Proceedings of Machine Learning Research 204:1–13, 2023 Conformal and Probabilistic Prediction with Applications

Conformalized Adversarial Attack Detection for Graph Neural Networks

Sofiane Ennadir ENNADIR@KTH.SE
Amr Alkhatib ALKHAT@KTH.SE
Henrik Boström BOSTROMH@KTH.SE
Michalis Vazirgiannis MVAZ@KTH.SE
School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Sweden

Editor: Harris Papadopoulos, Khuong An Nguyen, Henrik Boström and Lars Carlsson

Abstract

Graph Neural Networks (GNNs) have achieved remarkable performance on diverse graph representation learning tasks. However, recent studies have unveiled their susceptibility to adversarial attacks, leading to the development of various defense techniques to enhance their robustness. In this work, instead of improving the robustness, we propose a framework to detect adversarial attacks and provide an adversarial certainty score in the prediction. Our framework evaluates whether an input graph significantly deviates from the original data and provides a well-calibrated p-value based on this score through the conformal paradigm, therby controlling the false alarm rate. We demonstrate the effectiveness of our approach on various benchmark datasets. Although we focus on graph classification, the proposed framework can be readily adapted for other graph-related tasks, such as node classification.

Keywords: Conformal Prediction, Adversarial Attacks, Graph Neural Networks.

1. Introduction

Graph-structured data is an essential component of many domains, such as chemoinformatics, bioinformatics, and social network analysis. In these domains, graphs provide a natural way to represent complex relationships between entities, such as molecules, proteins, and social actors, as well as their attributes. The vast use of machine learning applications in these domains has motivated the development of specialized neural network models that can operate on graph-structured data, known as graph neural networks (GNNs). These models leverage the graph structure to learn meaningful representations of nodes that could be combined to represent the whole graph. Hence, GNNs have emerged as a powerful tool for learning both node and graph representations. They enable effective information sharing between connected nodes and can capture high-order relationships between them, which are meaningful in many applications. Over the past few years, GNNs have been successfully applied to a wide range of problems, including drug design, where they have been used to predict the efficacy and safety of drug candidates based on their molecular and biological properties (Kearnes et al., 2016); and session-based recommendation, where they have been used to recommend items to users based on their historical behavior (Wu et al., 2019b).

Adversarial attacks have been shown to be highly effective in compromising the accuracy and reliability of deep learning architectures, particularly in the field of computer

vision (Goodfellow et al., 2014). These attacks involve injecting small, well-crafted perturbations into the input data, which can lead to unreliable results and predictions, limiting the applicability of these models to real-world problems. Similar to other deep neural networks, GNNs are also vulnerable to adversarial attacks. Recent studies have shown that even small structural perturbations to the input graphs can easily fool a GNN (Dai et al., 2018a; Zügner et al., 2018; Günnemann, 2022). This vulnerability poses a critical threat to the reliability of GNNs, particularly in safety-critical applications such as finance and healthcare. For example, an attacker could manipulate social network data or online reviews to spread false information or influence recommendation systems, causing significant harm. As a result, different attack scenarios and schemes have been proposed to explore the robustness of GNNs. In parallel, a defense effort is being conducted to mitigate the possible effect of these perturbations. Recently, different defense techniques have been proposed; these include augmenting training data with adversarial examples and retraining the model (Feng et al., 2019), enhancing the robustness of an input GNN (Zhang and Zitnik, 2020), and more recently proposing robustness certificates (Schuchardt et al., 2021). We note that adversarial attacks on GNNs can manifest themselves in various forms due to the diverse nature of graphs. These forms include structural perturbation attacks, node feature attacks, and edge feature attacks. Structural perturbation attacks involve injecting or removing nodes or edges from the input graph, thereby producing erroneous predictions and classifications. Node feature and edge feature attacks entail manipulating the nodes/edges features within the input graph to achieve the attacker's desired outcome.

