GRIFFITH UNIVERSITY

Adapting Membership Inference Attacks to GNN for Graph Classification: Approaches and Implications

Author

Wu, Bang, Yang, Xiangwen, Pan, Shirui, Yuan, Xingliang

Published

2021

Conference Title

2021 IEEE International Conference on Data Mining (ICDM)

Version

Submitted Manuscript (SM)

DOI

10.1109/ICDM51629.2021.00182

Rights statement

© 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Downloaded from

http://hdl.handle.net/10072/416704

Griffith Research Online

https://research-repository.griffith.edu.au

Adapting Membership Inference Attacks to GNN for Graph Classification: Approaches and Implications

Bang Wu
Monash University
Melbourne, Australia
bang.wu@monash.edu

Xiangwen Yang
Monash University
Melbourne, Australia
wayne.yang@monash.edu

Shirui Pan
Monash University
Melbourne, Australia
shirui.pan@monash.edu

Xingliang Yuan
Monash University
Melbourne, Australia
xingliang.yuan@monash.edu

arXiv:2110.08760v1 [cs.LG] 17 Oct 2021

***Abstract*—Graph Neural Networks (GNNs) are widely adopted to analyse non-Euclidean data, such as chemical networks, brain networks, and social networks, modelling complex relationships and interdependency between objects. Recently, Membership Inference Attack (MIA) against GNNs raises severe privacy concerns, where training data can be leaked from trained GNN models. However, prior studies focus on inferring the membership of only the components in a graph, e.g., an individual node or edge. How to infer the membership of an entire graph record is yet to be explored.**

In this paper, we take the first step in MIA against GNNs for graph-level classification. Our objective is to infer whether a graph sample has been used for training a GNN model. We present and implement two types of attacks, i.e., training-based attacks and threshold-based attacks from different adversarial capabilities. We perform comprehensive experiments to evaluate our attacks in seven real-world datasets using five representative GNN models. Both our attacks are shown effective and can achieve high performance, i.e., reaching over 0.7 attack F1 scores in most cases[2]. Furthermore, we analyse the implications behind the MIA against GNNs. Our findings confirm that GNNs can be even more vulnerable to MIA than the models with non-graph structures. And unlike the node-level classifier, MIAs on graph-level classification tasks are more co-related with the overfitting level of GNNs rather than the statistic property of their training graphs.

***Index Terms*—Membership Inference Attacks, Graph Classification, Graph Neural Networks**

I. Introduction

Graph-structured data, with powerful capability presenting wealthy information of real-world data, has been broadly used in various applications such as social networks, medical screening, and chemical property prediction [1], [2]. To analyse rich graph data, deep learning methods have drawn much attention recently. Among them, a family of algorithms, called Graph Neural Networks (GNNs), have achieved state-of-the-art performance by generalising neural networks for graphs [3], [4].

Despite the great power of GNNs, privacy concerns about information exposure in GNNs have been raised in sensitive applications [5], [6]. The graph data for model training is commonly considered as a private property of data owners. For example, chemical and biomedical networks carefully collected from highly consuming experiments are deemed as proprietary assets of companies. In addition, some graph data may contain sensitive attributes, e.g., diagnosis of customers' behaviours. The leakage and abuse of such data may result in serious issues.

Prior studies show that many deep learning methods are vulnerable to a severe privacy attack named Membership Inference Attack (MIA) [5], [7]–[13]. Given access to a model, the attacker can infer whether an arbitrary data record has been used during the training period of this model. As an early study, Shokri *et al.* [13] have demonstrated that the attacker can propose a binary classifier identifying the membership of a record based on its posterior, i.e., the confidence value predicted by the target model. Since the ordinary neural networks often generate different posteriors for the member or non-member inputs, such an attack achieves high success rates and poses significant threats to the privacy of the sensitive training data.

In the domain of GNNs, recent studies [8], [9], [14] start to explore the feasibility of MIA over node-level GNN models. Specifically, node membership inference attacks [14], [15] can infer whether a given node has been used during the training of a target GNN model. Some other inference attacks [8] target at connectives and predict whether a specific pair of nodes are connected in the training graph. Note that those works only infer the membership of a component in the graph. Therefore, how to realise GNN MIAs in graph classification and how an entire graph record can be leaked by the GNNs are yet to be explored.

In this paper, we aim to thoroughly investigate the GNN MIAs in graph-level classification attacks. As aforementioned, the first question is how to realise such attacks in graph classification. In prior inference attacks against node-level GNNs, attackers infer the membership for only a node's attributes or connectivity between two nodes based on the

[1]The short version of this paper has been published in the IEEE International Conference on Data Mining (ICDM) 2021.

[2]The code and data used in the paper are released at https://github.com/TrustworthyGNN/MIA-GNN/

inherent property of graph data. For example, the membership inference of the connectivity between two nodes is based on the similarity of their attributes or predictions from the target model [8]. Therefore, such inference is implemented by utilising the strong correlation between connected nodes, which is common in graph-structure data. We observe that the above intuition does not appear to be extended to graph-level GNN MIAs for inferring the membership of individual graph records.

The second question we aim to tackle is how vulnerable GNN models are to the MIAs. Specifically, *1) What factors and how do they impact the performance of membership inference attacks?* According to previous studies [13], [16] in DNN models, overfitting is the most significant cause for the MIAs, while studies [14], [15] in node-level GNNs show that the graph property is also a significant contributor. In this paper, we will explore how GNN models memorise the training records and perform differently under divergent overfitting levels and classification tasks of various graph data. *2) How is the transferability of the MIAs on GNNs?* Most of the MIAs on DNN are shown to have strong transferability, i.e., the classifier identifying the membership of models can also infer the membership for the model trained with different domain data [13]. The transferring attacks with enhanced generalisability pose larger privacy threats. Therefore, the transferability of MIA on GNNs needs to be investigated.

Contributions. We present the first MIA attacks on graph-level GNN tasks in this work. To address the first question, we propose two types of attacks, i.e., *training-based attacks* and *threshold-based attacks* based on the different capabilities of the attackers. Both attacks can effectively identify the membership of a graph record regarding a target GNN model. Specifically, the attacker issues a query to the target model and receives confidence scores as a response. After that, for the training-based attack, a binary classifier is constructed to infer the membership based on the confidence scores. For the threshold-based attack, the attacker first extracts a confidence value, e.g., the highest confidence score, and then infers the membership by comparing this value to a threshold. Intuitively, since the confidence scores are different for member/non-member inputs, our attacks can identify their memberships in high confidence. We perform a set of experiments to assess our MIAs among different aggregation methods and training data for both our proposed training-based and threshold-based attacks.

