 1794-1803. https://doi.org/10.1145/3459637.3482393

88. Thompson, B., Yao, D. The Union-Split Algorithm and Cluster-Based Anonymization of Social Networks. In Proceedings of the 4th International Symposium on Information, Computer, and Communications Security, 2009, 218-227. https://doi.org/10.1145/1533057.1533088

89. Tian, H., Zheng, X., Zhang, X., Zeng, D. D. ∊-k Anonymization and Adversarial Training of Graph Neural Networks for Privacy Preservation in Social Networks. Electronic Commerce Research and Applications, 2021, 50, 101105. https://doi.org/10.1016/j.elerap.2021.101105

90. Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., Ristenpart, T. Stealing Machine Learning Models via Prediction APIs. In 25th USENIX Security Symposium (USENIX Security 16), 2016, 601-618. https://doi.org/10.48550/arXiv.1609.02943

91. Velicković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y. Graph Attention Networks. arXiv Preprint arXiv:1710.10903, 2017. https://doi.org/10.48550/arXiv.1710.10903

92. Vepakomma, P., Gupta, O., Swedish, T., Raskar, R. Split Learning for Health: Distributed Deep Learning without Sharing Raw Patient Data. arxiv preprint arxiv:1812.00564, 2018. https://doi.org/10.48550/arXiv.1812.00564

93. Wang, B., Guo, J., Li, A., Chen, Y., Li, H. Privacy-Preserving Representation Learning on Graphs: A Mutual Information Perspective. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery Data Mining, 2021, 1667-1676. https://doi.org/10.1145/3447548.3467273

94. Wang, J., Luo, M., Suya, F., Li, J., Yang, Z., Zheng, Q. Scalable Attack on Graph Data by Injecting Vicious Nodes. Data Mining and Knowledge Discovery, 2020, 34, 1363-1389. https://doi.org/10.1007/s10618-020-00696-7

95. Wang, X., Wang, W. H. Group Property Inference Attacks Against Graph Neural Networks. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, 2022, 2871-2884. https://doi.org/10.1145/3548606.3560662

96. Wang, Y., Wang, W., Liang, Y., Cai, Y., Hooi, B. Mixup for Node and Graph Classification. In Proceedings of the Web Conference 2021, 2021, 3663-3674. https://doi.org/10.1145/3442381.3449796

97. Wei, J., Yaxin, L., Han, X., Yiqi, W., Jiliang, T. Adversarial Attacks and Defenses on Graphs: A Review and Empirical Study. arxiv preprint arxiv:2003.00653, 2020. https://doi.org/10.48550/arXiv.2003.00653

98. Wu, B., Li, J., Yu, J., Bian, Y., Zhang, H., Chen, C., Houw, C., Fu, G., Chen, L., Xu, T., Rong, Y., Zheng, X., Huang, J., He, R., Wu, B., Sun, G., Cui, P., Zheng, Z., Liu, Z., Zhao, P. A Survey of Trustworthy Graph Learning: Reliability, Explainability, and Privacy Protection. arXiv Preprint arXiv:2205.10014, 2022. https://doi.org/10.1145/3534678.3542597

99. Wu, B., Liang, X., Zhang, S., Xu, R. Advances and Applications in Graph Neural Network. Chinese Journal of Computers, 2022, 45(1), 35-68. https://doi.org/10.11897/SP.J.1016.2022.00035

100. Wu, B., Pan, S., Yuan, X. Towards Extracting Graph Neural Network Models via Prediction Queries (Student Abstract). In Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 15925-15926. https://doi.org/10.1609/aaai.v35i18.17959

101. Wu, B., Yang, X., Pan, S., Yuan, X. Adapting Membership Inference Attacks to GNN for Graph Classification: Approaches and Implications. In 2021 IEEE International Conference on Data Mining (ICDM), 2021, 1421-1426. https://doi.org/10.1109/ICDM51629.2021.00182

102. Wu, B., Yang, X., Pan, S., Yuan, X. Model Extraction Attacks on Graph Neural Networks: Taxonomy and Realisation. In Proceedings of the 2022 ACM On Asia Conference on Computer and Communications Security, 2022, 337-350. https://doi.org/10.1145/3488932.3497753

103. Wu, C., Wu, F., Cao, Y., Huang, Y., Xie, X. Fedgnn: Federated Graph Neural Network for Privacy-Preserving Recommendation. arxiv preprint arxiv:2102.04925, 2021. https://doi.org/10.48550/arXiv.2102.04925