While the majority of defense methods for enhancing the robustness of victim models focus on modifying the message-passing scheme, pre-processing the input graph, or editing the underlying scheme, in this work, we take a different route by providing an adversarial estimation at inference time. Specifically, we aim to control the false alarm rate, i.e., incorrectly classifying a non-attack as an attack, while at the same time still being able to identify as large a fraction of the attacks as possible. In this work, we propose to leverage conformal prediction (CP) for this task. The proposed framework does neither require prior knowledge of the victim model nor the underlying data distribution, making it possible to approach the model in a black-box manner. The approach is therefore model-agnostic and allows for more straightforward implementation with various GNN architectures. We demonstrate instances of our framework on two popular GNNs and evaluate them on standard graph classification datasets. Our main contributions are summarized as follows:

- We propose a general framework based on conformal prediction for controlling the false alarm rate for adversarial attack detection.
- The framework assumes no knowledge about the underlying architecture or the underlying data distribution, thus allowing us to demonstrate it on two popular GNN models.
- We evaluate the proposed framework on several benchmark graph classification datasets where we show its ability to detect adversarial attacks.

2. Related work

Attacking machine learning models has received significant attention in recent years, with numerous studies focused on images (Goodfellow et al., 2014; Ren et al., 2020). More recently, research has emerged on attacking machine learning models operating on discrete data such as graphs. However, the discrete nature of graphs presents unique challenges for applying attack methods used in other domains. To address this issue, many existing graph-based attack methods frame the problem as a search problem, aiming to find the closest adversarial perturbation to a given input graph. This approach has led to the development of various attack strategies, such as Nettack (Zügner et al., 2018), which uses a greedy optimization algorithm to attack the graph structure and node features. Furthermore, Zügner and Günnemann (2019) introduced a bi-level optimization approach that leverages meta-gradients to generate adversarial attacks. Building on this work, Zhan and Pei (2021) proposed a black-box gradient attack algorithm to overcome limitations of previous work. Another approach, proposed by Dai et al. (2018b), uses reinforcement learning to solve the search problem and generate adversarial attacks. Notably, our proposed framework does not require any prior knowledge of the underlying model, enabling it to be applied to various GNN architectures in a black-box manner.

The area of defending GNNs against adversarial attacks is still relatively unexplored compared to that of models related to image classification. The majority of techniques are primarily centered on ad hoc defense mechanisms. Similar to image-based models, techniques such as robust training (Zügner and Günnemann, 2019a) and aggregation (Geisler et al., 2020) have been introduced as means to enhance the robustness of GNNs. Furthermore, low-rank matrix estimation approaches (Ma et al., 2021) have been used to protect against adversarial attacks. For instance, GNN-Jaccard (Wu et al., 2019a) employs preprocessing of the graph's adjacency matrix to identify potential edge manipulation. Similarly, GNN-SVD (Entezari et al., 2020) uses a low-rank approximation of the adjacency matrix to eliminate noise. Additionally, other methods such as edge pruning (Zhang and Zitnik, 2020) and transfer learning (Tang et al., 2020) have been employed to minimize the impact of poisoning attacks. Finally, a low-pass "message passing" mechanism that can be intergrated into existing GCN architecture has been proposed to both defend and provide theoretical guarantees against structural perturbations. Despite the moderate success of the aforementioned defense strategies in mitigating current adversarial attacks, their heuristic approach does not ensure the robustness of the underlying GNN. This may leave these defenses vulnerable to other new, more sophisticated attack methods in the future. Consequently, there has been an increasing focus on exploring robustness certificates (Zügner and Günnemann, 2019b; Bojchevski and Günnemann, 2019) as a promising direction in tackling adversarial attacks. These certificates offer attack-independent assurances of the model's prediction stability. For example, Bojchevski et al. (2020) utilized randomized smoothing to provide a highly scalable and model-agnostic certificate for graphs.

3. Preliminaries

Before continuing with our contribution, we introduce the graph classification problem and some key notation.

3.1. Problem Setup

Let $G = (V, E)$ be a graph where V is its set of vertices and E its set of edges. The number of vertices is denoted $n = |V|$ and, respectively the number of edges as $m = |E|$. The topology of a graph can be represented by its adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ that encodes edge information. The (i, j)-th element of the adjacency matrix is equal to the weight (1 in the case of unweighted graphs) of the edge between the i-th and j-th node of the graph and a weight of 0 in case the edge does not exist. In some settings, the nodes of a graph might be annotated with feature vectors. We use $\mathbf{X} \in \mathbb{R}^{n \times D}$ to denote the node features where D is the dimensionality; the features of the i-th node corresponds to the i-th row of $\mathbf{X}$.