To further investigate how vulnerable the GNN models are to the MIAs, we comprehensively measure the effectiveness of our attacks under various experimental settings from two perspectives: GNN methods and training datasets. In particular, we evaluate our attacks against GNN models with different overfitting levels by adjusting the hyperparameters during the training, such as the training epoch and model architecture. Meanwhile, we perform another set of experiments to assess the co-relations between the statistic property of the training graph and the performance of our MIAs. We comprehensively compare our graph-level MIAs to other node-level attacks and analyse the difference and implications behind them.

The contributions of our work are summarised as follows:

- We propose *the first* GNN membership inference attacks for the graph-level classification tasks with black-box access to target models.
- We propose two types of attacks to infer the membership of an arbitrary graph record from learning-based and threshold-based approaches, respectively. We present how to build the training dataset for learning-based attacks and how to choose the confidence value metric for the threshold-based attacks.
- We evaluate our attacks in seven real-world datasets from different domains using five representative GNN methods and conduct extensive evaluations for the transferability of our attacks. We thoroughly analyse the factors which impact the attack performance and reveal the implications of MIAs against GNNs on both node-level and graph-level classification tasks.

Highlights of our key findings. Particularly, we highlight the significant findings corresponding to the aforementioned research questions as follows.

- GNNs are vulnerable to the membership inference attacks and can be even more vulnerable than the ML models with non-graph structures in certain applications.
- Overfitting is the most significant factor for both training-based or threshold-based attacks, which is consistent with the observations of prior MIAs in DNN but different from prior MIAs in node-level GNN models.
- The training-based attacks have strong transferability among multiple GNN types and shadow datasets, while the threshold-based attacks achieve higher attack performance but poorer transferability.

II. Related Work

Recent studies [11]–[13], [16], [17] have shown that attackers can infer the training records of various machine learning models via MIAs, and achieve outperformed attack success rate and precision. Shokri *et al.* [13] first present the MIA against ML models by constructing a binary classifier. The classifier can infer the membership based on the posterior of a record. Leino *et al.* [17] present an effective MIA with high precision under white-box settings. Chen *et al.* [5] extend the MIA to generative models and develop a comprehensive framework for their attacks. Salem *et al.* [12] propose three MIAs for the attacker with different capabilities, and show that their attacks can be broadly applicable at low cost. Christopher *et al.* [10] investigate label-only attacks, which can infer the membership based on only the prediction label of the records. However, all the above-mentioned works show the successful MIAs on the machine learning model trained on Euclidean space. In this paper, we study MIA against graph-level classifiers via graph neural networks (GNNs), which is a family of ML models trained on graph structure data.

There are also several preliminary researches [8], [14], [15] on the MIAs against node-level classifiers. However, these

node-level MIAs only utilise the information of the sub-graph around the target node. For example, He *et al.* [8] derive the membership of a link by the variant distance between the posteriors of only the corresponding two nodes. Their other work [15] and Olatunji *et al.* [14] which identify the membership of a specific node, infer the final results based on the combination of the posterior of the target nodes and its 2-hop neighbour. All of them do not consider the embedding of the entire graph which falls short of showing the vulnerability of GNNs on graph-level. Recently, Zhang *et al.* [18] infer the basic graph properties and the membership of a sub-graph based on graph embedding, but not the membership of the entire graph. Therefore, how graph-level GNN classifiers are vulnerable to MIAs and what is the insights behind them are still yet to be explored.

III. Problem Formalisation

This section presents the formal definition of the proposed MIAs, and introduces the attack assumptions.

A. Problem Definition

Definition 2.1. (Graph Classification Model:) Graph classification aims to predict a categorical class for a given graph. Specifically, with a set of graphs $G = \{g_1, g_2, \cdots, g_n\}$ where each individual g_i has a class y_i in a set of label Y, a graph classification model is a mapping $f_\theta(.) : g_i \rightarrow y_i$ which infers the label of the graph.

GNN based graph classification is a fundamental analytics task and serves many applications such as fraud detection [19]–[22], malware detection [23]–[26], and healthcare [27]. It has attracted increasing attention due to its high prediction accuracy, and has become the standard toolkit for analysing and learning from data on graphs. Here, we define the graph classification model as the victim model and target its training data membership's confidentiality. The training dataset $G = \{(g_1, y_1), (g_2, y_2), ..., (g_n, y_n)\}$ is a set of labeled graphs with ground truth labels. The objective function of the graph classifier can be formulated as:

$$\min_{\theta} \sum_{g_i \in G, y_i \in Y} L(f_\theta(g_i), y_i), \tag{1}$$

where $L(.)$ is the loss function of the graph prediction.

Definition 2.2. (Membership Inference Attacks in GNNs:) Given a GNN model $f_\theta(.)$, it is trained on a labeled graph set $G_{Train} = (G, Y)$ for a graph-level classification task. The membership inference attack attempts to infer whether a specific graph g_i is in the target training dataset G. Formally,

$$\mathcal{A} : (g_i, f_\theta(.)) \rightarrow \{0, 1\}, \tag{2}$$

where $\mathcal{A}$ represents the attack model which outputs 1 if the record g_i is in the training set of $f_\theta(.)$, and outputs 0 otherwise.

B. Attack Assumptions

This section introduces the detailed attack scenarios by explaining the attacker's capability and background knowledge.

Black-box settings. We assume that the attacker can only obtain black-box access to the target models, which is the most common and realistic setting for the adversarial knowledge [7], [28]. In the black-box setting, the attacker has access to neither the parameters of the target models nor the internal representations during the inference. Only the model queries (model outputs of a chosen input) are accessible. Specifically, the well-trained models are normally deployed in the MLaaS, which only provides model queries to other users through an API. The black-box attacker attempts to exploit the difference between the prediction of their chosen queries and infer the membership of the records.

Attacker's background knowledge. Even though the attacker has only black-box access to the model parameters and internal representations, he can manage to obtain some public information.

- **Shadow dataset.** A Shadow dataset can be a dataset with the same domain as the target model training dataset. Since the target model's task is commonly public knowledge, the attacker can gather data and build the shadow dataset. In this paper, we also relax this assumption in the case that the shadow dataset comes from a different domain.
- **Training knowledge.** We assume the attacker knows the type of GNNs and how a GNN was trained (i.e., the training algorithm). This assumption is reasonable since many packages such as Deep Graph Library (DGL) [29] provides implementation of GNN models and the assumption is consistent with prior MIAs [12], [13]. We will further relax this assumption and discuss our attacks when the attacker does not know the type of GNNs.

IV. The Proposed Attacks

Fundamentally, MIAs exploit different responses between the member and non-member inputs. As discussed in Section III, in the black-box settings, the attacker can only obtain the responses, and the intuition of applying MIAs is to analyse the responses from the target model and identify the membership. There are two main methods of MIAs: *training-based* and *non-training-based* approaches. Here, we design our attacks from these two approaches, respectively.

