AI Open 2 (2021) 69–78

Contents lists available at ScienceDirect

AI Open

journal homepage: www.sciencedirect.com/journal/ai-open

Robustness of deep learning models on graphs: A survey

Jiarong Xu, Junru Chen, Siqi You, Zhiqing Xiao, Yang Yang *, Jiangang Lu

Zhejiang University, China

ARTICLE INFO

Keywords:
Model robustness
Graph mining

ABSTRACT

Machine learning (ML) technologies have achieved significant success in various downstream tasks, e.g., node classification, link prediction, community detection, graph classification and graph clustering. However, many studies have shown that the models built upon ML technologies are vulnerable to noises and adversarial attacks. A number of works have studied the robust models against noise or adversarial examples in image domains and text processing domains, however, it is more challenging to learn robust models in graph domains. Adding noises or perturbations on graph data will make the robustness even harder to enhance – the noises and perturbations of edges or node attributes are easy to propagate to other neighbors via the relational information on a graph. In this paper, we investigate and summarize the existing works that study the robust deep learning models against adversarial attacks or noises on graphs, namely the robust learning (models) on graphs. Specifically, we first provide some robustness evaluation metrics of model robustness on graphs. Then, we comprehensively provide a taxonomy which groups robust models on graphs into five categories: anomaly detection, adversarial training, pre-processing, attention mechanism, and certifiable robustness. Besides, we emphasize some promising future directions in learning robust models on graphs. Hopefully, our works can offer insights for the relevant researchers, thus providing assistance for their studies.

1. Introduction

Machine learning (ML) technologies have become increasingly popular. They have attained impressive performances and successful applications on various downstream tasks such as image classification, object detection, traffic prediction, malware detection (Grosse et al., 2017; Tao et al., 2018; Xu et al., 2020a), speech recognition (Vaswani et al., 2017), automatic language translation (Papineni et al., 2002), product recommendations (Cheng et al., 2016; Guo et al., 2017), self-driving vehicles (Bojarski et al., 2016), online fraud detection and stock market trading (Pandit et al., 2007; Patel et al., 2015), *etc*. Deep Neural Networks (DNNs), the most popular tool among machine learning technologies, are widely used in many real-world applications. However, many studies have shown that DNN models are not robust enough, that is, they are easily be fooled by noises or adversarial examples (that is, the examples that are carefully designed to deceive the models by making minor or even imperceptible modifications to benign examples). A line of existing works has shown that DNNs are vulnerable in many applications, such as malware detection (Grosse et al., 2017; Tao et al., 2018; Xu et al., 2020a), audio recognition (Carlini and Wagner, 2018), object recognition (Goodfellow et al., 2014), sentiment analysis systems (Ebrahimi et al., 2017), *etc*. It is an urgent need to study robust models using machine learning technologies.

The graph-structured data is ubiquitous and plays a key role in many practical fields, including social network analysis, bioinformatics, chemistry, program analysis, etc. These graphs provide rich topology functions and common connectivity patterns, thus can help us better understand relational data. Deep learning on graphs has also achieved significant success in a wide range of applications (Goyal and Ferrara, 2018), including financial surveillance (Paranjape et al., 2017), recommendation systems (Wang et al., 2019), molecule analysis (Hamilton et al., 2017) and drug discovery (Gilmer et al., 2017), *etc*. However, network data is hard to obtain and most networks obtained in the real world are error-prone and structurally flawed due to incomplete sampling (Gueorgi, 2006), imperfect measurements (Butts, 2003; Namata and Getoor, 2009), individual non-response and dropout (Schafer and Graham, 2002), *etc*. This will inevitably introduce many types of errors, including erroneous, ambiguous and redundant information (Xu et al., 2020b). Thus, most network data obtained depicts an imperfect and incomplete picture of topological structure. These

* Corresponding author.
E-mail addresses: xujr@zju.edu.cn (J. Xu), jrchen_cali@zju.edu.cn (J. Chen), ysseven@zju.edu.cn (S. You), zhiqing.xiao@zju.edu.cn (Z. Xiao), yangya@zju.edu.cn (Y. Yang), lujg@zju.edu.cn (J. Lu).

https://doi.org/10.1016/j.aiopen.2021.05.002
Received 3 December 2020; Received in revised form 11 March 2021; Accepted 13 May 2021
Available online 24 June 2021

inaccurate representation of networks can even have an adverse effect on how networks are interpreted and damage information diffusion process, resulting in misleading conclusions. On the other hand, graph learning models, *e.g.*, Graph Neural Networks (GNNs) (Bruna et al., 2013; Henaff et al., 2015; Defferrard et al., 2016; Levie et al., 2018; Hamilton et al., 2017; Monti et al., 2017; Niepert et al., 2016; Cao et al., 2016) and network embedding (Ribeiro et al., 2017; Perozzi et al., 2014; Zhang et al., 2019a; Grover and Leskovec, 2016), have been shown to be vulnerable to adversarial examples (Chen et al., 2020; Bojchevski and Günnemann, 2019a; Dai et al., 2018; Zügner and Günnemann, 2019a). Adversarial attacks on graphs pose themselves as serious security challenges for many real-world systems. There have been lots of works focus on learning the robust models in image domains, but few have been studied the robustness of models on graphs. Hence, *it is of practical importance to build robust learning models on graphs against noises or adversarial attacks.*

There have been a few surveys mentioning the robustness of deep learning models on graphs (Sun et al., 2018a; Jin et al., 2020). Although they provided their own categories of robust graph models, they did not include some important robustness metrics. In addition, anomaly detection, the most commonly-used approach to enhance robustness, is ignored in the previous surveys. In this survey, we first introduce the robustness metrics and then aim to summarize and discuss the robust learning models on graphs against noises and adversarial examples from a more comprehensive perspective. The major contributions can be summarized as follows:

- We target the critical yet overlooked robust models on graphs against noises and adversarial attacks.
- We provide some evaluation metrics of model robustness on graphs.
- We divide existing works of robust models on graphs into five categories: anomaly detection, adversarial training, pre-processing, attention mechanism, and certifiable robustness. We provide a detailed and systematic analysis of these studies.
- We present some exciting future directions of the model robustness on graphs.

Our manuscript is organized as follows. Some notations and backgrounds are mentioned in section 2. In section 3, we show some evaluation metrics of model robustness on graphs. In section 4, we introduce the five categories of robust models on graphs in detail. Some future directions are presented in section 5. We conclude our manuscript by providing a conclusion in section 6.

2. Preliminaries

2.1. Notations

Formally, we represent a network as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes with $|\mathcal{V}|$ nodes, while $\mathcal{E}$ is the set of edges with $|\mathcal{E}|$ edges. We further denote $\mathcal{A}$ as the adjacency matrix of $\mathcal{G}$ and $\mathcal{D}$ as the degree matrix of $\mathcal{A}$. We augment $\mathcal{G}$ with the node attribute matrix $\mathcal{X}$ if nodes have certain attributes in particular applications. Also in some applications where edges have attributes, we augment $\mathcal{G}$ with the edge attribute matrix $\mathcal{H}$.

2.2. Victim models

In this survey, we use victim models to denote the models attacked by adversarial examples. We briefly summarize the victim models which are proven to be susceptible to adversarial examples, also known as non-robust models. In our context, we mainly discuss studies of adversarial examples for graph neural networks which are powerful tools in learning the representation of graphs (Sun et al., 2018b).

Graph Convolutional Networks (GCN) (Kipf and Welling, 2016) bridged the spectral-based GNNs with spatial-based ones, which later became one of the most successful GNN variants. The intuition of GCN follows that of CNN that it keeps aggregating and transforming the information from neighbor nodes to learn the representations for each node. However, though GNNs can achieve impressive performance across many kinds of tasks, the vulnerability to adversarial attacks of GNNs including GCN has been demonstrated as potential threats to industry and society applications [41, ?]. Besides, there are also other important graph learning algorithms that are possible to be attacked by adversarial examples such as network embeddings including LINE (Tang et al., 2015) and Deepwalk (Perozzi et al., 2014), graph-based semi-supervised learning (G-SSL) (Xiaojin and Zoubin, 2002), and knowledge graph embedding (Bordes et al., 2013).

