All You Need Is Low (Rank): Defending Against Adversarial Attacks on Graphs

Negin Entezari
Department of Computer Science & Engineering
University of California Riverside
nente001@ucr.edu

Saba A. Al-Sayouri
Systems Science & Industrial Engineering
State University of New York at Binghamton
ssyouri1@binghamton.edu

Amirali Darvishzadeh
Department of Computer Science & Engineering
University of California Riverside
adarv001@ucr.edu

Evangelos E. Papalexakis
Department of Computer Science & Engineering
University of California Riverside
epapalex@cs.ucr.edu

ABSTRACT

Recent studies have demonstrated that machine learning approaches like deep learning methods are easily fooled by adversarial attacks. Recently, a highly-influential study examined the impact of adversarial attacks on graph data and demonstrated that graph embedding techniques are also vulnerable to adversarial attacks. Fake users on social media and fake product reviews are examples of perturbations in graph data that are realistic counterparts of the adversarial models proposed. Graphs are widely used in a variety of domains and it is highly important to develop graph analysis techniques that are robust to adversarial attacks. One of the recent studies on generating adversarial attacks for graph data is NETTACK. The NETTACK model has shown to be very successful in deceiving the Graph Convolutional Network (GCN) model. NETTACK is also transferable to other node classification approaches e.g. node embeddings. In this paper, we explore the properties of NETTACK perturbations, in search for effective defenses against them. Our first finding is that NETTACK demonstrates a very specific behavior in the *spectrum* of the graph: only high-rank (low-valued) singular components of the graph are affected. Following that insight, we show that a low-rank approximation of the graph, that uses only the top singular components for its reconstruction, can greatly reduce the effects of NETTACK and boost the performance of GCN when facing adversarial attacks. Indicatively, on the CiteSeer dataset, our proposed defense mechanism is able to reduce the success rate of NETTACK from 98% to 36%. Furthermore, we show that tensor-based node embeddings, which by default project the graph into a low-rank subspace, are robust against NETTACK perturbations. Lastly, we propose *LowBlow*, a low-rank adversarial attack which is able to affect the classification performance of both GCN and tensor-based node embeddings and we show that the low-rank attack is noticeable and making it unnoticeable results in a high-rank attack.

WSDM '20, February 3–7, 2020, Houston, TX, USA

ACM ISBN 978-1-4503-6822-3/20/02…$15.00
https://doi.org/10.1145/3336191.3371789

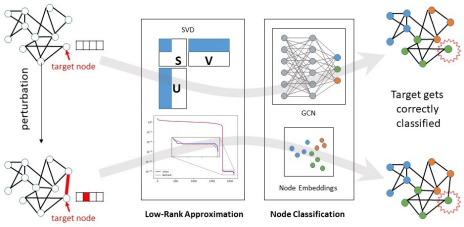


Figure 1: System Overview: low-rank approximation of graph structure and feature matrices to vaccinate the node classification method and discard perturbations.

KEYWORDS

Adversarial machine learning, graph mining, graph convolutional networks, graph representation learning, tensors

ACM Reference Format:
Negin Entezari, Saba A. Al-Sayouri, Amirali Darvishzadeh, and Evangelos E. Papalexakis. 2020. All You Need Is Low (Rank): Defending Against Adversarial Attacks on Graphs. In *The Thirteenth ACM International Conference on Web Search and Data Mining (WSDM '20), February 3–7, 2020, Houston, TX, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3336191.3371789

1 INTRODUCTION

Graphs are widely used because of their strength in representing real-world data in many domains, such as social networks, biological networks, and citation networks. Due to the ubiquity of graphs, analyzing them has gained significant attention in recent years. An important task in analyzing graph data is node classification. Given a partially labeled (attributed) graph, the goal is to classify the entire graph and predict the labels of the unknown nodes [4]. Graph representation learning [16, 30] and deep learning techniques [19, 32] have shown outstanding results in addressing the problem of node classification.