A. Training-based Attacks

Overviews. The idea of training-based attacks is to train an attack model to classify the membership of the input. This attack model will be a binary classifier whose output is "in" or "out" corresponding to whether one record has been or not been used during the target model training. Therefore, we first propose to construct a training dataset for the attack model. Meanwhile, supervised training is adopted to obtain a more accurate binary classifier, as it is better to build a training dataset whose records have been labelled as "in" or "out". With only black-box access to the target model, the attacker has no idea about the target model's training dataset, as he cannot get the membership label of a record. To construct the attack model, the attacker needs to build a surrogate model based on the background knowledge.

Algorithm 1 Algorithm for Attack Model Training

Input:
Shadow Dataset G_s
Output:
Attack Model $f_{attack}(.)$.

Split G_s to G_{member} and G_{non_member}
Train Shadow Model $f_{shadow}(.)$ on G_{member}
$V_{attack} = \emptyset$
for g_i in G_s **do**
    if g_i in G_{member} **then**
        $V_{attack} = V_{attack} \cup \{(f_{shadow}(g_i), \text{“}in\text{”})\}$
    if g_i in G_{non_member} **then**
        $V_{attack} = V_{attack} \cup \{(f_{shadow}(g_i), \text{“}out\text{”})\}$
Train Attack Model $f_{attack}(.)$ on V_{attack}

Construction. Algorithm 1 shows how to construct the attack model and the detailed steps are listed as followed:

- *Step 1. Shadow Datasets Processing.* The attacker gathers the dataset and splits it into two parts. One for shadow model training is the shadow model's membership set, and the other one is the non-member set.
- *Step 2. Shadow Model Training.* The attacker generates a surrogate model to mimic the prediction behaviour of the target model. Only the membership set constructed in the first step is used to train the model. Since the shadow dataset is of the same or similar domain as the target model, it is expected to behave similarly to the target model.
- *Step 3. Confidence Scores Gathering.* The attacker then feeds both membership and non-membership datasets into the shadow model and obtains their confidence scores. These confidence scores can be labelled as “in” or “out” corresponding to the dataset they are from.
- *Step 4. Attack Model Training.* The attacker finally trains the attack model to identify the membership of a record based on its confidence score. The training data for the attack model is the confidence score set generated in step 3.

To apply our attack on an arbitrary record, the attacker first issues a query to the target model and obtains the confidence score of the record. Then the attack model can infer the membership of the record based on its confidence score. In general, if the shadow model performs similarly to the target, the attack model can correctly identify the record's membership.

Remarks. If the shadow and target datasets are in the same domain, the confidence scores for them will have equal dimensions. However, in reality, the attacker may not have knowledge of the data domain. Thus, the obtained datasets may come from other domains which result in different class numbers, i.e., the dimensions of confidence scores are different to the target one. To address the issue, we propose to reduce the confidence scores with higher dimensions to be consistent with the others. Only the most significant values are kept during the reduction. That is, we select the top k values in terms of the confidence scores and construct the new confidence score with k dimensions.

Metrics		Definition
Highest confidence score		$max(x_1, x_2, ..., x_n)$
Loss	MSE	$\sqrt{\sum_i (x_i - y_i)}$
	Cross-Entropy	$-log(\frac{exp(max(x_i))}{\sum_j exp(x_j)})$
Distance	Cityblock	$\sum_i \lvert x_i - y_i \rvert$
	Canberra	$\sum_i \frac{\lvert x_i - y_i \rvert}{\lvert x_i + y_i \rvert}$

TABLE I: Confidence metrics. x_i represents the i-th component of confidence vector x. y_i represents the i-th component of a one-hot vector y corresponding to the predicted label.

B. Threshold-based Attacks

We then present how to apply the attacks without training the attack models. In some applications, the attacker may not have appropriate resources for expensive model training during membership inference. Instead, a non-training based approach with a much lower cost can be applied.

To identify the membership of a record only based on the output posteriors, the idea is that the model is more confident with the inputs it memorises. Thus, the input with comparably higher confidence can be considered as membership. To evaluate how the model is confident with the input, we propose to employ three different evaluation metrics. Detailed information is shown in Table I.

Confidence via confidence score. The label with the highest confidence score is considered the final prediction label. A higher score represents higher confidence in the model regarding the inference. Therefore, it is common to use the highest confidence score to represent how the model is confident with the input.

Confidence via loss. It is also reasonable to use the loss function as a metric for the confidence of the records. In particular, since the model is trained to reduce the loss of the training data, the loss of the memberships is expected to be lower than the non-memberships. Therefore, we calculate the loss of a record, and a smaller loss value means higher confidence from the target models.

In the proposed attacks, two popular loss functions, Mean Squared Error (MSE) and Cross-entropy are used as the confidence metrics. Note that, the calculation of the loss during the training requires the ground truth labels of the data. In our attack settings, the attacker cannot obtain these ground truth labels. Nevertheless, since the target models are often well-trained, we can use the predictions as their labels. So, we can still calculate the loss values and apply the threshold-based attacks.

Confidence via distance. Besides using some popular loss functions, we propose to measure the confidence via distance. Similar to the calculation of loss, we evaluate the distance between the confidence vectors and the one-hot output vectors

Dataset	#Graphs	#Classes	Avg. Nodes	Avg. Edges
PROTEIN_full	1113	2	39.06	62.14
DD	1178	2	284.32	715.66
ENZYMES	600	6	32.63	62.14
CIFAR10	60K	10	117.63	941.07
MNIST	70K	10	70.57	564.53
OGBG-PPA	158K	37	243.4	2266.4
NCI	4110	2	29.87	32.30

TABLE II: Summary statistics of seven datasets.

corresponding to the ground truth labels. Since some distances are positively related to the previous metrics, e.g., MSE is the same as the Euclidean distance, we select Cityblock distance (also called Manhattan distance) and Canberra distance as our confidence metrics.

Threshold selection. Given the confidence values, the attacker can infer the membership via a threshold value. The threshold value can be chosen by considering the requirements of the attacks as prior work [12]. Specifically, the higher threshold can be selected when the attacker focuses more on precision. In contrast, he can choose a lower threshold if he concentrates more on recall.

V. Evaluation and Analysis

Our evaluation aims at answering the following research questions:

RQ1 *Whether and how different types of GNNs can be vulnerable to MIAs?*

RQ2 *How is the transferability of MIAs? how is the attack performance affected if the attacker obtains less knowledge of the target model?*

RQ3 *What are the factors that impact the performance of MIAs? Is the overfitting only factor that affects the performance of attacks in GNNs?*

RQ4 *Does the target model performance (e.g. overfitting level) or the target dataset property (e.g. statistics of the graph) affect the attack performance more? What is the difference between the MIAs on graph-level classification tasks and other graph-based MIAs?*

To answer RQ1, we apply our attacks on various types of GNN models trained by different graph data, and then compare them with the baseline attacks on MLP. To answer RQ2, we apply the attacks with different adversarial background knowledge and evaluate their effects. To answer RQ3, we adjust several parameters to analyse their impacts on attack performance. To answer RQ4, we discuss the empirical results under different settings and compare our observations to prior studies.