2.3. Learning from graph data

In this section, we introduce the basic graph learning tasks such as node classification and graph classification. We use triple set $G = \{(c_i, \mathcal{G}_i, y_i)\}_{i\in[N]}$ to denote the training set with labels where N is the number of the samples. c_i is the i-th sample of the set and $\mathcal{G}_i$ and y_i respectively represents the corresponding (sub)graph and the label related to c_i. The uniform formula to represent both node classification and graph classification is given below:

$$\min_{\theta} \mathcal{L}_{train}(f_\theta(G)) = \sum_{(c_i, \mathcal{G}_i, y_i)\in G} \ell(f_\theta(c_i, \mathcal{G}_i), y_i), \tag{1}$$

where f_θ is the mapping function learned to predict the true labels with learnable parameters θ.

Node classification. For node-level classification, each node lies in the same graph $\mathcal{G}_i = \mathcal{G} = (\mathcal{V}, \mathcal{E})$ and $f_\theta(c_i, \mathcal{G}_i) = f_\theta(\mathcal{G})_i$ extracts the i-th node's representation from the whole single graph.

Graph classification. For graph-level classification, each individual graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ has a label and $f_\theta(c_i, \mathcal{G}_i) = f_\theta(\mathcal{G}_i)$ extracts the i-th graph's representation independent with other graphs.

2.4. Adversarial attacks on graphs

In this section, we give a general form of the objective for graph adversarial attacks and illustrate the damage of the attacks which indicates the urgent need to research into robust models on graphs (see Fig. 1).

Graph adversarial attacks. In image domain, the attack is straightforward to introduce small perturbations into pixels (showed as Fig. 2) which is a little different from that in graphs. As illustrated in Fig. 3, the target of graph attack can be both graph topology and node attributes. Formally, based on the formula showed in Section 2.3, we can define the attack objective on graph data as:

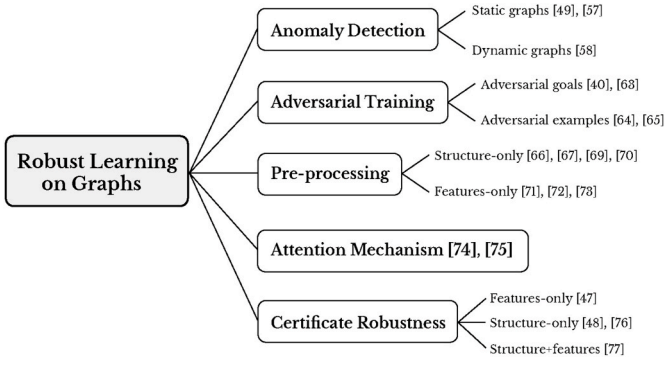

Fig. 1. The category of robust learning models on graphs.

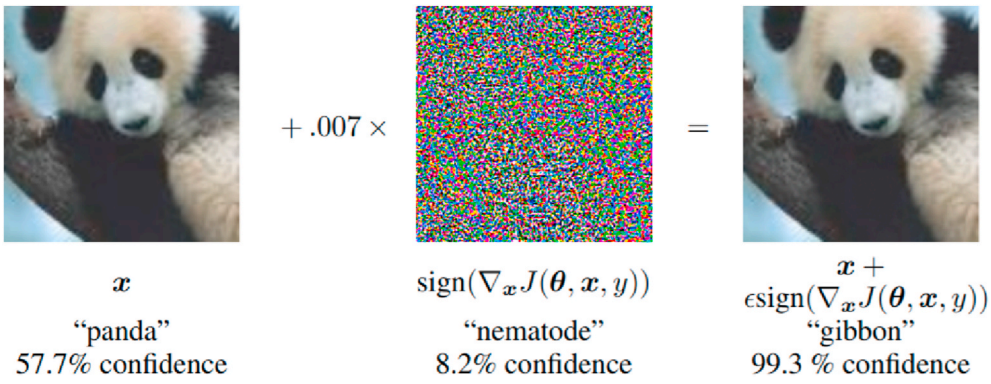

Fig. 2. A demonstration of adversarial example in image domain. By injecting a small perturbation, "panda" is classified as "gibbon". (Image Credit: (Goodfellow et al., 2014)).

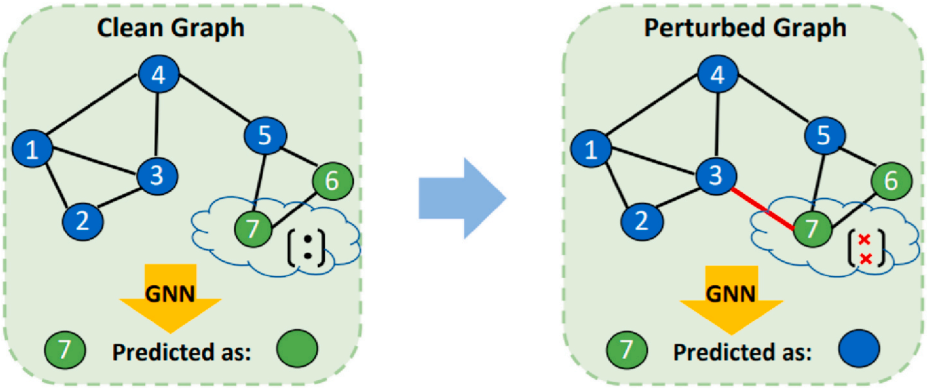

Fig. 3. An illustration of adversarial example in graph structure. By creating a new connection between node 3 and node 7 and modifying the features, originally green node 7 is predicted as blue one. (Image Credit: (Jin et al., 2020)).

$$\max_{\widehat{G}\in\Phi(G)} \sum_{\left(c_i,\widehat{\mathcal{G}}_i,y_i\right):i\in T(G)} \ell\left(f_{\theta^*}\left(c_i,\widehat{\mathcal{G}}_i\right),y_i\right) \tag{2}$$

$$s.t. \quad \theta^* = \underset{\theta}{\text{argmin}} \mathcal{L}_{train}(f_\theta(G')), \tag{3}$$

where $\widehat{G}$ denotes the perturbation set of G including adversarial graphs $\widehat{\mathcal{G}}_i = (\widehat{\mathcal{A}}_i, \widehat{\mathcal{X}}_i)$. As for target set, $T(G) = G$ holds in untargeted setting and $T(G)$ consists of targeted samples in targeted attacks. $G' = G$ represents the evasion attacks while $G' = \widehat{G}$ when the attacks are poisoned. Note that in most cases, the attacks should be limited in a constrained domain $\Phi(G)$ to ensure the perturbations are imperceptible. Formally, given the distance function d of $\mathcal{G}$ and the perturbation budget Δ, for any $\widehat{\mathcal{G}}_i \in \Phi(G)$, $\widehat{\mathcal{G}}_i$ should satisfy the constraint:

$$d\left(\widehat{\mathcal{G}}_i, \mathcal{G}_i\right) \leq \Delta. \tag{4}$$

Why to study graph robust learning. Significant success in a large number of applications (Goyal and Ferrara, 2018) has been promoted by deep learning on graphs, including molecule analysis (Hamilton et al., 2017), drug discovery (Gilmer et al., 2017), financial surveillance (Paranjape et al., 2017) and recommendation systems (Wang et al., 2019), etc. However, some works (Bojchevski and Günnemann, 2019a; Chen et al., 2020; Dai et al., 2018; Zügner and Günnemann, 2019a) have exposed the potential danger that these approaches are vulnerable to adversarial examples. In other words, the models are easy to be deceived by the attacks that are carefully designed to them by making subtle or even human-incomprehensible modifications to benign examples. Therefore, adversarial attacks themselves are serious security challenges for many real-world systems and identifying the weaknesses of these graph learning models to make them more robust to different kinds of attacks are very urgent.

3. Robustness metrics

We here introduce some metrics to measure the robustness of graph models. Note that in this section, we use $(\boldsymbol{a}, \boldsymbol{x})$ to denote an original example in the dataset where $\boldsymbol{a}$ is the adjacency matrix and $\boldsymbol{x}$ is the attribute matrix.