However, machine leaning models often suffer from vulnerabilities to adversarial perturbations [12]. In spite of the popularity and success of deep learning architectures, they have shown to be vulnerable to adversarial attacks [36]. Subtle perturbations of the data can be imperceptible, yet lead to wrong results. Even when the attacker does not have full knowledge of the network architecture, they are still able to perturb the data and affect the learning outcome.

Utilizing machine learning methods which are vulnerable to adversarial attacks have raised many concerns. Recent studies have addressed this concern and conducted research to analyze vulnerability of machine learning algorithms and also develop defense techniques and methods that are more robust to attacks [5, 13, 29]. However, only a few studies have investigated adversarial attacks in graph data [7, 11, 35, 38]. Graph Convolutional Networks (GCN) have shown great success in node classification task because of their non-linear nature and exploiting relational information of nodes [19]. Despite their success, they suffer from vulnerabilities against small perturbations. Changes to one node can lead to misclassification of other nodes in the graph [38]. Writing wrong or biased reviews on websites like Amazon and fake users on social networks are examples of adversarial attacks on graph data. Such activities aim to mislead machine learning techniques. Therefore, adversarial machine learning studies play a crucial role in graph domain and their goal is to detect and defend attacks and also introduce techniques that are more robust against perturbations.

One of the most prominent studies on generating adversarial attacks for graphs is called Nettack [38] proposed by Zügner et al. Their work shows that the classification performance of GCN drops significantly when Nettack perturbations target a node. In this paper, we investigate the properties of poisoning adversarial attacks generated by the Nettack algorithm and propose a method to defend against attacks and "vaccinate" the network. The term "vaccinate" was first introduced in [13] to describe a network equipped with defense mechanism against adversarial attacks.

Our contributions are as follows:

(1) **Nettack is a high-rank attack**: We explore the characteristics of Nettack perturbations and we show that these attacks result in changes in high-rank spectrum of the graph, which corresponds to low singular values.
(2) **Vaccinating GCN with low-rank approximations**: Building on the idea that the Nettack perturbations are high-rank, we show that the GCN model can significantly resist the attacks when a low-rank approximation of the graph is used.
(3) **Tensor-based node embeddings are robust to Nettack**: Adversarial attacks generated by Nettack model are transferable to other node classification approaches. Recently, a tensor-based node embedding technique has been proposed [2], which computes a low-rank representation of the graph. We exploit the effects of these attacks on a tensor-based node embedding method and we show that tensor-based node embeddings are very robust to adversarial attacks.
(4) ***LowBlow*: low-rank attack is noticeble**: Tensor-based node embeddings and vaccinated inputs to a GCN are robust against high-rank attacks. But what happens if the attack results in low-rank perturbations to the graph? We introduce the *LowBlow* attack, which modifies the Nettack perturbations so that it affects low-rank components of the graph and therefore, the new low-rank attack is able to fool both GCN and tensor-based embeddings. We show that the degree sequences of the graph after the proposed low-rank attack and the input graph are from different power-law distributions and therefore the attack is noticeable. We also show that modifying the low-rank attack to preserve the degree distribution of the graph makes it a high-rank attack.

The rest of this paper is organized as follows. In Section 2 we discuss related work. Section 3 introduces the necessary concepts and notations. We introduce our proposed method in Section 4 and provide experimental results in Section 5. Finally, in section 6 we offer conclusions and discuss future works.

2 RELATED WORK

2.1 Graph Representation Learning Methods

Our work focuses on defending adversarial attacks on graphs and we evaluate the robustness of a node embeddings method against the attacks. In this section, we briefly explain node embeddings for the task of node classification.

Recent years have witnessed an explosion in studying the problem of network representation learning. This interest is stimulated by the "relatively" new advancements in natural language processing (NLP) domain [22, 25, 26]. Specifically, the SkipGram model [25] that has been largely adopted in developing network representation learning techniques. DeepWalk [30], node2vec [16], and Walklets [31] are amongst the methods that employ the SkipGram model for node representation learning after identifying node neighborhoods using the intuition of random walks and they have shown to be very successful for the task of node classification.

A recent study has proposed a tensor-based node embedding method that utilizes tensor decomposition to learn network latent features using the CP decomposition of tensors [2].