104. Wu, C., Wu, F., Lyu, L., Qi, T., Huang, Y., Xie, X. A Federated Graph Neural Network Framework for Privacy-Preserving Personalization. Nature Communications, 2022, 13. https://doi.org/10.1038/s41467-022-30714-9

105. Wu, X., Fredrikson, M., Jha, S., Naughton, J. F. A Methodology for Formalizing Model-Inversion Attacks. In 2016 IEEE 29th Computer Security Foundations Symposium (CSF), IEEE, 2016, 355-370. https://doi.org/10.1109/CSF.2016.32

106. Xian, X., Hong, M., Ding, J. A Framework for Understanding Model Extraction Attack and Defense. arXiv Preprint arXiv:2206.11480, 2022. https://doi.org/10.48550/arXiv.2206.11480

107. Xie, B., Chang, H., Zhang, Z., Wang, X., Wang, D., Zhang, Z., Ying, R., Zhu, W. Adversarially Robust Neural Architecture Search for Graph Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, 8143-8152. https://doi.org/10.1109/CVPR52729.2023.00787

108. Xiong, J., Ma, R., Chen, L., Tian, Y., Li, Q., Liu, X., Yao, Z. A Personalized Privacy Protection Framework for Mobile Crowdsensing in IoT. IEEE Transactions on Industrial Informatics, 2019, 16(6), 4231-4241. https://doi.org/10.1109/TII.2019.2948068

109. Yang, B., Yih, W. T., He, X., Gao, J., Deng, L. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. arXiv Preprint arXiv:1412.6575, 2014. https://doi.org/10.48550/arXiv.1412.6575

110. Yang, C., Wang, H., Zhang, K., Chen, L., Sun, L. Secure deep graph generation with link differential privacy. arxiv preprint arxiv:2005.00455, 2020. https://doi.org/10.48550/arXiv.2005.00455

111. Yang, S., Doan, B. G., Montague, P., De Vel, O., Abraham, T., Camtepe, S., Kanhere, S. S. Transferable Graph Backdoor Attack. In Proceedings of the 25th International Symposium on Research in Attacks, Intrusions and Defenses, 2022, 321-332. https://doi.org/10.1145/3545948.3545976

112. Yao, L., Mao, C., Luo, Y. Graph Convolutional Networks for Text Classification. In Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33(1), 7370-7377. https://doi.org/10.1609/aaai.v33i01.33017370

113. Ye, D., Zhu, T., Gao, K., Zhou, W. Defending against Label-only Attacks via Meta-Reinforcement Learning. IEEE Transactions on Information Forensics and Security, 2024, 3295-3308. https://doi.org/10.1109/TIFS.2024.3357292

114. Yu, D., Yang, Y., Zhang, R., Wu, Y. Knowledge Embedding Based Graph Convolutional Network. In Proceedings of the Web Conference 2021, 2021, 1619-1628. https://doi.org/10.1145/3442381.3449925

115. Zhang, H., Wu, B., Wang, S., Yang, X., Xue, M., Pan, S., Yuan, X. Demystifying Uneven Vulnerability of Link Stealing Attacks Against Graph Neural Networks. In International Conference on Machine Learning, PMLR, 2023, 41737-41752. https://doi.org/10.56553/popets-2023-0103

116. Zhang, H., Wu, B., Yuan, X., Pan, S., Tong, H., Pei, J. Trustworthy Graph Neural Networks: Aspects, Meth-

ods, and Trends. arXiv Preprint arXiv:2205.07424, 2022. https://doi.org/10.48550/arXiv.2205.07424

117. Zhang, K., Yang, C., Li, X., Sun, L., Yiu, S. M. Subgraph Federated Learning with Missing Neighbor Generation. Advances in Neural Information Processing Systems, 2021, 34, 6671-6682. https://doi.org/10.48550/arXiv.2106.13430

118. Zhang, M., Wang, X., Shi, C., Lyu, L., Yang, T., Du, J. Minimum Topology Attacks for Graph Neural Networks. In Proceedings of the ACM Web Conference 2023, 2023, 630-640. https://doi.org/10.1145/3543507.3583509

119. Zhang, Q., Lee, H.K., Ma, J., Lou, J., Yang, C. Xiong, L. DPAR: Decoupled Graph Neural Networks with Node-Level Differential Privacy. In Proceedings of the ACM on Web Conference 2024, 2024, 1170-1181. https://doi.org/10.1145/3589334.3645531