Classification margin. Classification margin is commonly used to measure whether a node can be correctly classified, which has also been utilized to measure the robustness associated with GNNs (Zügner and Günnemann, 2019b; Bojchevski and Günnemann, 2019b). This metric focuses on the label space which implies it changes for different downstream tasks. Besides, classification margin measures the robustness in a static perspective and the scope of investigation is limited in a dataset itself. For example, given a model, the most vulnerable example lies in the dataset which achieves the maximum value of the metric.

Definition 1. (Classification margin.) Let y^* denotes the ground truth class of the example $(\mathbf{a}, \mathbf{x})$, then the classification margin of $(\mathbf{a}, \mathbf{x})$ can be defined as:

$$CM(\boldsymbol{a}, \boldsymbol{x}, g, y^*) = \max_{y \in \mathcal{Y} \setminus \{y^*\}} \ln p(\hat{y} = y) - \ln p(\hat{y} = y^*),$$

where g is the classifier, $\hat{y} = g(\boldsymbol{a}, \boldsymbol{x})$ and $\mathcal{Y}$ denotes the label space. The smaller the value of $CM(\boldsymbol{a}, \boldsymbol{x}, g, y^*)$, the more robust g is w.r.t the example $(\boldsymbol{a}, \boldsymbol{x})$. **Adversarial risk and adversarial gap.** Drawn from the

definition of classification margin, we can define adversarial risk and adversarial gap which measure the vulnerability of a given model under input perturbations on the joint input space (Zhu et al., 2020). Different from the classification margin, these two metrics measure the robustness in a probability manner. More specifically, they will examine the continuous adversarial examples in a small budget, and they will consider the robustness of the encoder for the whole dataset instead of focusing on one specific example.

Definition 2. Let $(\mathscr{S}, d)$ denotes the input metric space. For any classification model $g : \mathscr{S} \to \mathscr{Y}$, we define the **adversarial risk** of g with the adversarial budget $\tau \geq 0$ as follows:

$$AdvRisk_\tau(g) = \mathbb{E}_{p(\boldsymbol{s},y^*)}\big[\exists\, \boldsymbol{s}' = (\boldsymbol{a}', \boldsymbol{x}') \in \mathscr{B}(\boldsymbol{s},\tau) \; s.t.\; CM(\boldsymbol{a}', \boldsymbol{x}', g, y^*) \geq 0\big],$$

where $\mathscr{B}(\mathbf{s},\tau) = \{\mathbf{s}' \in \mathscr{S} : d(\mathbf{s}', \mathbf{s}) \leq \tau\}$ represents the perturbation set of **s**. Based on $AdvRisk_\tau(g)$, the **adversarial gap** is defined to measure the relative vulnerability of a given model g w.r.t τ as:

$$AG_\tau(g) = AdvRisk_{\tau>0}(g) - AdvRisk_{\tau=0}(g).$$

4. Robust models on graphs

The vulnerability of graph learning models poses major challenges to the reliable and secure applications on graphs. we target the critical yet far overlooked aspect of learning robust models on graphs.

In this section, we divide existing works of robust models on graph into the following five categories, *i.e.*, (1) anomaly detection, (2) adversarial training, (3) pre-processing, (4) attention mechanism, and (5) certifiable robustness. Due to its importance and wide range of applications (Jiang et al., 2020; Akoglu et al., 2015), we specifically classify anomaly detection as a category; while others are mainly divided based on the technical characteristics.

In more details, anomaly detection and pre-processing methods are both used to correct the underlying attacked graph and obtain a more robust model training on the fixed graph. Both methods can defend poisoning attacks through identifying the attack methods or utilizing some prior assumptions to refine the graph. However, the methods are not in an end-to-end manner which is more time-consuming in the inference stage. As for attention mechanism, it aims to decrease the negative influence of attacks during the aggregation process in the presence of adversarial attacks. But this will cause extra learnable parameters and processing time to infer the downstream tasks. Furthermore, adversarial training and certifiable robustness apply different strategies to generate attacks from clean graphs to train the robust models on them, which is from an attacking-free perspective.

4.1. Anomaly detection

Anomaly detection is one of the most straight-forward ways to enhance the robustness of models and systems. The main idea of anomaly detection is to identify rare and unusual patterns which significantly differ from the majority of data. There are usually two main categories of anomalies in anomaly detection (Jiang et al., 2020):

- *Point anomalies.* Anomaly detection on point anomalies means to detect an individual anomalous data sample only respect to some of other data samples.
- *Contextual or collective anomalies.* Anomaly detection on contextual or collective anomalies means to detect a set of related or conditional anomalous data samples respect to the entire graph.

Within anomaly detection methods, identifying and removing anomalies from the source of data can increase robustness and reliability of models and systems constructed on these data. Many technologies in anomaly detection have been widely used in a number of real-world applications, *e.g.*, fraud detection (Yang et al., 2019a, 2019b), game bot detection (Tao et al., 2018; Xu et al., 2020a), intrusion detection (Khraisat et al., 2019), fault detection (Miljković, 2011), novelty detection (Pimentel et al., 2014).

Considering the inter-dependent and relational nature of graph-structured data, the anomaly information will propagate from nodes to their neighbors, leading to more destructive results. Hence, anomaly detection on graphs is much more challenging.

Graph anomaly detection techniques can effectively protect graph data from graph adversarial attacks by exploring the intrinsic difference between adversarial structures and the clean ones (Ioannidis et al., 2019a). There are four methods to distinguish graph adversarial attacks and help correctly detect adversarial perturbations (Jin et al., 2020), *i.e.*, (1) link prediction, (2) sub-graph link prediction, (3) graph generation, and (4) outlier detection.

Existing works of anomaly detection on graphs mainly focus on dealing with static graphs and dynamic graphs (Akoglu et al., 2015):

- *Anomaly detection on static graphs.* Given the snapshot of a graph database, the objective is to find the nodes, edges or sub-graphs that are rare and unusual in the graph.
- *Anomaly detection on dynamic graphs.* Given a sequence of graphs, the objective is to find the timestamps that correspond to a change, as well as the top-k nodes, edges or sub-graphs that contribute most to the change.

There exist plenty of works on static graphs. Jiang et al. (2020) design a graph convolution network model to detect both anomalous behaviors of individual users and associated malicious threat groups. As shown in Fig. 4, this model can characterize entities' properties as well as structural information between them into graphs, because only considering entities' properties information easily leads to high false positives. Because traditional anomaly detection methods such as one-class support vector machine (OCSVM) lost their effectiveness in graph data, Wang et al. (2020) propose one-class graph neural network (OCGNN) to combine the powerful representation ability of graph neural networks along with the classical one-class objective. As illustrated in Fig. 5, this hypersphere learning framework is a natural extension of OCSVM in the field of graph data.

Compared with static graphs, there exist only a few works on spotting anomalies by exploiting dynamic attributed graphs. Du et al. (2017) propose a deep neural network model, named DeepLog, utilizing Long Short-Term Memory (LSTM) to model a system log as a natural language sequence. The model architecture is shown in Fig. 6. DeepLog can automatically learn log patterns from normal execution and detect anomalies when log patterns deviate from the model trained from log data under normal execution. In addition, DeepLog is able to adapt to new log patterns over time and construct workflows from the underlying system log.

Even though there have been plenty of works in developing graph-based abnormality detection problems and algorithms, there are still some limitations of anomaly detection. In theoretical research, there exist only a few works on spotting anomalies by exploiting dynamic attributed graphs compared to plenty of works on static graphs. From systems perspective, most methods focus too much on detection performance while ignoring adversarial robustness. In view of practice, it is often hard to predict what would boost a detection algorithm's performance the most, the methods are not end-to-end and ground truth data is often inexistent.