2.2 Adversarial Attacks for Graph Data

Research on adversarial attacks in machine learning has received lots of attention in recent years [6, 24]. Adversarial attacks deliberately attempt to attenuate the performance of machine learning algorithms by performing small and unnoticeable changes to the input. Most of the researches on adversarial machine learning are focused on algorithms to fool deep neural networks, mainly for the task of image classification [21, 27, 36].

Recently, a few work have investigated adversarial attacks on graph data [7, 11, 35, 38]. Zügner et al. [38] perform structure and feature perturbations on attributed graphs by an algorithm called Nettack. They generate unnoticeable perturbations by preserving graph's degree distribution and features co-occurrences. The performance of Nettack on attacking GCN shows that it can successfully fool GCN and lead to misclassification of the target node.

In another study, Dai et al. [11] proposed a reinforcement learning based attack that generates structure perturbations with full or limited information about the target classifier. Their approach has shown to be successful for supervised node classification problem. They also claim that adding adversarial examples during training can help to defend the attacks.

The other group of studies, investigate the effects of adversarial attacks on unsupervised node embeddings [7, 35, 38]. In [38], they transfered their attack model to DeepWalk embeddings [30] and observed that the performance of DeepWalk drops on a perturbed graph.

Knowing the fact that graph neural networks and node embeddings are highly vulnerable to adversarial attack, there is an urgent

need to design defense mechanism or effective methods that are more robust against attacks. There are some studies on defense techniques in tasks like image classification [5, 13]. In [13], JPEG compression has been used to "vaccinate" deep neural network. The idea is that adversarial attacks on images add noise to high frequency spectrum so that the noise is visually imperceptible, therefore, JPEG compression can greatly destroy them. In another study, Bhagoji et al. [5], proposed a defense mechanism that utilizes Principal Component Analysis (PCA) for dimensionality reduction.

When it comes to physical adversaries and the goal is the detection of dense block-like behavior, which points to fraud (e.g., fraudulent Twitter followers), [33] leverages Singular Value Decomposition (SVD) and low-rank approximation of adjacency matrix of users in Twitter and Amazon to detect suspicious behavior. They analyze the impacts of evasion attacks in their study, where the malicious data are modified at test time to bypass the result. Another group of attacks are poisoning attacks which perturb the training instances and the classification model is retrained on the perturbed data. In our study, we explore poisoning attacks on graph data and we propose a mechanism to defend the attacks.

3 PRELIMINARIES

In this section, we describe concepts and notations used in the paper.

3.1 Singular Value Decomposition

Singular Value Decomposition (SVD) is one of the most popular matrix decomposition techniques. SVD is a widely used tool to decompose a matrix into sum of rank-1 matrices. Let $A \in \mathbb{R}^{I \times J}$ be a real-valued matrix. The SVD of A is computed as follows:

$$A = U\Sigma V^T \tag{1}$$

where $U \in \mathbb{R}^{I \times I}$ and $V \in \mathbb{R}^{J \times J}$ are orthogonal matrices. Column of U are called the left singular vectors and columns of V are the right singular vector. $\Sigma \in \mathbb{R}^{I \times J}$ is a non-negative diagonal matrix such that $\Sigma_{i,i} = \sigma_i$ where σ_i is the ith singular value and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{min(I,J)}$.

The SVD is an elegant tool to compute the best rank-r approximation of matrix A. The rank-r approximation of A is computed as follows:

$$A_r = U_r \Sigma_r V_r^T = \sum_{i=1}^{r} u_i \sigma_i v_i^T \tag{2}$$

where A_r is the rank-r approximation of A derived from SVD of A. U_r and V_r are the matrices containing the top r singular vectors and Σ_r is the diagonal matrix containing only the r singular values.