120. Zhang, S., Yin, H., Chen, T., Huang, Z., Cui, L., Zhang, X. Graph Embedding for Recommendation Against Attribute Inference Attacks. In Proceedings of the Web Conference 2021, 2021, 3002-3014. https://doi.org/10.1145/3442381.3449813

121. Zhang, X., Zitnik, M. Gnnguard: Defending Graph Neural Networks Against Adversarial Attacks. Advances in neural information processing systems, 2020, 33, 9263-9275. https://doi.org/10.48550/arXiv.2006.08149

122. Zhang, Y., Khan, S., Coates, M. Comparing and Detecting Adversarial Attacks for Graph Deep Learning. In Proc. Representation Learning on Graphs and Manifolds Workshop, Int. Conf. Learning Representations, New Orleans, LA, USA, 2019.

123. Zhang, Z., Chen, M., Backes, M., Shen, Y., Zhang, Y. Inference Attacks Against Graph Neural Networks. In 31st USENIX Security Symposium (USENIX Security 22), 2022, 4543-4560. https://doi.org/10.48550/arXiv.2110.02631

124. Zhang, Z., Jia, J., Wang, B., Gong, N. Z. Backdoor Attacks to Graph Neural Networks. In Proceedings of the 26th ACM Symposium on Access Control Models and Technologies, 2021, 15-26. https://doi.org/10.1145/3450569.3463560

125. Zhang, Z., Liu, Q., Huang, Z., Wang, H., Lu, C., Liu, C., Chen, E. GraphMI: Extracting Private Graph Data from Graph Neural Networks. arXiv Preprint arXiv:2106.02820, 2021. https://doi.org/10.24963/ijcai.2021/516

126. Zhao, T., Zhang, X., Wang, S. GraphSMOTE: Imbalanced Node Classification on Graphs with Graph Neural Networks. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining, 2021, 833-841. https://doi.org/10.1145/3437963.3441720

127. Zhao, Y., Chen, J. A Survey on Differential Privacy for Unstructured Data Content. ACM Computing Surveys (CSUR), 2022, 54(10s), 1-28. https://doi.org/10.1145/3490237

128. Zheng, H., Xiong, H., Chen, J., Ma, H., Huang, G. Motif-Backdoor: Rethinking the Backdoor Attack on Graph Neural Networks via Motifs. IEEE Transactions on Computational Social Systems, 2023. https://doi.org/10.1109/TCSS.2023.3267094

129. Zhong, D., Yu, R., Wu, K., Wang, X., Xu, J., Wang, W. H. Disparate Vulnerability in Link Inference Attacks against Graph Neural Networks. Proceedings on Privacy Enhancing Technologies, 2023, 149-169. https://doi.org/10.56553/popets-2023-0103

130. Zhou, J., Han, X., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M. GEAR: Graph-Based Evidence Aggregating and Reasoning for Fact Verification. arXiv Preprint arXiv:1908.01843, 2019. https://doi.org/10.18653/v1/P19-1085

131. Zhou, K., Michalak, T. P., Rahwan, T., Waniek, M., Vorobeychik, Y. Attacking similarity-based link prediction in social networks. arxiv preprint arxiv:1809.08368, 2018. https://doi.org/10.48550/arXiv.1809.08368

132. Zhou, Z., Zhou, C., Li, X., Yao, J., Yao, Q., Han, B. On Strengthening and Defending Graph Reconstruction Attack with Markov Chain Approximation. arXiv Preprint arXiv:2306.09104, 2023. https://doi.org/10.48550/arXiv.2306.09104

133. Zhu, Y., Li, X., Wu, Z., Wu, D., Hu, M., Li, R. H. FedTAD: Topology-aware Data-free Knowledge Distillation for Subgraph Federated Learning. arxiv preprint arxiv:2404.14061, 2024. https://doi.org/10.24963/ijcai.2024/632

134. Zou, X., Zheng, Q., Dong, Y., Guan, X., Kharlamov, E., Lu, J., Tang, J. TDGIA: Effective Injection Attacks on Graph Neural Networks. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, 2461-2471. https://doi.org/10.1145/3447548.3467314

135. Zügner, D., Akbarnejad, A., Günnemann, S. Adversarial Attacks on Neural Networks for Graph Data. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining, 2018, 2847-2856. https://doi.org/10.1145/3219819.3220078

This article is an Open Access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 (CC BY 4.0) License (http://creativecommons.org/licenses/by/4.0/).