4.2. Adversarial training

Adversarial training is an important way to enhance the robustness of neural network. The main idea of adversarial training is to insert slight perturbations into the training set and then retrain the model,

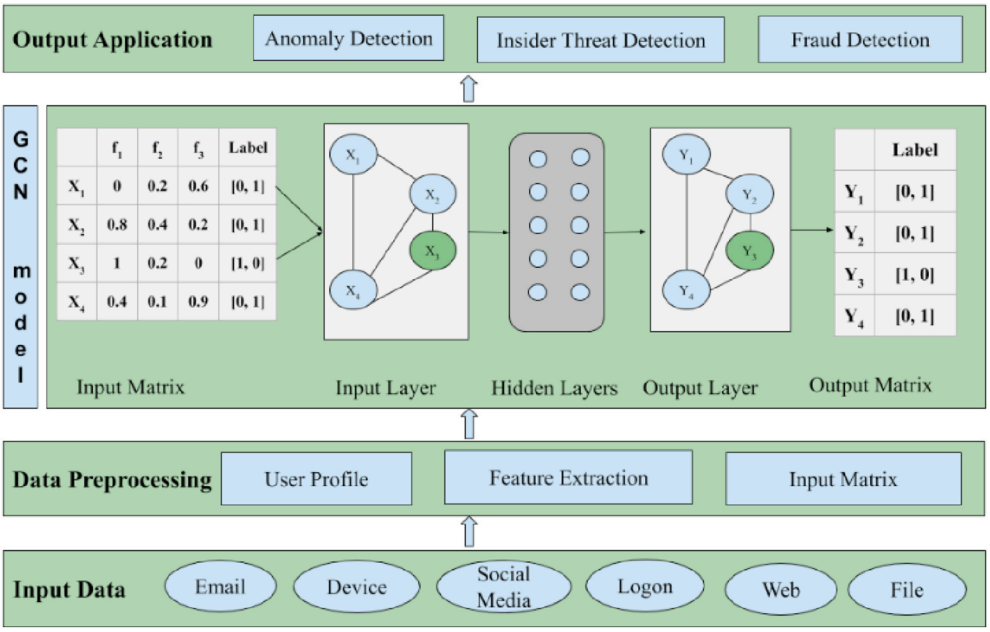

Fig. 4. Graph convolution network model for anomaly detection using graph data as input. (Image Credit: (Jiang et al., 2020)).

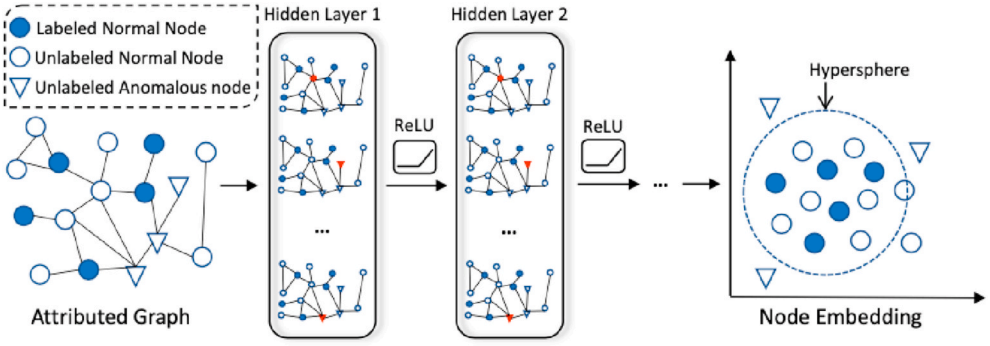

Fig. 5. The overall framework of OCGNN. (Image Credit: (Wang et al., 2020)).

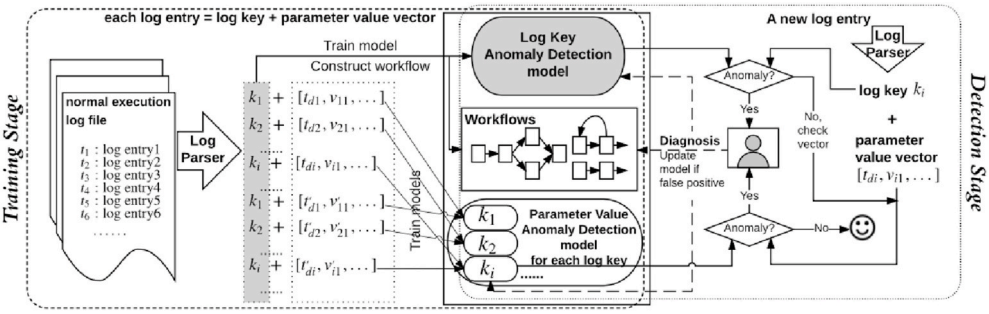

Fig. 6. The model architecture of DeepLog. (Image Credit: (Du et al., 2017)).

which normally has good performance on clean data.

In the image classification scenario, as illustrated in Fig. 2, these adversarial examples which look like the original images can fool the network. Generally, these results have often been interpreted as being a flaw in deep networks (Goodfellow et al., 2014). So studying adversarial examples in image data is thought to be extremely important. There are some training methods, such as FGSM (Goodfellow et al., 2014), Fast (Wong et al., 2020), TRADES (Zhang et al., 2019b), YOPO (Zhang et al., 2019c).

In graph domains, the attacker can modify the graph structure or node features to generate graph adversarial perturbations to mislead the prediction of GNN models. Since adversarial training has already been widely used in the image data, we can also take this strategy into consideration to defend graph adversarial attacks. There are two types of adversarial training: The first one is training with adversarial goals. Some adversarial training methods gradually optimize the model in a continuous min-max method under the guidance of two opposite (minimize and maximize) objective functions, as shown below (Jin

et al., 2020; Li et al., 2020),

$$\min_{\theta} \max_{\delta_A \in \mathscr{P}_A, \delta_X \in \mathscr{P}_X} \mathscr{L}_{train}(f_\theta(A+\delta_A, X+\delta_X)). \tag{5}$$

where δ_A, δ_X represent the perturbation added to A, X, respectively; $\mathscr{P}_A$, $\mathscr{P}_X$ denote the areas of unnoticeable perturbation.The min-max optimization problem in Eq (5) shows that graph adversarial training includes two processes: (1) maximize the prediction loss by adding perturbations and (2) minimize the prediction loss by retraining model to update parameters. Through the above two processes, we can get a robust model. Since there are two inputs, i.e., adjacency matrix A and feature matrix X, adversarial training can be done on them separately. The second one is training with adversarial examples. During the training process, other models based on the adversarial model are provided to the adversarial samples, which helps the model learn and adjust to adapt to the adversarial samples, thereby reducing the impact of these potential attack samples. For instance, Deng et al. (2019) proposed batch virtual adversarial training (BVAT) algorithms, which aim to generate virtual adversarial perturbations to perceive the connectivity patterns between nodes in the graph to improve the smoothness of the output distribution of the node classifier (shown in Fig. 7 and Fig. 8). Chen et al. (2019) proposed two special adversarial training strategies: global adversarial training (Global-AT) that for all nodes protection and target label adversarial training (Target-AT) that can protect the target labeled nodes from attack. In Global-AT, we select the target pair of nodes firstly, then update the adjacency matrix $\widehat{A}^{t-1}$ of the $(t-1)$ adversarial network and get the adjacency matrix $\widehat{A}^{t}$:

$$\widehat{A}^{t}_{ij} = \widehat{A}^{t-1}_{ij} + \theta_{ij}. \tag{6}$$

where $\widehat{A}^{t}_{ij}$ and $\widehat{A}^{t-1}_{ij}$ are the elements of $\widehat{A}^{t}$ and $\widehat{A}^{t-1}$. Target-AT only consider the target labeled nodes, and use the link selected by adversarial network attack to update the adversarial network.

4.3. Pre-processing

Both adversarial training or certifiable defense methods only aim at resisting evasion attacks, which means that the attack occurs during the test time. But poisoning attacks will insert several fake samples into the training set. Attackers usually prefer to add edges rather than remove edges or modify features, and tend to connect different nodes. This training process with fake samples can cause bad performance on the test data. Therefore, purifying the perturbed graph data and then training the GNN model on the clean graph data will get better results.

There are some models can be used for graph pre-processing before training normal graph models like GNNs. Xu et al. (2018) proposed different methods based on the graph generation model, and used link prediction as pre-processing to detect potential malicious edges. Zhang et al. (2019d) focused on the problem of detecting nodes which have been subject to topological perturbations calculated by the Nettack (Zügner et al., 2018). Through observing the discrepancy between the

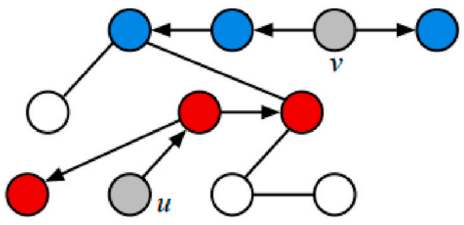

Fig. 7. In sample-based BVAT (S-BVAT), two nodes u and v are selected to calculate the LDS loss, and the virtual adversarial perturbation is applied to the elements that have no intersection in its acceptance area (marked in red and blue). (Image Credit: (Deng et al., 2019)).