In a graph classification setting, we have a set of graphs $\mathcal{G} = \{G_1, G_2, \ldots, G_N\}$ and each graph G_i is associated with a class label $y_i \in \mathcal{Y} = \{1, 2, ..., K\}$, K being the number of classes. The graph classification problem is defined as the task of finding a classifier $h \colon \mathcal{G} \rightarrow \mathcal{Y}$ that associates to each graph G_i a predicted label $\hat{y}_i$ while minimizing a classification loss (e. g., cross-entropy loss).

3.2. Graph Neural Networks

GNNs have recently become the standard approach for dealing with graph classification tasks. A GNN typically consists of a series of neighborhood aggregation layers that use both the graph structure and the features of the nodes to iteratively generate new representations. In other words, a GNN updates the feature vectors of the nodes by aggregating local neighborhood information.

Suppose we have a GNN model that contains T neighborhood aggregation layers. Let also $\mathbf{h}_v^{(0)}$ denote the initial feature vector of node v, i. e., the row of matrix $\mathbf{X}$ that corresponds to node v. At each iteration ($t > 0$), the hidden state $\mathbf{h}_v^{(t)}$ of a node v is updated as follows:

$$\begin{aligned} \mathbf{a}_v^{(t)} &= \text{AGGREGATE}^{(t)}\Big(\big\{\mathbf{h}_u^{(t-1)} \colon u \in \mathcal{N}(v)\big\}\Big) \\ \mathbf{h}_v^{(t)} &= \text{COMBINE}^{(t)}\Big(\mathbf{h}_v^{(t-1)}, \mathbf{a}_v^{(t)}\Big) \end{aligned} \tag{1}$$

where AGGREGATE is a permutation invariant function that maps the feature vectors of the neighbors of a node v to an aggregated vector. This aggregated vector is passed along with the previous representation of v (i. e., $\mathbf{h}_v^{(t-1)}$) onto the COMBINE function which combines those two vectors and produces the new representation of v. After T iterations of neighborhood aggregation, to produce a graph-level representation, GNNs apply a permutation invariant readout function, e. g., the sum or mean operator, to the feature vectors of all nodes of the graph as follows:

$$\mathbf{h}_G = \text{READOUT}\Big(\big\{\mathbf{h}_v^{(T)} \colon v \in V\big\}\Big) \tag{2}$$

4. Methods

In this section, we present our proposed method for detecting adversarial examples via conformal prediction. Firstly, we give an overview of adversarial attacks in the graph domain. Then, we introduce the conformal prediction process and provide a detailed description of the approach.

4.1. Adversarial attacks

We consider three measurable spaces with a defined norm over each space $(\mathcal{G}, \|.\|_{\mathcal{G}})$, $(\mathcal{X}, \|.\|_{\mathcal{X}})$ and $(\mathcal{Y}, \|.\|_{\mathcal{Y}})$. Let $\{(A_1, X_1, y_1), \ldots, (A_N, X_N, y_N)\} \in (\mathcal{G}, \mathcal{X}, \mathcal{Y})^N$ be a sampled set from an underlying probability distribution $\mathcal{D}$ defined on $(\mathcal{G}, \mathcal{X}, \mathcal{Y})$. We consider a graph-based classifier f trained as described in section 3. Given an input graph point $G \in \mathcal{G}$ and its corresponding label $y \in \mathcal{Y}$ where $f(G) = y$, the goal of an adversarial attack is to produce a perturbed graph $\tilde{G}$ slightly different from the original graph with its predicted class being different from the predicted class of G. This could be formulated as finding a $\tilde{G}$ with $f(\tilde{G}) = \tilde{y} \neq y$ such that the perturbed graph $\tilde{G}$ is semantically similar to the input graph. We quantify this semantic similarity using a graph distance defined on our considered input metric spaces, for instance, we can use the following distance:

$$d([A, X], [\tilde{A}, \tilde{X}]) = \min_{P \in \Pi} \{\|A - P\tilde{A}P^T\|_2 + \|X - P\tilde{X}\|_2\}. \tag{3}$$

where Π denotes the collection of permutation matrices. It is worth mentioning that in the case of unattributed graphs, we can only consider the first part related to structural comparison, and in this case, this quantity is equivalent to the commonly used notion of graph edit distance. This measures the resemblance between two graphs by determining the minimum number of edge modifications that are necessary to transform one graph into another, taking into account graph isomorphism. It is important to note that even though we use the l_2 norm for our graph distance formulation, any other valid metric defined on the input space could be utilized instead.