In this section, we first introduce the experimental settings for our attacks. Then, we present the results for both learning-based and threshold-based attacks regarding the above research questions and further discuss and compare our observations with others.

A. Experimental Setup

To obtain a comprehensive insight into how GNNs are vulnerable to the MIAs, we evaluate the attacks among various types of GNNs trained by multiple different datasets.

Datasets. We use seven real-world datasets from different domains to evaluate our attacks. Specifically, PROTEIN_full [30], DD [31], ENZYMES [30] and OGBG-PPA [32] are all protein graph datasets. CIFAR10 [33] and MNIST [34] are classical image classification datasets converted into graphs using super-pixels [35]. The graphs in NCI [36] represent the atoms and chemical bond connections of the chemical compounds. The statistics of the datasets is given in Table II.

Here, we evaluate our attacks by considering the worst-case scenario for the attacker, where no same data in the shadow dataset is compared to the target one. In particular, we equally split them into two equal parts: one is the target models' training dataset, while the other is the shadow dataset obtained by the attackers. Note that in practice, it is possible for the attacker to obtain a shadow dataset overlapping with the target dataset. In this case, the shadow model will be more similar to the target model and benefit our attacks. The purpose of this setting is to show that even in the worst case, our attacks are quite powerful and can still achieve high performance in membership inference.

GNN types. We evaluate five popular GNNs using these datasets. For PROTEIN_full, DD, ENZYMES, we use GCN [37], GateGCN [38], GIN [39], GAT [40] as the target models. And for CIFAR10 and MNIST, we use the above four GNNs as well as GraphSAGE [41] for the target models. As a common setting in practice, all of these models contain only two layers. To investigate the vulnerability of deeper GNNs with more parameters, we further evaluate DeepGCN [42] in OGBG-PPA and NCI. We adjust the architectures of these deeper GNNs and demonstrate their impacts. We also use MLP as a non-graph-structure baseline model as [30] to help evaluate our attacks. It simply updates the representation for each node independently without considering their neighbours. Other parameters, such as the number of the model layers, are set to be the same as GNNs.

Evaluation metrics. Because membership inference is a binary classification problem, in our experiments, we adopt the attack precision, recall and F1 score as our evaluation metrics to measure the overall performance of our attacks, which are consistent with prior work on MIAs [13], [16]. We run all our experiments 15 times.

B. Attack Performance

1) Performance Overview: We first show how different types of GNN models are vulnerable to both our training-based and threshold-based attacks on different datasets.

Findings #1: *Several popular types of GNN models are all vulnerable to the MIAs on different training data.*

Dataset	Model	Training Accuracy	Testing Accuracy	Train-test Gap	Attack Precision	Attack Recall	Attack F1 Score
PROTEIN_ful	GateGCN	0.990	0.710	0.280	0.602(0.070)	0.588(0.058)	0.595(0.063)
	GCN	1.000	0.688	0.313	**0.745(0.055)**	**0.607(0.046)**	**0.668(0.043)**
	GIN	0.730	0.690	0.040	0.563(0.078)	0.559(0.074)	0.561(0.075)
	GAT	1.000	0.660	0.340	0.678(0.094)	0.590(0.047)	0.630(0.066)
	MLP(baseline)	0.990	0.660	0.340	0.691(0.086)	0.602(0.057)	0.642(0.065)
DD	GateGCN	1.000	0.630	0.370	**0.890(0.029)**	**0.881(0.035)**	**0.885(0.030)**
	GCN	1.000	0.630	0.370	0.793(0.065)	0.682(0.059)	0.733(0.057)
	GIN	1.000	0.610	0.390	0.600(0.050)	0.594(0.048)	0.597(0.051)
	GAT	1.000	0.670	0.330	0.831(0.035)	0.758(0.043)	0.792(0.036)
	MLP(baseline)	0.780	0.650	0.130	0.553(0.054)	0.551(0.054)	0.552(0.054)
ENZYMES	GateGCN	1.000	0.550	0.450	**0.865(0.083)**	0.715(0.072)	**0.782(0.072)**
	GCN	1.000	0.520	0.480	0.843(0.101)	0.725(0.088)	0.778(0.088)
	GIN	0.940	0.480	0.460	0.593(0.071)	0.592(0.063)	0.592(0.067)
	GAT	1.000	0.530	0.470	0.817(0.044)	**0.736(0.051)**	0.774(0.043)
	MLP(baseline)	1.000	0.380	0.620	0.821(0.072)	0.625(0.063)	0.707(0.058)
CIFAR10	GateGCN	1.000	0.476	0.524	0.877(0.011)	**0.843(0.009)**	0.859(0.010)
	GCN	0.995	0.363	0.632	0.760(0.014)	0.748(0.012)	0.754(0.013)
	GIN	0.984	0.319	0.666	0.601(0.013)	0.599(0.013)	0.600(0.013)
	GAT	0.989	0.416	0.573	0.693(0.013)	0.672(0.012)	0.682(0.012)
	GraphSAGE	1.000	0.494	0.506	**0.881(0.009)**	0.841(0.007)	**0.860(0.008)**
	MLP(baseline)	1.000	0.354	0.646	0.768(0.013)	0.680(0.009)	0.721(0.010)
MNIST	GateGCN	1.000	0.918	0.082	0.673(0.068)	0.585(0.009)	0.625(0.031)
	GCN	1.000	0.805	0.195	**0.803(0.010)**	**0.720(0.007)**	**0.759(0.008)**
	GIN	0.984	0.755	0.228	0.547(0.010)	0.546(0.010)	0.547(0.010)
	GAT	1.000	0.888	0.112	0.709(0.071)	0.596(0.009)	0.646(0.033)
	GraphSAGE	1.000	0.910	0.090	0.762(0.027)	0.636(0.005)	0.693(0.013)
	MLP(baseline)	1.000	0.856	0.144	0.666(0.058)	0.567(0.009)	0.612(0.028)
OGBG_PPA	DeepGCN	1.000	0.588	0.412	0.843(0.014)	0.810(0.010)	0.826(0.011)
NCI	GCN	1.000	0.622	0.378	0.689(0.038)	0.613(0.024)	0.649(0.029)

TABLE III: Accuracy of the target models for different datasets and the corresponding performance on training-based attacks.

Training-based attacks. Table III shows the attack performance of our training-based on seven datasets with different GNN methods. For most of the datasets, our attacks can achieve more than 65% attack precision, which confirms our attacks are effective in various GNN models on different tasks. Note that, for some of the datasets, such as DD and CIFAR10, our attacks achieve more than 0.85 F1 Score, which means the target models are highly vulnerable to our proposed training-based MIAs. The results also depict that different GNN methods may have different levels of robustness against MIAs. It is observed that GateGCN and GCN are more vulnerable to inference attacks while GIN is more robust. It might be due to the different aggregation and transformation strategies used by these models. GIN aggregated via MLP can be more generalised to different input records, which increase the difficulty of membership inference.