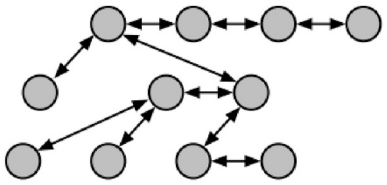

Fig. 8. In optimization-based BVAT (O-BVAT), all nodes are included to calculate LDS loss, and the virtual adversarial perturbation of all nodes is optimized together. (Image Credit: (Deng et al., 2019)).

first-order proximity information of v_i and the neighbors of v_i which created by Nettack, they using a relatively simple threshold test find the Nettack perturbations on GCN.

Similarly, in order to discard the high-rank perturbations generated by Nettack, Entezari et al. (2020) proposed the low-rank approximation and then retrain GCN with the low-rank approximation matrices (See Fig. 9).

Except for the above mentioned, GraphSAC filters out sets contaminated by abnormal nodes based on the graph-aware criterion calculated on a subset of nodes randomly, the formula is given as below (Ioannidis et al., 2019b):

$$\widehat{\mathbf{P}} = f\left(\{y_n\}_{n \in \mathscr{L}}, \mathbf{A}\right), \tag{7}$$

where y_n are sample labels at random subsets of nodes $n \in \mathscr{L} \subset \mathscr{V}$, $\mathbf{A}$ is the graph connectivity, and $\widehat{\mathbf{P}}_{(n,c)} \in [0,1]$ can be denoted as the probability that $y_n = c$. The choice of $f(\cdot)$ is determined by the specific attributes it wants to capture. Then, GraphSAC compares the accuracy of $f(\cdot)$ using the ratio of nodes in the consensus set to a prespecified threshold T to judge it whether contain anomalies.

These models only rely on network topology for attack detection. On attributed graphs, based on the observations that attackers prefer adding edges than removing edges and the edges are often added between dissimilar nodes. Based on these findings, Xu et al. (2020c) sampled sub-graphs from the poisoned training data and then used outlier detection methods to detect and filter adversarial edges. And Wu et al. (2019) proposed a defense method by eliminating the edges whose two end nodes have small Jaccard Similarity, the Jaccard Similarity score is given as (Said et al., 2010):

$$J_{u,v} = \frac{M_{11}}{M_{01} + M_{10} + M_{11}}. \tag{8}$$

where M_{11} represents the number of features where both node u and node v have a value of 1. Similarly, M_{10}, M_{01}, M_{00} represent the number of feature values of node u and node v, 1 and 0, 0 and 1, 0 and 0, respectively.

4.4. Attention mechanism

Different from pre-processing methods which try to purify the perturbed graph data to enhance the robustness of GNN models, attention-based models aim to improve the robustness of GNNs in the presence of adversarial attacks. More specifically, the designed attention mechanism are trained to distinguish the adversarial edges and nodes with the clean ones. When aggregating the information from neighbors, the learned attention weights will penalize the perturbed part of data through making them contribute less during the propagation process.

RGCN (Zhu et al., 2019) makes the assumption that adversarial nodes may have high prediction uncertainty. From Fig. 10, since the plain vectors cannot adapt to the abnormal changes, RGCN proposes to model the hidden representations of nodes in all graph convolutional layers as Gaussian distributions to automatically reflect the effects of adversarial changes in the variances. As a result, the variance-based attention mechanism will penalize the nodes with high variance to

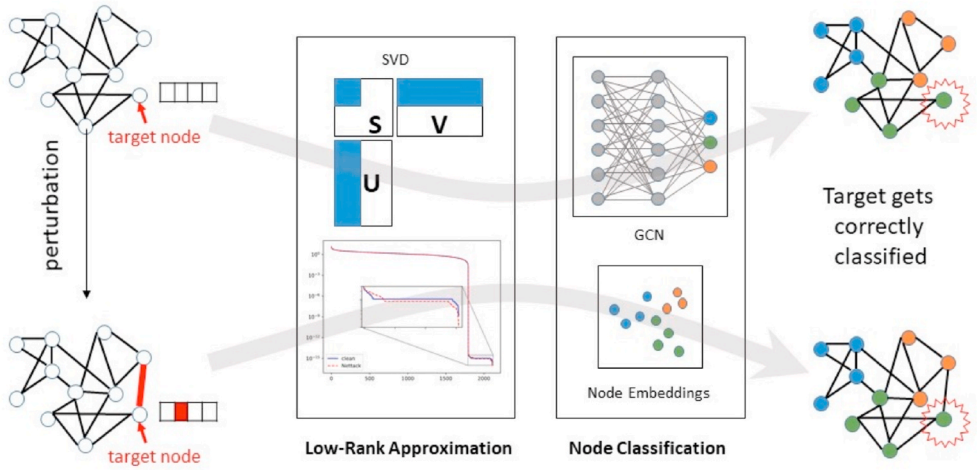

Fig. 9. The overall system: low-rank approximation of graph structure and feature matrices to vaccinate the node classification method and discard high-rank perturbations. (Image Credit: (Entezari et al., 2020)).

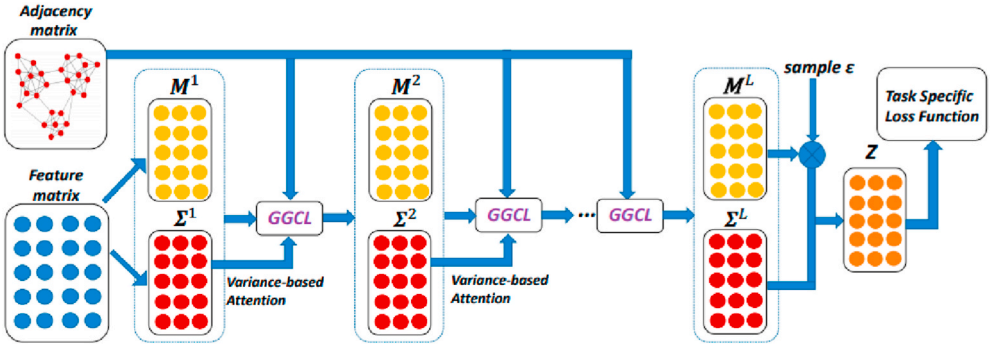

Fig. 10. The framework of RGCN. And the GGCL represents the Gaussian-base graph convolutional layer. (Image Credit: (Zhu et al., 2019)).

help mitigate the propagation of negative impact caused by adversarial examples. The attention weights of node v_j in layer l are defined as

$$\boldsymbol{\alpha}_j^{(l)} = \exp\left(-\gamma \boldsymbol{\sigma}_j^{(l)}\right), \tag{9}$$

where $\boldsymbol{\sigma}_j^{(l)}$ denotes the variance and γ is a hyper-parameter.

PA-GNN (Tang et al., 2020) introduces the supervised information about real perturbations in a poisoned graph to help improve the robustness of target GNN models. The intuition is from the fact that there usually exist clean graphs sharing the similar topological distributions and node attributes with the poisoned graph. For example, co-review networks like Yelp and Foursquare and social networks like Facebook and Twitter both share similar domains. Therefore, PA-GNN first learns to discriminate adversarial edges generated by attacking the clean graphs with supervised knowledge of known perturbations. With supervision knowledge, PA-GNN designs a loss function to guarantee less attention weights for adversarial edges as

$$\mathscr{L}_{dist} = -\min\left(\eta, \mathop{\mathbb{E}}_{e_{ij}\in\mathscr{E}\setminus\mathscr{P}} \boldsymbol{\alpha}_{ij}^l - \mathop{\mathbb{E}}_{e_{ij}\in\mathscr{P}} \boldsymbol{\alpha}_{ij}^l\right), \tag{10}$$

where $\mathscr{E}$ and $\mathscr{P}$ represents the set of all edges and that of perturbed edges and $\boldsymbol{\alpha}_{ij}^l$ denotes the self-attention coefficient assigned for e_{ij} on the l-th layer. η is a hyper-parameter controlling the margin between the expectations of two distributions. Then a meta-optimization algorithm is proposed to train the initialization of PA-GNN and further fine-tunes the model on the poisoned graph to achieve robustness.