4.2. Proposed framework

The main assumption of the proposed framework is that we have access to a set of graphs, all sampled IID from some static but unknown underlying distribution of *non-attacked* graphs. In contrast, we do not assume access to a similar sample of *attacked* graphs, due to the difficulty of collecting such graphs in reality and also that the underlying distribution of attacked graphs hardly can be assumed to be static, as the different types of attack are constantly evolving. The original training set, which through the application of the conformal prediction framework, will be further divided into a proper training set and a calibration set, hence containing instances with one label only. For the test instances, each prediction set will accordingly consist of either a singleton (the label *non-attack*) or be empty, which means that the label can be rejected at the specified level of confidence. An empty-set prediction may signal that the corresponding graph should be further investigated, as it, with high probability, could be excluded from the set of non-attacked graphs. Assuming *attack* to be the positive class and *non-attack* to be the negative, the confidence level hence

serves to control the false positive rate (or false alarm rate), i.e., the number of false positives divided by the sum of the number of false positives and true negatives. By lowering the confidence, the recall of the positive class will increase at the cost of an increased false alarm rate. Conversely, by increasing confidence, a reduced false alarm rate will lead to fewer graphs that need to be inspected, at the cost of missing out on more true attacks.

The main idea of the proposed framework is to define the nonconformity measure using a binary GNN classifier, trained from the proper training set (where all instances are labeled *non-attack* and a synthetically generated set of graphs (all labeled *attack*). Since the conformal prediction framework allows us to freely define the nonconformity measure, the exact way in which the synthetic (adversarial) graphs are constructed will not affect the validity (or false alarm rate). Still, if these adversarial graphs are properly generated, we can expect to see a higher proportion of true attacks among the graphs for which the non-attack label has been rejected.

Given an input graph G_{q+1} with its corresponding node features X_{q+1}, a set prediction is formed, using the conformal predictor formed from a chosen nonconformity measure.

Algorithm 1 Conformal Prediction for Adversarial Attacks

Input : Confidence level $1-\epsilon$, a trained GNN victim model f, a training set $\mathcal{N}$, a calibration set $\mathcal{C} = G_1, .., G_q$, input graph G_{q+1}

Output: Prediction set P_{q+1}

1 - Generate a set of adversarial graphs $\mathcal{A}$ with respect to f and $\mathcal{N}$

2 - Train a GNN S from $\mathcal{A} \cup \mathcal{N}$, where elements of the former set are labeled 1 ("attack") and the latter 0 ("non-attack")

3 - Compute nonconformity scores $\alpha_1, \ldots, \alpha_q$ for the calibration set, where $\alpha_i = S(G_i)$, for $i = 1, \ldots, q$

4 - Compute nonconformity score α_{q+1} for the input graph, where $\alpha_{q+1} = S(G_{q+1})$

5 - Compute p-value for the input graph:

$p_{q+1} = \frac{|\{i \in \{1,\ldots,q\} \text{ s.t. } \alpha_i > \alpha_{q+1}\}| + \tau |\{i \in \{1,\ldots,q\} \text{ s.t. } \alpha_i = \alpha_{q+1}\}|}{q+1}$ *where* $\tau \sim \mathcal{U}(0,1)$

6 - Produce prediction set: $P_{q+1} = \emptyset$ if $p_{q+1} < \epsilon$, else $P_{q+1} = \{0\}$

4.3. Nonconformity scores and synthetic adversarial graphs

Our proposed framework relies on identifying a suitable non-conformity score to accurately evaluate and rank the distance between an input graph and the available calibration set. In graph analysis, there are various metrics and similarity measures that can be employed, including traditional semantic measures such as degree, closeness, or betweenness centrality, as well as more advanced kernels such as the Random Walk kernel. Although these unsupervised methods are advantageous, our experiments have revealed that they may have limitations in terms of effectiveness. To overcome these limitations, we propose to enrich our original training set with crafted adversarial points and train a GNN-based model to classify the attack/non-attack aspect. As shown in Algorithm 1, this classifier will be the basis for the nonconformity scores. While in this work, we focus on structural perturbations, our framework can easily be adapted for node-feature-based attacks.

