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

***Abstract*—In light of the wide application of Graph Neural Networks (GNNs), 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. 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[1]. Our findings also confirm that, 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.**

***Keywords*-Membership Inference Attacks, Graph Classification, Graph Neural Networks**

I. Introduction

Graph Neural Networks (GNNs) have achieved state-of-the-art performance by generalising neural networks for graphs and been widely used to analyse the graph-structure data in a wide range of applications [1], [2]. Despite their great power, privacy and security concerns about information exposure in GNNs have been raised in sensitive applications [3]–[5]. 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. 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) [3], [6]–[9]. Given access to a model, the attacker can infer whether an arbitrary data record has been used during the training period of this model. In the domain of GNNs, recent studies [6], [7], [10] start to explore the feasibility of MIA over node-level GNN models. Specifically, node membership inference attacks [10], [11] can infer whether a given node has been used during the training of a target GNN model. Some other inference attacks [6] 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 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 inherent property of graph data. Such inference is implemented by utilising the strong correlation between connected nodes, which 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 [9] in DNN models, overfitting is the most significant cause for the MIAs, while studies [10], [11] 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 [9]. The transferring attacks with enhanced generalisability pose larger privacy threats. Therefore, the transferability of MIA on GNNs needs also 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

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

capabilities of the attacker. Specifically, the attack issues a query to the target model and receives confidence scores as a response. Intuitively, since the confidence scores are different for member/nonmember inputs, our attacks can identify their memberships in high confidence. 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. 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 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 [8], [9], [12] have shown that attackers can infer the training records of various machine learning models via MIAs, and achieve outperformed attack success rate and precision. However, all above mentioned works show the successful MIAs on the machine learning model trained on Euclidean space. There are also several preliminary researches [6], [10], [11] on the MIAs against node-level classifiers. However, these node-level MIAs only utilise the information of the sub-graph around the target node and do not consider the embedding of the entire graph. Recently, Zhang *et al.* [13] 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 Formalization

A. Problem Definition

In this paper, we define the graph classification model as the victim model and a Membership Inference Attack targeting the confidentiality of the victim model's training data membership.
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.
Definition 2.2. (Membership Inference Attacks in GNNs:) Given a GNN model $f_\theta(.)$, it is trained on a labelled 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,

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

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 [14]. 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. 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. 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). We will further relax this assumption and discuss our attacks when the attacker does not know the type of GNNs.

IV. The Proposed Attacks

A. Training-based Attacks

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

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}

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 shows how to construct the attack model and the detailed steps for the attack model construction 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.
- *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 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.

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 results 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.

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. We propose 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, Cross-entropy is used as the loss function for 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. Note that, the threshold value can be chosen by considering the requirements of the attacks as prior work [8]. Specifically, the higher threshold can be selected when the attacker focuses more on precision, while choosing 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.

A. Experimental Setup

Datasets. To evaluate our attacks, we use seven real-world datasets [15]: PROTEIN_full, DD, ENZYMES, OGBG-PPA, CIFAR10, MNIST and NCI. Note that, CIFAR10 and MNIST are converted into graphs using super-pixels [16]. 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 attacker.

GNN types. We evaluate five popular GNNs using these datasets. For PROTEIN_full, DD, ENZYMES, we use GCN [17], GateGCN [18], GIN [19], GAT [20] as the target models. And for CIFAR10 and MNIST, we use the above four GNNs as well as GraphSAGE [21] 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 [22] in OGBG-PPA and NCI. We also use MLP as a non-graph-structure baseline model as [23] 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. Since membership inference is a binary classification problem, we adopt the attack F1 score to measure the overall performance of our attacks, following prior work on MIAs [9]. We run all our experiments 15 times.

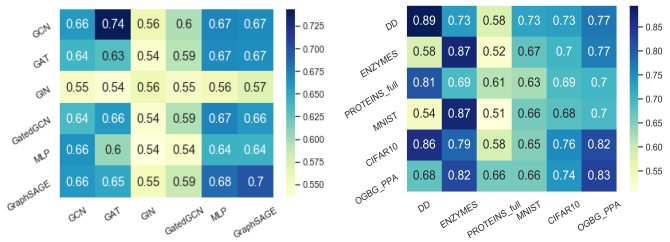
B. Attack Performance

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.*

Table I shows the attack performance of both our attacks on seven datasets with different GNN methods. For most of the datasets, our attacks can achieve more than 70% F1 Score, which confirms our attacks are effective in various GNN models on different tasks. 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.

We compare our attacks on GNNs with the MLP baseline as [23]. From the results, our attacks reach better performance on GNNs than MLP for most of the datasets. It is shown that



(a) GNN Methods (b) Graph Datasets

Fig. 1: Confusion matrix of the attack transferability evaluations.

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 the GateGCN model trained on DD has a 0.885 F1 score, which is about 0.333 higher than 0.552 for the MLP model.

2) Transferability: 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 #2:** *Our training-based MIAs have strong transferability with the shadow models trained as different GNN types and different datasets.*

Figure 1(a) 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. Figure 1(b) 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.

Different from the training-based attacks, we observe that the mismatching of the dataset domain can lead to a dramatic reduction of the threshold-based attack effectiveness for the transferring attacks in cross-domain. Namely, 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.

3) Comparison between Two Proposed Attacks: Finally, we summarise the advantages and disadvantages of both our training-based attacks and the threshold-based attacks.