4.5. Certifiable robustness

In most previous works, the robustness of GNNs is exploited heuristically and experimentally. However, the criteria of measuring the safety of input graphs under adversarial perturbation is not solved in the previous works. Therefore, to research the problem that how to verify that small perturbations to input data will not cause dramatic effect to a GNN is important (see Fig. 11).

In (Zügner and Günnemann, 2019b), they try to derive an efficient principle for robustness certificates. More specifically, they want to provide a certificate to measure that for which nodes the given trained GNN can guarantee that the predictions will not change under any admissible perturbations given a specific attack budget (see Fig. 13). To tackle this problem, they aim to find the worst case margin (see Fig. 12) for the node t under some set $\mathscr{X}_{q,Q}(\dot{X})$ of admissible perturbations to the node attributes:

$$m^t(y^*, y) := \min_{\widetilde{X}} f_\theta^t\left(\widetilde{X}, \dot{A}\right)_{y^*} - f_\theta^t\left(\widetilde{X}, \dot{A}\right)_y \tag{11}$$

$$\text{s.t.} \widetilde{X} \in \mathscr{X}_{q,Q}\left(\dot{X}\right), \tag{12}$$

where y^* denotes the class of node t given by the ground truth or predicted and $f_\theta^t(\cdot)$ represents the classifier which outputs the logits of each class. It is easy to see that the GNN is certifiably robust *w.r.t* node t when $m^t(y^*, y) > 0$ for all $y \neq y^*$, which means there exists no adversarial examples that can change its prediction for node t. Through some relaxations, they obtain a lower bound of $m^t(y^*, y)$ which is tractable to calculate. Thus, they can use this certificate to find how many nodes in a

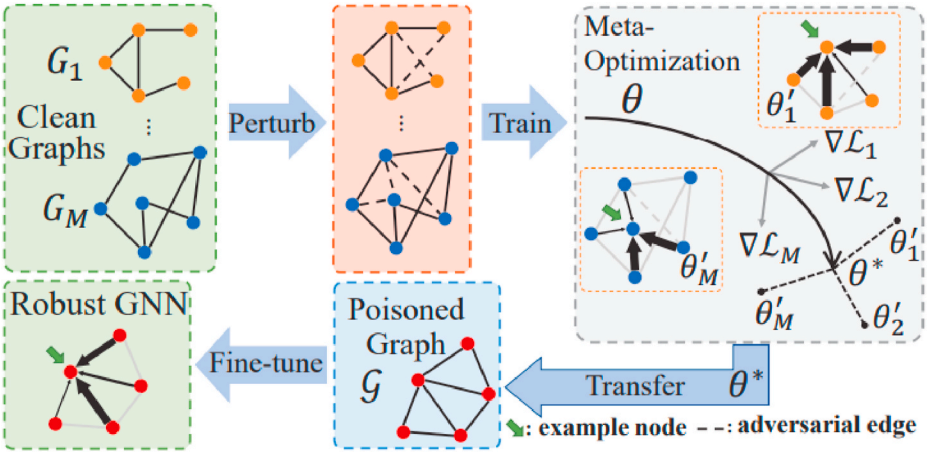

Fig. 11. Overall framework of PA-GNN. Thicker arrows indicate higher attention coefficients. θ^* denotes the model initialization from meta-optimization. (Image Credit: (Tang et al., 2020)).

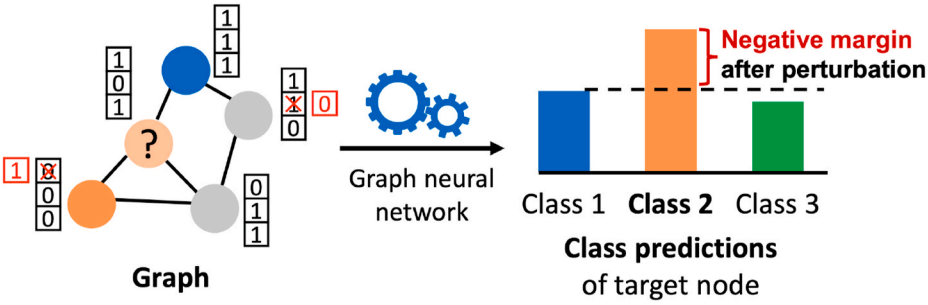

Fig. 12. Intuitive idea of the classification margin (Zügner and Günnemann, 2019b).

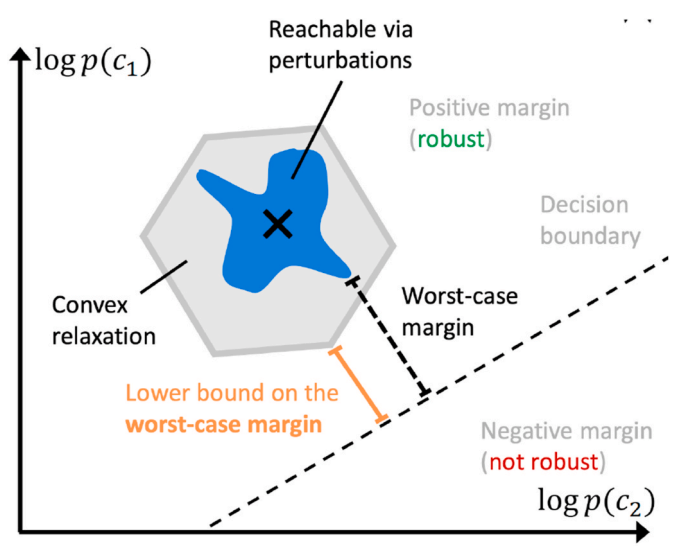

Fig. 13. Illustration of certifiable robustness on graphs (Zügner and Günnemann, 2019b).

graph is certifiably safe. Furthermore, the certificate can be taken as the objective to help more nodes safer through maximizing the worst case margin.

However, Zügner et al. (Zügner and Günnemann, 2019b) only considers perturbations to the node features. Bojchevski et al. (Bojchevski and Günnemann, 2019b) is completely orthogonal to (Zügner and Günnemann, 2019b) since they consider adversarial perturbations only to the graph structure instead. This work derives the robustness certificates for the models where the prediction is a linear function of (personalized) PageRank. Based on the observation, the work transforms robustness certificates to worst-case margin of node t between class y_t and class c under any admissible perturbation $\widetilde{G} \in \mathscr{C}_{\mathscr{T}}$:

$$m^*_{y_t,c}(t) = \min_{G \in \mathscr{C}_{\mathscr{T}}} m_{y_t,c}(t) \tag{13}$$

$$= \min_{G \in \mathscr{C}_{\mathscr{T}}} \boldsymbol{\pi}_{\widetilde{G}}(\boldsymbol{e}_t)^T \left(\boldsymbol{H}_{:,y_t} - \boldsymbol{H}_{:,c}\right), \tag{14}$$

where $\boldsymbol{H}_{:,y_t}$ and $\boldsymbol{H}_{:,c}$ denote the prediction logits vectors of class y_t and class c respectively. And $\boldsymbol{\pi}_{\widetilde{G}}(\boldsymbol{e}_t)$ is the personalized PageRank vector of node t. Then they aim to find a set of fragile edges into included/excluded to obtain a perturbed graph $\widetilde{G}$ maximizing the margin. Furthermore, inspired by the Markov decision process (MDP), they reformulate the problem as a non-convex Quadratically Constrained Linear Program (QCLP) to be able to handle the global budget; They utilize the Reformulation Linearization Technique (RLT) to construct a convex relaxation of the QCLP, enabling to efficiently compute a lower bound on the worst-case margin. As an extension of (Bojchevski and Günnemann, 2019b), Zügner et al. (Zügner and Günnemann, 2020) covers the highly important principle of graph convolutional networks. They rephrases the objective function as a jointly constrained bilinear program to make the optimization tractable.