According to Eckart-Young-Mirsky theorem [14], A_r is the optimal rank-r approximation of matrix A. For any rank-r matrix B, the following holds:

$$\|A - A_r\|_F \leq \|A - B\|_F \tag{3}$$

3.2 Tensors

A tensor, denoted by $\underline{\mathbf{X}}$, is a multidimensional matrix. The order of a tensor is the number of modes/ways which is the number of indices required to index the tensor [28]. In this paper, we deal with three-mode tensors. Given a three-mode tensor $X \in \mathbb{R}^{I \times J \times K}$ its CP decomposition (also know as CANDECOMP/PARAFAC) [9, 18] is defined as a sum of rank-1 tensors and is formulated as follows:

$$\underline{\mathbf{X}} \approx \sum_{r=1}^{R} a_r \circ b_r \circ c_r \tag{4}$$

where $a_r \in \mathbb{R}^I, b_r \in \mathbb{R}^J, c_r \in \mathbb{R}^K$, and their three-way outer product is computed as $(a_r \circ b_r \circ c_r)(i,j,k) = a_r(i)b_r(j)c_r(k)$. The minimal value of R is called the tensor *rank*. For a more compact representation, CP decomposition is usually represented by the *factor matrices* $A \in \mathbb{R}^{I \times R}$, $B \in \mathbb{R}^{J \times R}$, and $C \in \mathbb{R}^{K \times R}$ where a_r, b_r and c_r are the rth columns of A,B, and C respectively.

$$\underline{\mathbf{X}} = A \circ B \circ C \tag{5}$$

For more details about tensors and tensor decomposition, we refer the interested reader to [20, 28].

4 PROPOSED METHOD

In this section, we present our low-rank matrix approximation method to defend adversarial attacks for graph data. In Section 4.1, we briefly explain the method to generate adversarial attacks for graph data, known as Nettack [38] and we examine the characteristics of the attacks generated by this approach. We show that these attacks impose high-rank changes to the graph which can be greatly ignored by discarding the high-rank components of the graph. In Section 4.2, we present two low-rank methods that are robust to adversarial attacks generated by Nettack. Moreover, in Section 4.3 we propose a low-rank attack and investigate its characteristics compared to Nettack.

4.1 Nettack: a High-Rank Attack

Recently, Zügner et al. [38] introduced an algorithm to generate adversarial attacks for attributed graphs to fool Graph Convolutional Networks. Given an attributed graph $G = (A, X)$, where $A \in \{0,1\}^{N \times N}$ is the undirected adjacency matrix and $X \in \{0,1\}^{N \times D}$ represents the nodes' feature matrix, the goal is to perform small perturbations on the graph $G^{(0)} = (A^{(0)}, X^{(0)})$, so that the result graph $G' = (A', X')$ has lower classification performance and leads to misclassification of the target node. Structure perturbations refer to the changes to the adjacency matrix A, while feature perturbations refer to the changes to the feature matrix X . Nettack produces unnoticeable perturbations by imposing some restrictions to ensure that the attack preserves graph structure and node features. To generate unnoticeable structure perturbations, attacks that preserve the degree distribution of the graph are considered unnoticeable. Whereas, to generate unnoticeable feature perturbations, co-occurrence of the features is taken into consideration. We refer the interested reader to [38] for more details.

Intuition: Nettack perturbations affect small number of nodes. Thus, the footprint of the spectrum of this attack will be comparably smaller than the footprint of the regular structure in the graph, therefore, the attack will likely appear in small singular values, corresponding to higher ranks in the singular value spectrum.

To experimentally validate our intuition and understand the charac-

teristics of the perturbations generated by the NETTACK model, we examined the adjacency and feature matrices before and after the attack. We plotted the singular values of matrices for the clean and perturbed graphs to visualize the differences. Figure 3 illustrates the singular values of the adjacency matrix on semi-logarithmic scale. The singular values shown in Figure 3 correspond to singular values of the adjacency matrix before and after one single attack on the target node. Singular values of the clean and attacked matrices are mainly different at higher ranks. We visualized singular values of adjacency and feature matrices for multiple attacks and observed that singular values are very close at lower ranks but vary at higher ranks.

In Section 4.2, we take advantage of this intuition and present two low-rank solutions that can effectively resist against NETTACK.