Impact of attack strategies. We evaluate how the size of the input vector affects our attacks. Section IV-B mentions that the dimensions of confidence vectors may be different for a cross-domain attack, so we distil them to vectors with lower dimensions. Table V shows the comparison among the attacks on OGBG_PPA, whose class is 37, with different sizes of confidence vectors used for attack models. It can be found that distilled vectors with 10 dimensions achieve higher attack rates compared to the others. Therefore, appropriately distilling the confidence vectors for attack model training can improve the attack performance.

Threshold-based attacks. Table IV demonstrates the attack performance of our threshold-based MIAs based on different matrix thresholds. Comparing training-based attacks, the threshold-based attacks are only effective via some of the thresholds while performing poorly via others.

Impact of the confidence metrics selection. We investigate how the confidence metrics affect our attacks. From the results in Table IV, it is obvious that only the attacks using cross-entropy as the metric achieve satisfying performance. Other attacks perform much worse in most of the attack settings. The reason behind it is that only using the metrics for target model training can achieve high performance. As mentioned in Section IV, the intuition of our design is that the models will have higher confidence values for the members. However, based on our observation, this phenomenon only happens when the confidence value is measured by its loss function. So the attacks may perform badly when using other metrics.

Dataset	Model	Highest Confidence Score	Cross-Entropy	MSE	Cityblock	Canberra
PROTEIN_ful	GateGCN	0.501(0.070)	**0.678(0.021)**	0.326(0.029)	0.400(0.061)	0.399(0.061)
	GCN	0.533(0.148)	**0.690(0.028)**	0.232(0.019)	0.263(0.060)	0.263(0.049)
	GIN	0.297(0.010)	**0.665(0.005)**	0.107(0.001)	0.144(0.009)	0.144(0.008)
	GAT	0.544(0.026)	**0.677(0.028)**	0.284(0.065)	0.241(0.055)	0.284(0.065)
	MLP	0.544(0.012)	**0.701(0.027)**	0.292(0.020)	0.285(0.098)	0.285(0.084)
DD	GateGCN	0.406(0.340)	**0.845(0.049)**	0.366(0.017)	0.426(0.030)	0.426(0.079)
	GCN	0.666(0.006)	**0.771(0.001)**	0.209(0.027)	0.191(0.063)	0.191(0.045)
	GIN	0.044(0.009)	**0.673(0.002)**	0.000(0.000)	0.015(0.006)	0.015(0.005)
	GAT	0.724(0.007)	**0.793(0.007)**	0.584(0.002)	0.194(0.004)	0.194(0.006)
	MLP	0.046(0.029)	0.533(0.014)	0.507(0.011)	**0.598(0.009)**	**0.598(0.008)**
ENZYMES	GateGCN	0.844(0.052)	**0.848(0.038)**	0.581(0.034)	0.598(0.075)	0.598(0.070)
	GCN	0.812(0.030)	**0.854(0.019)**	0.578(0.009)	0.684(0.000)	0.684(0.000)
	GIN	0.011(0.015)	**0.668(0.004)**	0.590(0.024)	0.579(0.009)	0.579(0.036)
	GAT	0.841(0.007)	**0.854(0.009)**	0.584(0.002)	0.682(0.004)	0.682(0.003)
	MLP	0.715(0.034)	**0.807(0.011)**	0.576(0.027)	0.581(0.008)	0.581(0.006)
CIFAR10	GateGCN	0.844(0.011)	**0.860(0.007)**	0.542(0.0003)	0.561(0.013)	0.561(0.011)
	GCN	0.688(0.035)	**0.759(0.012)**	0.547(0.005)	0.554(0.013)	0.554(0.015)
	GIN	0.421(0.221)	**0.757(0.176)**	0.544(0.016)	0.567(0.027)	0.567(0.023)
	GAT	0.314(0.161)	**0.581(0.050)**	0.550(0.007)	0.573(0.020)	0.573(0.026)
	GraphSAGE	0.697(0.154)	**0.755(0.107)**	0.552(0.005)	0.556(0.001)	0.556(0.001)
	MLP	0.482(0.125)	**0.661(0.064)**	0.542(0.002)	0.544(0.008)	0.544(0.011)
MNIST	GateGCN	0.445(0.030)	**0.712(0.008)**	0.539(0.005)	0.575(0.005)	0.575(0.004)
	GCN	0.551(0.077)	**0.662(0.075)**	0.546(0.01)	0.545(0.010)	0.545(0.025)
	GIN	0.120(0.023)	**0.674(0.002)**	0.541(0.001)	0.587(0.003)	0.587(0.009)
	GAT	0.404(0.042)	**0.719(0.006)**	0.542(0.002)	0.577(0.004)	0.577(0.009)
	GraphSAGE	0.517(0.023)	**0.737(0.006)**	0.544(0.001)	0.593(0.001)	0.583(0.000)
	MLP	0.484(0.024)	**0.671(0.034)**	0.541(0.001)	0.580(0.001)	0.580(0.005)
OGBG_PPA	DeepGCN	0.717(0.152)	**0.796(0.088)**	0.526(0.005)	0.578(0.007)	0.577(0.008)

TABLE IV: F1 score of the threshold-based attacks targeting different models for six datasets. The results that are higher than training-based attacks are underlined. The best results are bold.

Model Architecture	Train-test Gap	Ordinary Vectors	Top-10 Vectors	Top-3 Vectors
28-layer	0.412	0.826(0.011)	**0.839(0.009)**	0.835(0.012)
22-layer	0.484	0.858(0.009)	0.864(0.011)	**0.865(0.009)**
20-layer	0.480	0.860(0.013)	**0.873(0.012)**	0.867(0.008)
18-layer	0.490	**0.864(0.010)**	0.862(0.012)	0.859(0.010)
16-layer	0.510	0.852(0.012)	**0.863(0.012)**	0.861(0.011)
12-layer	0.500	0.838(0.010)	0.841(0.007)	**0.844(0.011)**

TABLE V: F1 scores comparison among attack models with different input vectors. The best results are highlighted.

2) Comparison to Non-graph-structure Model: We now compare our attacks on GNNs with the MLP baseline as [30].

Findings #2: *Comparing with MLP, graph-level GNN classification models are even more vulnerable to the MIAs in certain applications.*

We consider the MLP baseline as the non-graph-structure ML method which simply updates the node independent as $h_i^{l+1} = \sigma(W^l h_i^l)$. From the results, our attacks reach better performance on GNNs than MLP for most of the datasets. Table III highlights the attacks with the best performance. It is shown that all the MLP models are more robust than GateGCN and GCN models; namely, some types of GNNs are more vulnerable to MIAs than MLPs. For example, our attacks targeting at GateGCN model trained on DD has a 0.885 F1 score, which is about 0.333 higher than 0.552 for the MLP model.