Bojchevski et al. (2020) proposes an approach that can handle both types of perturbations and be applied to any GNN utilizing randomized smoothing framework. In this framework, the certificate is defined as:

$$\rho_{\boldsymbol{x},\boldsymbol{x}}(p,y) = \min_{\substack{h \in \mathscr{H}: \\ \Pr(h(\varphi(\boldsymbol{x}))=y)=p}} \Pr(h(\varphi(\widetilde{\boldsymbol{x}})) = y), \tag{15}$$

where $\tilde{x}$ is a given neighboring point, and $\mathscr{H}$ is the set of measurable classifiers with respect to φ. φ is a randomization scheme to be specified, which assigns probability mass $\Pr(\varphi(x) = z)$ for each randomized outcome z. h is a base classifier outputting a single prediction class. Based on this definition, they define the data-dependent sparsity-aware noise distribution:

$$\Pr(\varphi(\boldsymbol{x})_i \neq \boldsymbol{x}_i) = p_-^{\boldsymbol{x}_i} p_+^{(1-\boldsymbol{x}_i)}, \tag{16}$$

where the randomization scheme φ deletes an existing edge with probability p_-, and similarly adds a new edge with probability p_+. Through theoretic analysis and further relaxation, they conduct experiments to verify the effectiveness and efficiency of their algorithm. This work also gives the certificates for graph-level classification models for the first time. Except for node and graph classification tasks, there are also other works concentrating on certifiable robustness on other applications such as community detection (Jia et al., 2020).

5. Future directions

We have thoroughly investigated robust models on graphs and gained an overview of this emerging research field, robust learning on graphs. The deep understandings of this area allows us to discuss some promising research directions.

- *Graph data.* Compared with considerable amount of work on static graphs, there still remain problems on dynamic graphs, *e.g.*, detecting anomalies on attributed dynamic graphs, work with the trace of edge or node updates. Besides, when it comes to an explicit graph representation, to add or remove latent edges may also be possible, that is, augmented graphs, *e.g.*, edges based on similarities or domain knowledge.
- *Graph construction.* The data does not form a network or there is more than one network available. To use graph-based techniques, how to use the source of data to construct a best representation, a graph or multi-graphs, remains an open problem.
- *Balance performance.* Most methods focus on anomaly detection performance while ignoring adversarial robustness. How to balance detection performance and the robustness of models is still an open challenge.
- *Evaluation.* Ground truth data is often inexistent and humans cannot easily tell whether adversarial perturbations on graph data are imperceptible or not, thus to find concise evaluation measure is urgent.

6. Conclusion

In this survey, we conduct a comprehensive review on robust learning models on graphs. Specifically, we present the recent developments of this area, we first provide some robustness evaluation metrics of model robustness on graphs, and then comprehensively divide existing works of robust models on graphs into five categories: anomaly detection, adversarial training, pre-processing, attention mechanism, and certifiable robustness. Besides, we further emphasize some potential future directions in learning robust models on graphs.

We hope our works can serve as a reference to help researchers get a systematical and comprehensive understanding of robust models on graph, thus providing more insights for their studies.

Acknowledgements

This work is supported by the National Key Research and Development Project of China (No. 2018AAA0101900).

References

Akoglu, L., Tong, H., Koutra, D., 2015. Graph-based anomaly detection and description: a survey. Data Min. Knowl. Discov. 29 (3), 626–688.
Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., et al., 2016. End to End Learning for Self-Driving Cars arXiv preprint arXiv:1604.07316.
Bojchevski, A., Günnemann, S., 2019a. Adversarial attacks on node embeddings via graph poisoning. In: Proceedings of the 36th International Conference on Machine Learning. ICML 2019, Long Beach, California, USA, 9-15 June 2019.
Bojchevski, A., Günnemann, S., 2019b. Certifiable robustness to graph perturbations. In: Advances in Neural Information Processing Systems, pp. 8319–8330.
Bojchevski, A., Klicpera, J., Günnemann, S., 2020. Efficient robustness certificates for discrete data: sparsity-aware randomized smoothing for graphs, images and more. In: International Conference on Machine Learning (ICML), pp. 11647–11657.
Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O., 2013. Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, pp. 2787–2795.
Bruna, J., Zaremba, W., Szlam, A., LeCun, Y., 2013. Spectral Networks and Locally Connected Networks on Graphs arXiv preprint arXiv:1312.6203.
Butts, C.T., 2003. Network inference, error, and informant (in)accuracy: a bayesian approach. Soc. Network. 25 (2), 103–140.
Cao, S., Lu, W., Xu, Q., 2016. Deep neural networks for learning graph representations. in *AAAI* 16, 1145–1152.
Carlini, N., Wagner, D., 2018. Audio adversarial examples: targeted attacks on speech-to-text. In: 2018 IEEE Security and Privacy Workshops (SPW). IEEE, pp. 1–7.
Chen, J., Wu, Y., Lin, X., Xuan, Q., 2019. Can adversarial network attack be defended? CoRR vol. abs/1903.05994.
Chen, L., Li, J., Peng, J., Xie, T., Cao, Z., Xu, K., He, X., Zheng, Z., 2020. A survey of adversarial learning on graphs. ArXiv.
Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al., 2016. Wide & deep learning for recommender systems. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, pp. 7–10.
Dai, H., Li, H., Tian, T., Huang, X., Wang, L., Zhu, J., Song, L., 2018. Adversarial attack on graph structured data. In: Proceedings of the 35th International Conference on Machine Learning. ICML 2018, Stockholmsmässan, Stockholm, Sweden. July 10-15, 2018.
Defferrard, M., Bresson, X., Vandergheynst, P., 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems, pp. 3844–3852.
Deng, Z., Dong, Y., Zhu, J., 2019. Batch Virtual Adversarial Training for Graph Convolutional Networks.
Du, M., Li, F., Zheng, G., Srikumar, V., 2017. Deeplog: anomaly detection and diagnosis from system logs through deep learning. In: Acm Sigsac Conference on Computer & Communications Security.
Ebrahimi, J., Rao, A., Lowd, D., Dou, D., 2017. Hotflip: White-box Adversarial Examples for Text Classification arXiv preprint arXiv:1712.06751.
Entezari, N., Al-Sayouri, A.S., Darvishzadeh, A., Papalexakis, E.E., 2020. All you need is low (rank) - defending against adversarial attacks on graphs. In: WSDM '20: the Thirteenth ACM International Conference on Web Search and Data Mining Houston TX USA February, 2020, pp. 169–177.
Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E., 2017. Neural message passing for quantum chemistry. In: Proceedings of the 34th International Conference on Machine Learning. ICML 2017, Sydney, NSW, Australia, 6-11 August 2017.
Goodfellow, I.J., Shlens, J., Szegedy, C., 2014. Explaining and Harnessing Adversarial Examples arXiv preprint arXiv:1412.6572.
Goyal, P., Ferrara, E., 2018. Graph embedding techniques, applications, and performance: a survey. Knowl. Base Syst. 151.
Grosse, K., Papernot, N., Manoharan, P., Backes, M., McDaniel, P., 2017. Adversarial examples for malware detection. In: European Symposium on Research in Computer Security. Springer, pp. 62–79.
Grover, A., Leskovec, J., 2016. node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864.
Gueorgi, K., 2006. Effects of missing data in social networks. Soc. Network. 28 (3), 247–268.
Guo, H., Tang, R., Ye, Y., Li, Z., He, X., 2017. Deepfm: a Factorization-Machine Based Neural Network for Ctr Prediction arXiv preprint arXiv:1703.04247.
Hamilton, W.L., Ying, Z., Leskovec, J., 2017. Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, 4-9 December 2017, Long Beach, CA, USA, 2017.
Henaff, M., Bruna, J., LeCun, Y., 2015. Deep Convolutional Networks on Graph-Structured Data arXiv preprint arXiv:1506.05163.
Ioannidis, V.N., Berberidis, D., Giannakis, G.B., 2019a. Graphsac: Detecting Anomalies in Large-Scale Graphs.
Ioannidis, V.N., Berberidis, D., Giannakis, G.B., 2019b. Graphsac: Detecting Anomalies in Large-Scale Graphs arXiv preprint arXiv:1910.09589.
Jia, J., Wang, B., Cao, X., Gong, N.Z., 2020. Certified robustness of community detection against adversarial structural perturbation via randomized smoothing. In: Proceedings of the Web Conference 2020, pp. 2718–2724.
Jiang, J., Chen, J., Gu, T., Choo, K.K.R., Liu, C., Yu, M., Huang, W., Mohapatra, P., 2020. Anomaly detection with graph convolutional networks for insider threat and fraud detection. In: MILCOM 2019 - 2019 IEEE Military Communications Conference (MILCOM).