4.2 Low-Rank Solutions to Resist Attacks

4.2.1 ***Vaccinating GCN with Low-Rank Approximation.*** To discard the high-rank perturbations generated by NETTACK, we compute the low-rank approximation of the adjacency and feature matrices derived from their SVD decomposition according to Equation 2. We then retrain GCN with the low-rank approximation matrices. With a proper choice of r, the rank-r approximation of the attacked graph can boost the performance of GCN and achieve a performance close to the performance of GCN on the clean graph.

Let A and A' be the adjacency matrices of the clean and attacked graphs respectively. $\delta A = A' - A$ is the difference between the clean and attacked graphs after a series of perturbations. These are the edges added to the clean graph or removed from it as a result of the attack. We compute the SVD of A and δA as follows:

$$A = U\Sigma V^T \tag{6}$$

$$\delta A = U_\delta \Sigma_\delta V_\delta^T \tag{7}$$

Let n be the number of perturbations performed during one attack on the target node v_0. According to [38], $n = d_{v_0} + 2$, where d_{v_0} is the degree of the target node. Leveraging the proof from [34], the leading singular value of δA is computed as follows:

$$\sigma_{\delta 1} = \sqrt{n} = \sqrt{d_{v_0} + 2} \tag{8}$$

In a rank-r approximation of the attacked graph, singular values smaller than σ_r are discarded. Therefore, if $\sigma_{\delta 1}$ is smaller than σ_r,

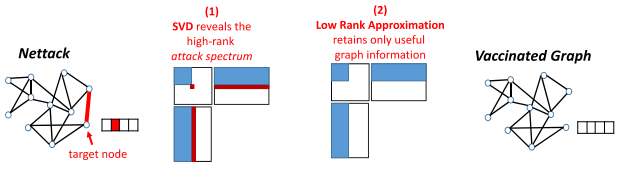


Figure 2: A quick sketch of our proposed vaccination: Taking the SVD of the graph reveals the spectrum of the attack and the healthy parts of the graph. Based on our extensive empirical observations on the high-rankness of NETTACK, we retain a truncated SVD that contains only the top-k singular values for the graph, and reconstruct the graph from them. The output is the vaccinated graph.

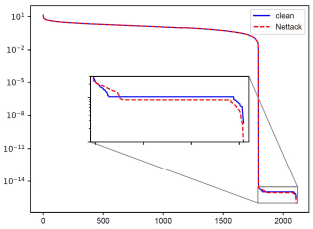


Figure 3: Singular values of adjacency matrix before and after the attack in a semi-logarithmic scale

the perturbations will get eliminated. The goal is to pick rank r so that with a high probability the following holds:

$$\begin{aligned} \sigma_r &> \sigma_{\delta 1} \\ \sigma_r &> \sqrt{d_{v_0} + 2} \end{aligned} \tag{9}$$

$$d_{v_0} < \sigma_r^2 - 2 \tag{10}$$

In other words, a rank-r approximation may not detect attacks on target nodes with degree greater than $\sigma_r^2 - 2$. Formally, we pose the following problem:

Problem: Given an input graph adjacency matrix A with rth largest singular value σ_r, find the value of r so that the probability of the nodes with degree greater than σ_r^2 is less than a given threshold τ:

$$Pr(X \geq \sigma_r^2) < \tau \tag{11}$$

Degree distribution of graphs in real networks has a power-law form. We can write the degree distribution of a graph in form of a discrete power-law with parameter α:

$$\begin{aligned} p(d) = Pr(X = d) &= \frac{d^{-\alpha}}{\sum_{k=d_{min}}^{d_{max}} k^{-\alpha}} \\ &= \frac{d^{-\alpha}}{\sum_{k=0}^{d_{max}-d_{min}} (k + d_{min})^{-\alpha}} \\ &= \frac{d^{-\alpha}}{\zeta(\alpha, d_{min}) - \zeta(\alpha, d_{max} + 1)} \end{aligned} \tag{12}$$