Given a trained GNN model f built on a specific task related to the considered dataset $\mathcal{G}$, a set of perturbed graphs $\{\tilde{G}_1, \tilde{G}_2, \ldots, \tilde{G}_M\}$ are generated from the set of training graphs that are correctly classified by the GNN. Those graphs are produced by applying a set of perturbation matrices $\{\mathbf{\Delta}_1, \mathbf{\Delta}_2, \ldots, \mathbf{\Delta}_M\}$ to the correctly classified graphs. It should be noted that the perturbations can be generated using various schemes. For example, the user may choose to use an available adversarial attack or opt for random perturbations. In our work, we focused on random perturbations drawn from a Gaussian distribution. We use this synthetically generated training set comprising original training graphs (corresponding to the "non-attack" class) and adversarial graphs (representing the "attack" class) to train another GNN that classifies attacks and non-attacks. This GNN serves as our non-conformity score.

We should note that the test and calibration set from the original dataset

We should note that when interacting with this latter GNN (during both training and testing), we

When interacting with this latter GNNs, it is important to note that we do not use the calibration and test set and their corresponding labels to eithet train or test it.

Instead, we use them to represent the "non-attack" aspect, assuming they are drawn from the same underlying distribution.

Based on the above, the proposed framework, as described in Algorithm 1, can be divided into two main components that are considered to be independent and hence respect the black-box setting.

(1) **Classifier.** This is an instance of a GNN model following the general graph classification scheme presented in Section 3. This model is trained for the graph classification task of the given dataset and serves as the main victim model on which we aim to apply our proposed framework.

(2) **Non-conformity Score.** This is another GNN-based classifier built using the original training set enriched with different generated adversarial attacks, as previously explained.

4.4. Connection to other available defenses

We argue that our proposed framework can easily be connected to other defense methodologies mentioned in Section 2. Specifically, low-rank matrix estimation-based methods such as GNN-Jaccard or GNN-SVD aim to make the input graph as similar as possible to the available training dataset to attenuate its adversarial effect while retaining the essential information in the input. The former can be reformulated as minimizing our non-conformity score while keeping the main information/signal contained on the input graph. Therefore, although it is not our primary focus, we can utilize our framework to generate pre-processing edits that could be used to enhance the robustness of graph-based classifiers against adversarial attacks for both structural and node-feature-based perturbations.

5. Experimental Evaluation

In this section, we first give details about the experimental settings and next describe the applied empirical evaluation. We afterward report on the performance of the proposed approach on real-world datasets.

Table 1: Statistics of the graph classification datasets used in our experiments.

Dataset	#Graphs	#Nodes	#Edges	#Classes
D&D	1178	284.32	715.66	2
NCI1	4110	29.87	32.30	2
PROTEINS	1113	39.06	72.82	2
MUTAG	187	18.03	39.80	2

5.1. Experimental Setup

We used a Graph Convolutional Network (GCN) (Kipf and Welling, 2017) to produce our non-conformity score where the number of message passing layers and hidden dimensions have been arranged differently from the GNN-based classifier (especially if the underlying victim model is a GCN) since the black-box setting, which we are interested in, assumes no knowledge about the inner architecture of the underlying model. We focused on perturbations based on adding Gaussian noise $\mathcal{N}(0, \mathbf{I})$ with a scaling parameter σ to control the attack budget. We demonstrate instances of the proposed framework on two main GNNs: (1) Graph Convolutional Network (GCN) (Kipf and Welling, 2017) and (2) Graph Isomorphism Network (GIN) (Xu et al., 2019) on real benchmark datasets. Each of these models consists of 2 layers corresponding to two message-passing processing (with their relevant Aggregate and Combine functions depending on the chosen underlying architecture). We evaluate the proposed approach on standard graph classification datasets derived from bioinformatics and chemoinformatics (MUTAG, PROTEINS, NCI1, D&D) (Morris et al., 2020); the datasets are summarized in Table 1. Note that these graph datasets come with either node labels or node attributes. We perform 10-fold cross-validation to estimate the generalization performance of the approach where we used the same folds as used in previous work (Errica et al., 2020). For all models, we used the sum operator as the readout function to produce graph-level representations. Furthermore, we train the different models by minimizing the cross-entropy loss function with the Adam optimizer and an initial learning rate of 10^{-3}. [1]

5.2. Validity Evaluation

We start by empirically evaluating the validity of the proposed method in terms of its statistical guarantees from the conformal prediction perspective. We will be using the available test set, which only represents the negative class since the corresponding graphs are all supposed to be drawn from the same underlying distribution of non-attacked graphs. We recall that setting a confidence level $1 - \epsilon$ limits the false positive rate not to exceed ϵ. To test this empirically, we evaluate the rejection level for various values of ϵ. Table 2 shows a summary, presenting the results over the four considered benchmark datasets for two different significance levels and for different values of σ (attack budget). Analyzing

1. The source code for training our proposed framework and reproducing the results is available at shorturl.at/aqs57.

Table 2: Conformal prediction summary for a GCN for different benchmark datasets and for different considered values.