Jin, W., Li, Y., Xu, H., Wang, Y., Tang, J., 2020. Adversarial Attacks and Defenses on Graphs: A Review and Empirical Study arXiv preprint arXiv:2003.00653.
Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., 2019. Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity 2 (1), 20.
Kipf, T.N., Welling, M., 2016. Semi-supervised Classification with Graph Convolutional Networks arXiv preprint arXiv:1609.02907.
Levie, R., Monti, F., Bresson, X., Bronstein, M.M., Cayleynets, ", 2018. Graph convolutional neural networks with complex rational spectral filters. IEEE Trans. Signal Process. 67 (1), 97–109.
Li, Y., Jin, W., Xu, H., Tang, J., 2020. Deeprobust: A Pytorch Library for Adversarial Attacks and Defenses arXiv preprint arXiv:2005.06149.
Miljković, D., 2011. Fault Detection Methods: A Literature Survey, pp. 750–755, 05.
Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., Bronstein, M.M., 2017. Geometric deep learning on graphs and manifolds using mixture model cnns. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5115–5124.
Namata Jr., G.M.S., Getoor, L., 2009. Identifying graphs from noisy and incomplete data. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD'09, New York, NY, USA, pp. 23–29. ACM.
Niepert, M., Ahmed, M., Kutzkov, K., 2016. Learning convolutional neural networks for graphs. In: International Conference on Machine Learning, pp. 2014–2023.
Pandit, S., Chau, D.H., Wang, S., Faloutsos, C., 2007. Netprobe: a fast and scalable system for fraud detection in online auction networks. In: Proceedings of the 16th International Conference on World Wide Web, pp. 201–210.
Papineni, K., Roukos, S., Ward, T., Zhu, W.-J., 2002. Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318.
Paranjape, A., Benson, A.R., Leskovec, J., 2017. Motifs in temporal networks. In: Proceedings of the 10th ACM International Conference on Web Search and Data Mining. WSDM 2017, Cambridge, United Kingdom. February 6-10, 2017.
Patel, J., Shah, S., Thakkar, P., Kotecha, K., 2015. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. Expert Syst. Appl. 42 (1), 259–268.
Perozzi, B., Al-Rfou, R., Skiena, S., 2014. Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '14, New York, NY, USA), pp. 701–710. ACM.
Pimentel, M.A., Clifton, D.A., Clifton, L., Tarassenko, L., 2014. A review of novelty detection. Signal Process. 99, 215–249.
Ribeiro, L.F., Saverese, P.H., Figueiredo, D.R., 2017. struc2vec: learning node representations from structural identity. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 385–394.
Said, A., De Luca, E.W., Albayrak, S., 2010. How social relationships affect user similarities. In: Proc. Of the 2010 Workshop on Social Recommender Systems, pp. 1–4.
Schafer, J., Graham, J., 2002. Missing data: our view of the state of the art. Psychol. Methods 7, 147–177, 06.
Sun, L., Wang, J., Yu, P.S., Li, B., 2018a. Adversarial attack and defense on graph data: a survey. CoRR, 10528 vol. abs/1812.
Sun, L., Dou, Y., Yang, C., Wang, J., Yu, P.S., Li, B., 2018b. Adversarial Attack and Defense on Graph Data: A Survey arXiv preprint arXiv:1812.10528.
Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q., 2015. Line: large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web, pp. 1067–1077.
Tang, X., Li, Y., Sun, Y., Yao, H., Mitra, P., Wang, S., 2020. Transferring robustness for graph neural network against poisoning attacks. In: Proceedings of the 13th International Conference on Web Search and Data Mining, pp. 600–608.
Tao, J., Xu, J., Gong, L., Li, Y., Fan, C., Zhao, Z., Nguard, ", 2018. A game bot detection framework for netease mmorpgs. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 811–820.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008.
Wang, H., Zhang, F., Zhang, M., Leskovec, J., Zhao, M., Li, W., Wang, Z., 2019. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD 2019, Anchorage, AK, USA. August 4-8, 2019.
Wang, X., Jin, B., Du, Y., Cui, P., Yang, Y., 2020. One-class Graph Neural Networks for Anomaly Detection in Attributed Networks.
Wong, E., Rice, L., Kolter, J.Z., 2020. Fast Is Better than Free: Revisiting Adversarial Training arXiv preprint arXiv:2001.03994.
Wu, H., Wang, C., Tyshetskiy, Y., Docherty, A., Lu, K., Zhu, L., 2019. Adversarial Examples on Graph Data: Deep Insights into Attack and Defense arXiv preprint arXiv:1903.01610.
Xiaojin, Z., Zoubin, G., 2002. Learning from Labeled and Unlabeled Data with Label Propagation.
Xu, X., Yu, Y., Li, B., Song, L., Liu, C., Gunter, C., 2018. Characterizing Malicious Edges Targeting on Graph Neural Networks.
Xu, J., Luo, Y., Tao, J., Fan, C., Zhao, Z., Lu, J., 2020a. Nguard+ an attention-based game bot detection framework via player behavior sequences. ACM Trans. Knowl. Discov. Data 14 (6), 1–24.
Xu, J., Yang, Y., Wang, C., Liu, Z., Zhang, J., Chen, L., Lu, J., 2020b. Robust network enhancement from flawed networks. IEEE Trans. Knowl. Data Eng. 1–1.
Xu, X., Yu, Y., Song, L., Liu, C., Kailkhura, B., Gunter, C., Li, B., 2020c. "Edog: Adversarial Edge Detection for Graph Neural Networks," Tech. Rep. Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States).
Yang, Y., Xu, Y., Wang, C., Sun, Y., Wu, F., Zhuang, Y., Gu, M., 2019a. Understanding default behavior in online lending. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 2043–2052.
Yang, Y., Xu, Y., Sun, Y., Dong, Y., Wu, F., Zhuang, Y.T., 2019b. Mining fraudsters and fraudulent strategies in large-scale mobile social networks. In: IEEE Transactions on Knowledge and Data Engineering.
Zhang, F., Liu, X., Tang, J., Dong, Y., Yao, P., Zhang, J., Gu, X., Wang, Y., Shao, B., Li, R., , et al.Oag, ", 2019a. Toward linking large-scale heterogeneous entity graphs. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2585–2595.
Zhang, H., Yu, Y., Jiao, J., Xing, E.P., Ghaoui, L.E., Jordan, M.I., 2019b. Theoretically Principled Trade-Off between Robustness and Accuracy arXiv preprint arXiv:1901.08573.
Zhang, D., Zhang, T., Lu, Y., Zhu, Z., Dong, B., 2019c. arXiv preprint arXiv:1905.00877. You Only Propagate once: Painless Adversarial Training Using Maximal Principle, vol. 2. no. 3.
Zhang, Y., Khan, S., Coates, M., 2019d. Comparing and detecting adversarial attacks for graph deep learning. In: Proc. Representation Learning on Graphs and Manifolds Workshop, Int. Conf. Learning Representations, New Orleans, LA, USA.
Zhu, D., Zhang, Z., Cui, P., Zhu, W., 2019. Robust graph convolutional networks against adversarial attacks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1399–1407.
Zhu, S., Zhang, X., Evans, D., 2020. Learning adversarially robust representations via worst-case mutual information maximization. In: International Conference on Machine Learning (ICML).
Zügner, D., Günnemann, S., 2019a. Adversarial attacks on graph neural networks via meta learning. In: 7th International Conference on Learning Representations. ICLR 2019, New Orleans, LA, USA. May 6-9, 2019.
Zügner, D., Günnemann, S., 2019b. Certifiable robustness and robust training for graph convolutional networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 246–256.
Zügner, D., Günnemann, S., 2020. Certifiable robustness of graph convolutional networks under structure perturbations. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1656–1665.
Zügner, D., Akbarnejad, A., Günnemann, S., 2018. Adversarial attacks on neural networks for graph data. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2847–2856.