3) Transferability: Finally, we explore the transferability of our attacks by relaxing the assumptions of the attacker's background knowledge about the target models and shadow

dataset.

Findings #3: *Our MIAs have strong transferability. The attacks are still effective with the shadow models trained as different GNN types and different datasets, or without training the shadow models.*

Training-based attacks. As discussed in Section III, the attacker is assumed to have some knowledge of the GNN methods of the target models. However, in real-world scenarios, he may not know what types of GNNs are used for training the target models. As a result, the transferability of our attacks when mismatching the GNN methods needs to be investigated. Figure 1 reports the confusion matrix of the attack performance for the targeted/shadow model using different GNN methods. It can be found that, most of our transferring attacks are effective with a reduction of F1 score within 0.1.

Besides the knowledge about the GNN methods, another important knowledge is the shadow dataset. In practice, it might be difficult for the attacker to gather shadow datasets in the same domain as the target. So, we evaluate our attacks' transferability in the cross-domain of shadow datasets. Figure 2 shows the confusion matrix of the attack performance for the targeted/shadow model trained on different datasets. It can be found that our transferring attacks among cross-domain shadow datasets are still effective. As a result, the knowledge of the shadow dataset affects less in our attacks and our training-based MIAs are shown to have strong transferability.

Threshold-based attacks. Different from the training-based attacks, the threshold-based attacks do not need to train shadow models. As a result, the GNN methods of the target model do not affect the attacks and we only consider the transferability of our attacks for the cross-domain settings. The attackers should properly select the thresholds for their attacks based on the threshold distribution, which can be obtained by querying the target model with records in the same or similar domain. In practice, the optimal thresholds vary for data from different domains which places a higher demand on the knowledge of the attackers. We observe that the mismatching of the dataset domain can lead to a dramatic reduction of the attack effectiveness for the transferring attacks in cross-domain. As a result, the transferability of the threshold-based attacks is poor and it is hard for the attacker to apply the attacks without the same domain knowledge.

4) Comparison between Two Proposed Attacks: After confirming both our training-based attacks and the threshold-based attacks can be effective when inferring the membership of the GNNs, we summarise the advantages and disadvantages for both of them.

Findings #4: *The threshold-based MIAs can achieve even better attack performance but much less transferability compared to the training-based attacks.*

Firstly, the threshold-based attacks achieve the highest attack performance in most of the attack settings. In addition, there is no GNN method or dataset which is significantly more robust than others under the threshold-based attacks, while we observed that GIN is less vulnerable compared with training-based attacks against other types of GNNs. Furthermore, the threshold-based attacks require fewer computation resources since no attack models need to be trained during the attacks. However, the threshold-based attacks also have limitations. The selection of the thresholds is non-trivial, which may significantly affect the attack performance. Moreover, the transferability of the threshold-based attacks is much poorer than the training-based attacks, as the selection for the confidence metric is critical.

C. Factors Affecting Attack Performance

We have shown the effectiveness of our attacks under different attack settings. To fully understand the implications behind them, we further analyse our MIAs by adjusting several potential factors and discussing their effects on the attack performance.

1) Effect of the Target Model Property: We first discuss how the target model property can affect the attack performance.

Findings #5: *Overfitting is the most significant factor that affects the MIA performance on graph-level classification tasks.*

Impact of the overfitting. Similar to the prior works [13], [16], we first analyse the relationship between the overfitting level of the target GNN models and the attack performance. Overfitting represents how the model treats the input differently for members or non-member. Intuitively, the more difference the GNN models treat between the member and non-member input records, the easier our attack can be applied. Based on our experiments, it is also a significant factor that impacts our attack performance.

We measure the overfitting level by the train-test gap of the target model as previous MIA works [13], [16]. Figure 3 shows how the train-test gap and the F1 score change when increasing the training epoch. Generally, after more epoch training, the target model becomes overfit and the train-test gap increases. Accordingly, the F1 scores of the attack also increase significantly which indicates that GNN models with higher overfitting levels are more vulnerable to the MIAs.

We evaluate how overfitting affects threshold-based attacks. Figure 4 illustrates the target model train-test gap and the attack performance for increasing training epochs. The results indicate that the threshold-based attacks are also positively correlated to the overfitting. In addition, comparing the training-

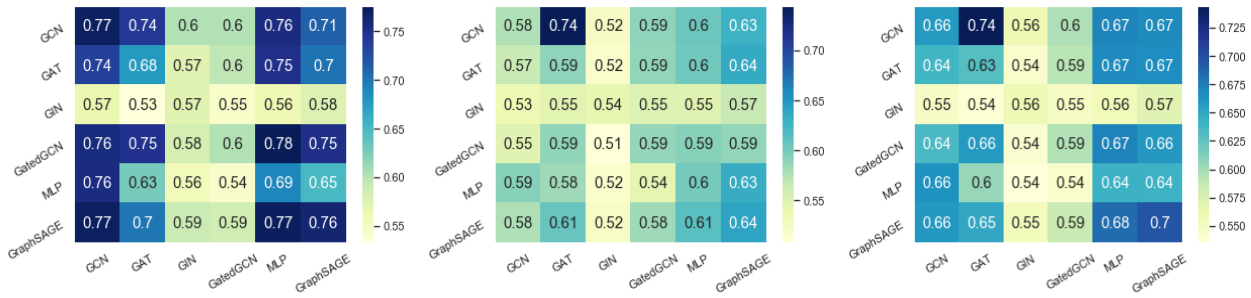


Fig. 1: Confusion matrix of the attack performance for using different GNN methods (left is the attack precision, the middle is the attack recall and right is the F1 score).

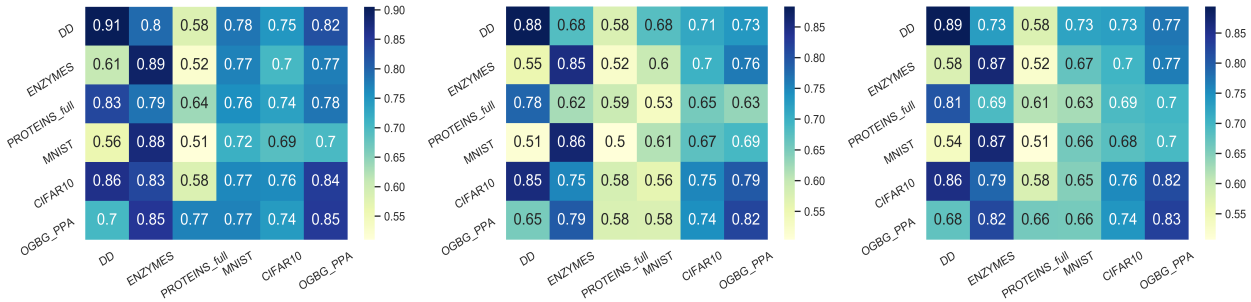


Fig. 2: Confusion matrix of the attack performance for using different shadow datasets (left is the attack precision, the middle is the attack recall and right is the F1 score).