Dataset	$\sigma = 0.5$		$\sigma = 1.0$	
	$\epsilon = 0.05$	$\epsilon = 0.1$	$\epsilon = 0.05$	$\epsilon = 0.1$
PROTEINS	.043	.081	.032	.113
MUTAG	.041	.063	.039	.068
NCI1	.027	.076	.043	.083
D&D	.063	.094	.058	.104

Table 3: Accuracy ($\pm$ standard deviation) of the proposed approach for detecting adversarial attacks on the different benchmark datasets.

Model		PROTEINS	MUTAG	D&D
GCN	$\sigma = 0.5$	93.2 ± 1.9	92.3 ± 5.4	95.4 ± 0.8
	$\sigma = 1.0$	95.8 ± 1.6	94.6 ± 4.9	95.9 ± 1.1
GIN	$\sigma = 0.5$	82.9 ± 3.8	85.6 ± 5.6	96.1 ± 1.3
	$\sigma = 1.0$	93.8 ± 2.7	95.7 ± 4.6	91.5 ± 0.4

the results shows that the proposed method is valid and reasonably well-calibrated for the different cases.

5.3. Defense Performance Evaluation

One aspect of our proposed methodology involves detecting adversarial attacks, which can then be used as a defense approach against such attacks. We are interested therefore in this part in assessing the efficacy of our method in detecting adversarial attacks, we utilized an evaluation strategy similar to that used in the field of anomaly detection, specifically in supervised anomaly detection. Initially, we considered the available test set consisting of graphs that are, by definition, non-adversarial. Then, we augmented this test set by generating additional adversarial attacks. These attacks were meticulously designed to ensure that the final testing set had a balanced number of adversarial and non-adversarial samples. By adopting this evaluation strategy, we aimed to replicate a real-world scenario where a machine learning model is deployed and it encounters both legitimate and malicious inputs. Our objective was to verify if our method could accurately distinguish between these two types of inputs and identify adversarial attacks, even in the presence of non-adversarial inputs. We would like to emphasize that our primary evaluation hypothesis is that the model is susceptible to the same types of attacks that were used during the training of our non-conformity score.

Table 3 presents the classification accuracy and their corresponding standard deviations. Our observations indicate that the proposed framework can effectively and precisely detect

adversarial attacks, thereby providing an additional security layer to enhance the robustness of the underlying victim model. As expected, the ability of the model to detect adversaries increases with the σ value, which controls the amplitude of the attack. In practice, attackers typically aim to create the smallest possible perturbation, and our framework demonstrates that it can also detect such attacks effectively. This is evidenced by our results for $\sigma = 0.5$, which indicate that our framework can detect adversaries with a high degree of accuracy even when the perturbations are relatively small.

6. Concluding Remarks

In this paper, we have applied conformal prediction to the task of detecting adversarial attacks, given an underlying static victim model. Our approach is model-agnostic and operates in a black-box manner, making it easier to adapt to various architectures and use cases. While our focus has been on structural perturbations, the method can be adapted to different aspects of graph attacks, such as node/edge feature-based perturbations. Our empirical evaluation confirms the validity guarantees provided by the conformal scheme and demonstrates the effectiveness of the proposed approach in detecting adversarial attacks for different benchmark graph classification datasets.

We consider that the proposed framework can be easily adapted to other deep learning-related tasks, such as computer vision or natural language processing. Thus, the next step is to work on the generalization of the method. In addition, we should investigate the applicability of our method in tasks where the IID assumption cannot necessarily be made, such as for the node classification task. Such an investigation can provide insights into the potential limitations of our method and help identify potential areas for improvement.

Acknowledgments

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The computation (through GPU) was enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at Alvis partially funded by the Swedish Research Council through grant agreement no. 2022-06725."

References

Aleksandar Bojchevski and Stephan Günnemann. Certifiable robustness to graph perturbations, 2019. URL https://arxiv.org/abs/1910.14356.