> **Findings #4:** *The threshold-based MIAs can achieve even better attack performance but much lower 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

Dataset	Model	Training Accuracy	Testing Accuracy	Train-test Gap	Training-based Attack	Threshold-based Attack
PROTEIN_ful	GateGCN	0.990	0.710	0.280	0.595(0.063)	0.678(0.021)
	GCN	1.000	0.688	0.313	**0.668(0.043)**	0.690(0.028)
	GIN	0.730	0.690	0.040	0.561(0.075)	0.665(0.005)
	GAT	1.000	0.660	0.340	0.630(0.066)	0.677(0.028)
	MLP(baseline)	0.990	0.660	0.340	0.642(0.065)	**0.701(0.027)**
DD	GateGCN	1.000	0.630	0.370	**0.885(0.030)**	**0.845(0.049)**
	GCN	1.000	0.630	0.370	0.733(0.057)	0.774(0.001)
	GIN	1.000	0.610	0.390	0.597(0.051)	0.673(0.002)
	GAT	1.000	0.670	0.330	0.792(0.036)	0.793(0.007)
	MLP(baseline)	0.780	0.650	0.130	0.552(0.054)	0.533(0.014)
ENZYMES	GateGCN	1.000	0.550	0.450	**0.782(0.072)**	0.848(0.038)
	GCN	1.000	0.520	0.480	0.778(0.088)	**0.854(0.019)**
	GIN	0.940	0.480	0.460	0.592(0.067)	0.668(0.004)
	GAT	1.000	0.530	0.470	0.774(0.043)	**0.854(0.009)**
	MLP(baseline)	1.000	0.380	0.620	0.707(0.058)	0.807(0.011)
CIFAR10	GateGCN	1.000	0.476	0.524	0.859(0.010)	**0.860(0.007)**
	GCN	0.995	0.363	0.632	0.754(0.013)	0.759(0.012)
	GIN	0.984	0.319	0.666	0.600(0.013)	0.757(0.176)
	GAT	0.989	0.416	0.573	0.682(0.012)	0.581(0.050)
	GraphSAGE	1.000	0.494	0.506	**0.860(0.008)**	0.755(0.107)
	MLP(baseline)	1.000	0.354	0.646	0.721(0.010)	0.661(0.064)
MNIST	GateGCN	1.000	0.918	0.082	0.625(0.031)	0.712(0.008)
	GCN	1.000	0.805	0.195	**0.759(0.008)**	0.662(0.075)
	GIN	0.984	0.755	0.228	0.547(0.010)	0.674(0.002)
	GAT	1.000	0.888	0.112	0.646(0.033)	0.719(0.006)
	GraphSAGE	1.000	0.910	0.090	0.693(0.013)	**0.737(0.006)**
	MLP(baseline)	1.000	0.856	0.144	0.612(0.028)	0.671(0.034)
OGBG_PPA	DeepGCN	1.000	0.588	0.412	0.826(0.011)	0.796(0.088)
NCI	GCN	1.000	0.622	0.378	0.649(0.029)	-

TABLE I: Accuracy of the target models for different datasets and the corresponding performances.

Model Architecture	Train-test Gap	F1 Score
28-layer	0.412	0.826(0.011)
22-layer	0.484	0.858(0.009)
20-layer	0.480	0.860(0.013)
18-layer	0.490	0.864(0.010)
16-layer	0.510	0.852(0.012)
12-layer	0.500	0.838(0.010)

TABLE II: F1 scores comparison among attack models with different model complexity.

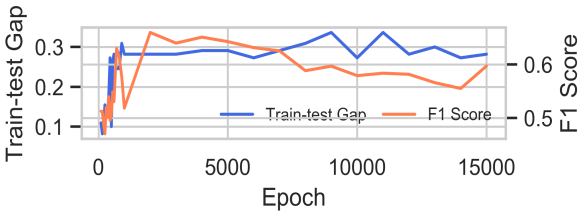


Fig. 2: Impact of the overfitting.

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

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 [9], we first analyse the relationship between the overfitting level of the target GNN models and the attack performance. Figure 2 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.

Impact of the model architecture complexity. We then evaluate how the model architecture affects our attacks. Table II 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 [22], 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.

			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	[10]	GCN	-	-	-	0.3857	-0.2524
	[11]	GraphSAGE	-	-	0.7023	-0.0550	0.2986

TABLE III: Correlations between several potential effect factors and attack performance.

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.*

Impact of the target graph data property. 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 III. 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, node-level classification, [10] emphasises that the graph can significantly affect their attacks while the overfitting level of the target model does not. The experimental results in [11] about the graph density also satisfy this observation as shown in Table III.

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. As shown in Table III, 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 [10], [11]. It is consistent with their attack design, where the attack model derives the membership based on the posterior of also the neighbour nodes. 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 also 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.

References

[1] 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.

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

[3] 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.

[4] 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.

[5] H. Zhang, B. Wu, X. Yang, C. Zhou, S. Wang, X. Yuan, and S. Pan, "Projective ranking: A transferable evasion attack method on graph neural networks," in *Proc. CIKM '20*. ACM, 2020.

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

[7] 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.

[8] 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.

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

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

[11] 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.

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

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

[14] 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.

[15] 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.

[16] 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.

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

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

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

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

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

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

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