			Target Graph Data Property			Target GNN Model Property	
			#Nodes	#Edges	Graph Density	#Classes	Train-Test Gap
Graph-level	Training-based	GCN	-0.0296	-0.0256	-0.0155	0.8420	0.8110
		MLP	-0.1071	-0.1125	0.1038	0.4748	0.5562
	Threshold-based	GCN	0.0235	0.0268	0.0687	0.2385	-
		MLP	0.0001	0.0099	0.0846	0.3662	-
Node-level	[14]	GCN	-	-	-	0.3857	-0.2524
	[15]	GraphSAGE	-	-	0.7023	-0.0550	0.2986

TABLE VI: Correlations between several potential factors and attack performance.

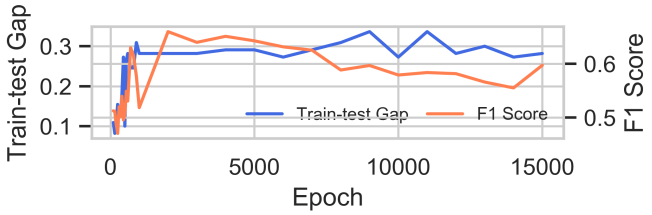
based attacks, the threshold-based attacks are more sensitive to the fluctuation of the overfitting.

Impact of the model architecture complexity. We then evaluate how the model architecture affects our attacks. As known, the complexity of the model structure may affect the overfitting level of the model and then the attack performance. To eliminate the impacts from GNN aggregation methods, we evaluate it on DeepGCN [42], which is a large scale GNN with deeper layers. Table V reports the relationship between the number of layers in the DeepGCN model and the effectiveness of the attacks. We observe that adding more layers to the DeepGCN model will reduce the train-test gaps and increase the F1 scores of our attacks. As mentioned in [42], adding layers may lead to higher training loss. In MIAs, higher training loss translates to less confidence in the member data which reduces the attack performance. Therefore, we can observe that the models with deeper layers achieve slightly stronger robustness.

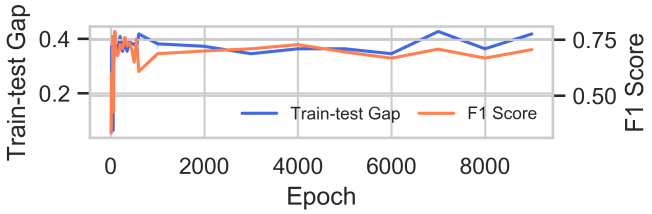
2) Effect of the Target Graph Data Property: We then explore how the property of the training data will affect the attack performance. Namely, whether some of the datasets are vulnerable or robust to the MIAs inherently.

> **Findings #6:** *GNNs for graph classification with more classes or trained by graph data with larger average degrees are inherently more vulnerable to the MIAs.*

We evaluate the correlation coefficient between the statistic values and the F1 score. The results of Spearman's correlations for both graph-level classification on PORITIEN_full and node-level classification on CORA are shown in Table VI. It can be found that for graph-level GNNs, the property related to the model such as the overfitting level (train-test gap) is highly related to the attack effectiveness, while the statistic of the training graph does not affect the attacks. However,

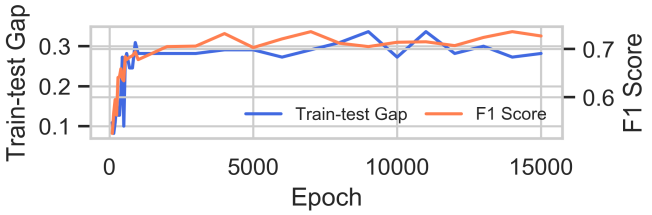


(a) PROTIEN_full

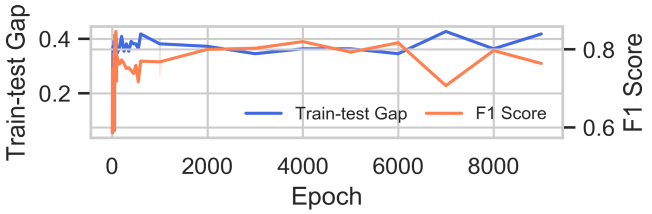


(b) DD

Fig. 3: Impact of the overfitting on training-based attacks.



(a) PROTIEN_full



(b) DD

Fig. 4: Impact of the overfitting on threshold-based attacks.

node-level classification, [14] emphasises that the graph can significantly affect their attacks while the overfitting level of the target model does not. The experimental results in [15] about the graph density also satisfy this observation as shown in Table VI.

The intuition behind this phenomenon is that the confidence vectors with a higher number of classes analysed by the attacker have higher dimensions and contain further information. Previous MIA works on images have also shown similar observations where MIAs are less effective when targeting binary classification tasks [13], [16]. As a result, it is more challenging to design and train attack models with high accuracy that can identify the membership based on these confidence vectors with low dimensions. Besides, we found that the GNN models trained by the dataset with a higher average degree are more vulnerable to our MIAs, which might be caused by the aggregation strategies of GNNs. According to our observations, models trained on graphs with a larger average degree are more confident in the membership records. Consequently, such confidence in the membership records benefits the classification of the attack models and improves the attack performance. We consider exploring the rationale behind this observation as our future work.

D. Comparison to MIAs on Node-level GNNs

To investigate the difference between our MIAs on graph-level GNNs and prior works on node-level GNNs, we compare and discuss our above findings to theirs. Table VI shows the comparison among the important effect factors of MIAs on different GNN applications. It can be found that graph-level MIAs are more correlated to the model property, while the performance of node-level MIAs depends more on the graph data property.

This observation actually satisfied the implications behind the two attacks. Previous attack performance on inferring the membership of only one node has shown to be highly correlated with its neighbours [14], [15]. It is consistent with their attack design, where the attack model derives the membership based on the posterior of also the neighbouring nodes. Therefore, the intuition behind their attack is exploiting the correlation among the target nodes and their neighbours. On the contrary, our attacks infer the membership of the inputs based on the final posterior of the entire graph. The effectiveness of the attacks fully relies on how the target GNN model overfits its training graph data. Therefore, the correlation between the target model property, especially the overfitting level, becomes the most significant factor in MIAs on graph-level classification.

VI. Conclusion and Future Directions

This paper investigates how GNNs are vulnerable to the MIAs and develops training-based and threshold-based attacks against various target GNN models. The experiment results demonstrate that our attacks are effective against GNN models. We investigate several impact factors of the MIAs, which are common in other ML models or unique in GNNs. Our findings show that overfitting is still the most significant factor that affects the attack performance. To further evaluate the attack transferability, we apply and compare the attacks with different adversarial background knowledge on the shadow datasets and the GNN types. The results demonstrate that the attack can still be effective even the type of the target model is unknown.