Aleksandar Bojchevski, Johannes Klicpera, and Stephan Günnemann. Efficient robustness certificates for discrete data: Sparsity-aware randomized smoothing for graphs, images and more, 2020. URL https://arxiv.org/abs/2008.12952.

Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. Adversarial attack on graph structured data, 2018a. URL https://arxiv.org/abs/1806.02371.

Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. Adversarial Attack on Graph Structured Data. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1115–1124, 2018b.

Negin Entezari, Saba A. Al-Sayouri, Amirali Darvishzadeh, and Evangelos E. Papalexakis. All you need is low (rank): Defending against adversarial attacks on graphs. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, page 169–177, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368223. doi: 10.1145/3336191.3371789. URL https://doi.org/10.1145/3336191.3371789.

Federico Errica, Marco Podda, Davide Bacciu, and Alessio Micheli. A Fair Comparison of Graph Neural Networks for Graph Classification. In *8th International Conference on Learning Representations*, 2020.

Fuli Feng, Xiangnan He, Jie Tang, and Tat-Seng Chua. Graph adversarial training: Dynamically regularizing based on graph structure, 2019. URL https://arxiv.org/abs/1902.08226.

Simon Geisler, Daniel Zügner, and Stephan Günnemann. Reliable graph neural networks via robust aggregation, 2020. URL https://arxiv.org/abs/2010.15651.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2014. URL https://arxiv.org/abs/1412.6572.

Stephan Günnemann. Graph neural networks: Adversarial robustness. In *Graph Neural Networks: Foundations, Frontiers, and Applications*, pages 149–176. Springer, 2022.

Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of Computer-Aided Molecular Design*, 30(8):595–608, 2016.

Thomas N Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations*, 2017.

Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z. Sheng, Hui Xiong, and Leman Akoglu. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2021. doi: 10.1109/tkde.2021.3118815. URL https://doi.org/10.1109%2Ftkde.2021.3118815.

Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. TUDataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020.

Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. Adversarial attacks and defenses in deep learning. *Engineering*, 6(3):346–360, 2020. ISSN 2095-8099. doi: https://doi.org/10.1016/j.eng.2019.12.012. URL https://www.sciencedirect.com/science/article/pii/S209580991930503X.

Jan Schuchardt, Aleksandar Bojchevski, Johannes Gasteiger, and Stephan Günnemann. Collective robustness certificates: Exploiting interdependence in graph neural networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=ULQdiUTHe3y.

Xianfeng Tang, Yandong Li, Yiwei Sun, Huaxiu Yao, Prasenjit Mitra, and Suhang Wang. Transferring robustness for graph neural network against poisoning attacks. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. ACM, jan 2020. doi: 10.1145/3336191.3371851. URL https://doi.org/10.1145%2F3336191.3371851.

Huijun Wu, Chen Wang, Yuriy Tyshetskiy, Andrew Docherty, Kai Lu, and Liming Zhu. Adversarial examples for graph data: Deep insights into attack and defense. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4816–4823. International Joint Conferences on Artificial Intelligence Organization, 7 2019a. doi: 10.24963/ijcai.2019/669. URL https://doi.org/10.24963/ijcai.2019/669.

Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based Recommendation with Graph Neural Networks. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pages 346–353, 2019b.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How Powerful are Graph Neural Networks? In *7th International Conference on Learning Representations*, 2019.

Haoxi Zhan and Xiaobing Pei. Black-box Gradient Attack on Graph Neural Networks: Deeper Insights in Graph-based Attack and Defense. *arXiv preprint arXiv:2104.15061*, 2021.

Xiang Zhang and Marinka Zitnik. Gnnguard: Defending graph neural networks against adversarial attacks, 2020. URL https://arxiv.org/abs/2006.08149.

Daniel Zügner and Stephan Günnemann. Adversarial attacks on graph neural networks via meta learning. In *7th International Conference on Learning Representations*, 2019.

Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial Attacks on Neural Networks for Graph Data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2847–2856, 2018.

Daniel Zügner and Stephan Günnemann. Certifiable robustness and robust training for graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery &amp Data Mining*. ACM, jul 2019a. doi: 10.1145/3292500.3330905. URL https://doi.org/10.1145%2F3292500.3330905.

Daniel Zügner and Stephan Günnemann. Certifiable robustness and robust training for graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery &amp Data Mining*. ACM, jul 2019b. doi: 10.1145/3292500.3330905. URL https://doi.org/10.1145%2F3292500.3330905.

Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, jul 2018.