As a future direction, defence against the MIAs on GNNs becomes emerging. Moreover, how to deploy the MIAs utilising label-only exposures as [10] can also be an interesting extension of our work. We leave them as future work.

References

[1] H. Xu, C. Huang, Y. Xu, L. Xia, H. Xing, and D. Yin, "Global context enhanced social recommendation with hierarchical graph neural networks," in *Proc. IEEE ICDM*, 2020.

[2] W. Wang, H. Yin, X. Du, W. Hua, Y. Li, and Q. V. H. Nguyen, "Online user representation learning across heterogeneous social networks," in *Proc. ACM SIGIR*, 2019.

[3] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[4] Y. Liu, S. Pan, M. Jin, C. Zhou, F. Xia, and P. S. Yu, "Graph self-supervised learning: A survey," *arXiv:2103.00111*, 2021.

[5] D. Chen, N. Yu, Y. Zhang, and M. Fritz, "Gan-leaks: A taxonomy of membership inference attacks against generative models," in *Proc. ACM CCS*, 2020.

[6] Y. Zhu, X. Luo, Y. Li, B. Bu, K. Zhou, W. Zhang, and M. Lu, "Heterogeneous mini-graph neural network and its application to fraud invitation detection," in *Proc. IEEE ICDM*, 2020.

[7] R. Pang, X. Zhang, S. Ji, X. Luo, and T. Wang, "Advmind: Inferring adversary intent of black-box attacks," in *Proc. ACM KDD*, 2020.

[8] X. He, J. Jia, M. Backes, N. Z. Gong, and Y. Zhang, "Stealing links from graph neural networks," in *Proc. USENIX Security*, 2021.

[9] N. Z. Gong and B. Liu, "Attribute inference attacks in online social networks," *ACM Trans. Priv. Secur.*, vol. 21, no. 1, pp. 3:1–3:30, 2018.

[10] C. A. Choquette-Choo, F. Tramèr, N. Carlini, and N. Papernot, "Label-only membership inference attacks," *CoRR*, vol. abs/2007.14321, 2020.

[11] S. Truex, L. Liu, M. Gursoy, L. Yu, and W. Wei, "Demystifying membership inference attacks in machine learning as a service," *IEEE transactions on services computing*, pp. 1–1, 2019.

[12] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models," in *Proc. NDSS*, 2019.

[13] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE S&P*, 2017.

[14] I. E. Olatunji, W. Nejdl, and M. Khosla, "Membership inference attack on graph neural networks," *CoRR*, vol. abs/2101.06570, 2021.

[15] X. He, R. Wen, Y. Wu, M. Backes, Y. Shen, and Y. Zhang, "Node-level membership inference attacks against graph neural networks," *arXiv preprint arXiv:2102.05429*, 2021.

[16] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *Proc. IEEE CSF*, 2018.

[17] K. Leino and M. Fredrikson, "Stolen memories: Leveraging model memorization for calibrated white-box membership inference," in *Proc. USENIX Security*, 2020.

[18] Z. Zhang, M. Chen, M. Backes, Y. Shen, and Y. Zhang, "Inference attacks against graph neural networks," in *Proc. USENIX Security*, 2022.

[19] J. Huang, Y. Xie, F. Yu, Q. Ke, M. Abadi, E. Gillum, and Z. M. Mao, "Socialwatch: detection of online service abuse via large-scale social graphs," in *Proc. ACM AsiaCCS*, 2013.

[20] N. Z. Gong, M. Frank, and P. Mittal, "Sybilbelief: A semi-supervised learning approach for structure-based sybil detection," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 6, pp. 976–987, 2014.

[21] M. Weber, G. Domeniconi, J. Chen, D. K. I. Weidele, C. Bellei, T. Robinson, and C. E. Leiserson, "Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics," *CoRR*, vol. abs/1908.02591, 2019.

[22] B. Wang, J. Jia, and N. Z. Gong, "Graph-based security and privacy analytics via collective classification with joint weight learning and propagation," in *Proc. NDSS*. The Internet Society, 2019.

[23] D. Kong and G. Yan, "Discriminant malware distance learning on structural information for automated malware classification," in *Proc. ACM KDD*, 2013.

[24] S. D. Nikolopoulos and I. Polenakis, "A graph-based model for malware detection and classification using system-call groups," *J. Comput. Virol. Hacking Tech.*, vol. 13, no. 1, pp. 29–46, 2017.

[25] M. Hassen and P. K. Chan, "Scalable function call graph-based malware classification," in *Proc. ACM CODASPY*, 2017.

[26] J. Yan, G. Yan, and D. Jin, "Classifying malware represented as control flow graphs using deep graph convolutional neural network," in *Proc. IEEE DSN*, 2019.

[27] J. Li, D. Cai, and X. He, "Learning graph-level representation for drug discovery," *CoRR*, vol. abs/1709.03741, 2017.

[28] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong, "Memguard: Defending against black-box membership inference attacks via adversarial examples," in *Proc. ACM CCS*, 2019.

[29] M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song, J. Zhou, C. Ma, L. Yu, Y. Gai, T. Xiao, T. He, G. Karypis, J. Li, and Z. Zhang, "Deep graph library: A graph-centric, highly-performant package for graph neural networks," *arXiv preprint arXiv:1909.01315*, 2019.

[30] V. P. Dwivedi, C. K. Joshi, T. Laurent, Y. Bengio, and X. Bresson, "Benchmarking graph neural networks," *CoRR*, vol. abs/2003.00982, 2020.

[31] P. D. Dobson and A. J. Doig, "Distinguishing enzyme structures from non-enzymes without alignments," *Journal of molecular biology*, 2003.

[32] D. Szklarczyk, A. L. Gable, D. Lyon, A. Junge, S. Wyder, J. Huerta-Cepas, M. Simonovic, N. T. Doncheva, J. H. Morris, P. Bork, L. J. Jensen, and C. von Mering, "STRING v11: protein-protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets," *Nucleic Acids Res.*, vol. 47, no. Database-Issue, pp. D607–D613, 2019.

[33] A. Krizhevsky, "Learning multiple layers of features from tiny images," *University of Toronto*, 05 2012.

[34] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[35] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, 2012.

[36] C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, and M. Neumann, "Tudataset: A collection of benchmark datasets for learning with graphs," *CoRR*, vol. abs/2007.08663, 2020.

[37] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. ICLR*. OpenReview.net, 2017.

[38] X. Bresson and T. Laurent, "Residual gated graph convnets," *CoRR*, vol. abs/1711.07553, 2017.

[39] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *Proc. ICLR*, 2019.

[40] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. ICLR*, 2018.

[41] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. NIPS*, 2017.

[42] G. Li, M. Müller, A. K. Thabet, and B. Ghanem, "Deepgcns: Can gcns go as deep as cnns?" in *Proc. ICCV*, 2019.