where $\zeta(\alpha, x) = \sum_{k=0}^{\infty}(k + x)^{-\alpha}$ is the Hurwitz zeta function. d_{min} is the minimum degree of a node required to be considered in the power-law distribution and d_{max} is the maximum degree in the graph. Therefore, for a given graph G, we can write its degree distribution as follows:

$$p(d)_{d \in \mathcal{D}_G} \approx \frac{d^{-\alpha}}{\zeta(\alpha, d_{min}) - \zeta(\alpha, d_{max} + 1)} \tag{13}$$

where $\mathcal{D}_G = \{d_v^G | v \in \mathcal{V}, d_v^G \geq d_{min}\}$ is the list of node degrees in the graph G. Clauset et al. [10] drived an approximate expression to estimate the scaling parameter α for a discrete power-law distribution. For graph G:

$$\alpha \approx 1 + |\mathcal{D}_G|. \left[\sum_{d_i \in \mathcal{D}_G} log \frac{d_i}{d_{min} - \frac{1}{2}} \right]^{-1} \tag{14}$$

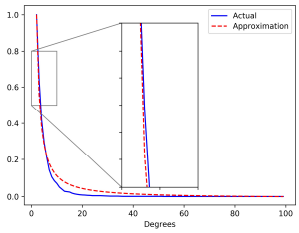


Figure 4: Reverse Cumulative Degree Distribution of CiteSeer

To find the solution for Equation 11, we compute the reverse cumulative probability of power-law as follows:

$$\begin{aligned} Pr(X \geq d) &= \frac{\sum_{d_i=d}^{d_{max}} Pr(X = d_i)}{\zeta(\alpha, d_{min}) - \zeta(\alpha, d_{max}+1)} \\ &= \frac{\sum_{d_i=d}^{d_{max}} d_i^{-\alpha}}{\zeta(\alpha, d_{min}) - \zeta(\alpha, d_{max}+1)} \\ &= \frac{\zeta(\alpha, d) - \zeta(\alpha, d_{max}+1)}{\zeta(\alpha, d_{min}) - \zeta(\alpha, d_{max}+1)} \end{aligned} \tag{15}$$

And for graph G:

$$Pr(X \geq d)_{d \in \mathcal{D}_G} \approx \frac{\zeta(\alpha, d) - \zeta(\alpha, d_{max}+1)}{\zeta(\alpha, d_{min}) - \zeta(\alpha, d_{max}+1)} \tag{16}$$

From equations 11 and 16, we can formulate the problem as follows:

$$Pr(X \geq \sigma_r^2)_{\mathcal{D}_G} \approx \frac{\zeta(\alpha, \sigma_r^2) - \zeta(\alpha, d_{max}+1)}{\zeta(\alpha, d_{min}) - \zeta(\alpha, d_{max}+1)} < \tau \tag{17}$$

Figure 4 shows the actual reverse cumulative degree distribution of the CiteSeer dataset vs. its approximation using Equation 16. The behavior on Cora-ML and PoliticalBlogs datasets is very similar to Figure 4 as well. As illustrated in Figure 4, the reverse cumulative probability quickly drops to zero. On CiteSeer dataset, the probability of degrees greater than 20 is less than 1%. Thus, a low-rank approximation of the graph with high probability can eliminate the perturbations.

In the experimental evaluations that follows, we evaluate Equation 11 for different values of rank r on the real-world datasets and experimentally show that how big r needs to be to successfully vaccinate GCN against adversarial perturbations.

4.2.2 *Robust Tensor-Based Node Embeddings*. t-PINE

Recently, Al-Sayouri et al. [2] proposed a tensor-based node embedding method that utilizes tensor's CP decomposition to capture the relations between nodes using low-dimensional latent components. They examined the performance of t-PINE in the context of node embeddings, however the robustness of t-PINE has not been evaluated in an adversarial context. Due to the inherent low-rank nature of t-PINE, it is a good candidate to defend high-rank perturbations generated by Nettack. Here, we briefly explain t-PINE:

t-PINE jointly encodes explicit[1] and implicit[2] network structure [2] using CP decomposition [9], which greatly allows for a systematic exploration of higher-order proximities. Due to the use of multi-aspect data, t-PINE forms a three-mode tensor to represent a network which has two slices: (1) The adjacency matrix, and (2) K-nearest neighbor matrix computed for the feature matrix. Then, tensor $\underline{\mathbf{x}} \in \mathbb{R}^{N \times N \times 2}$ is decomposed using CP decomposition as given in Equation 4. The CP model is solved using the Alternating Least Squares (ALS) algorithm [9]. For a predefined parameter d which is the embedding dimension, tensor $\underline{\mathbf{X}}$ is decomposed as:

$$\mathcal{L} \approx \min \|\underline{\mathbf{X}} - A \circ B \circ C\|_F^2 \tag{18}$$

where $A \in \mathbb{R}^{N \times d}$, $B \in \mathbb{R}^{N \times d}$, and $C \in \mathbb{R}^{2 \times d}$ are the factor matrices. When CP decomposition is used in multi-label classification problem, the *tensor rank* R denotes the number of classes, however, in t-PINE, $R = d$, indicates the embedding dimensionality, as the CP decomposition is tailored for representation learning purpose.

In contrast to state-of-the-art approaches [16, 17, 30, 31, 37], t-PINE yields highly predictable representations on different multi-label classification problems. Further, it generates nicely interpretable embeddings, where we can understand how each view contributes to the learned representation vectors. For more details, we refer the reader to [2].

4.3 *LowBlow*: a Low-Rank Attack

As explained in Section 4.1, Nettack perturbations cause changes to high-rank singular values which could be defended using a low-rank approximation of the graph. Now a question that might arise is that what happens if we have a low-rank attack. Is a low-rank perturbation able to successfully attack the graph and fool GCN and if so, are we able to defend it using our proposed low-rank mechanism? Is the low-rank attack still unnoticeable? We will answer these questions in the experimental evaluations. In this section, we will manipulate the Nettack perturbations so that it results to low-rank attacks that can hinder the performance of both GCN and t-PINE.

To generate a low-rank attack, we replace some of the most significant singular values and vectors of δA with the corresponding singular values and vectors of A. In other words, singular values of δA within range $[i, j]$ $(\sigma_{\delta i}, ..., \sigma_{\delta j})$ are replaced by singular values of A $(\sigma_i, ..., \sigma_j)$. i and j are relatively small values so that the corresponding singular values are significant (e.g. less than 100). Then we reconstruct the low-rank δA from altered singular values and vectors. Adding this low-rank δA to the clean adjacency matrix A will result to a low-rank attack that is able to perturb GCN and t-PINE.

$$A'_{low-rank} = A + \delta A_{low-rank} \tag{19}$$

We compute the low-rank perturbations on the feature matrix in an analogous way. Let F and F' be the feature matrices of the clean and attacked graphs, respectively. $\delta F = F' - F$ is the matrix representing features added to/removed from the original feature matrix F. After computing the SVD of F and δF, we replace some of the most significant singular values and vectors of δF with the

[1] Refers to network first-order proximity connections
[2] Refers to second- or higher-order proximity connections

corresponding singular values and vectors of F. $\delta F_{low-rank}$ is reconstructed using these modified singular values and vectors. To get the low-rank feature perturbations, we add $\delta F_{low-rank}$ to F.

$$F'_{low-rank} = F + \delta F_{low-rank} \tag{20}$$

In the experimental evaluation that follows, we verify the effectiveness of *LowBlow*, and we also evaluate the extent to which *LowBlow* alters the perception of the graph to an observer, in the form of the node degree distributions.

5 EXPERIMENTS

5.1 Datasets and Experiment Setup

Datasets: In order to compare our results to the adversarial attack paper [38], we use the same datasets in our experiments. The datasets are CiteSeer [15], Cora-ML[8], and PoliticalBlogs [1]. Cora-ML is the subset of machine learning papers from the well-know Cora dataset [23]. Table 1 provides the statistics for each dataset. All experiments are performed on the largest connected component (LCC) of the graphs.

Dataset	|V|	|E|	$\|V_{LCC}\|$	$\|E_{LCC}\|$	Classes
CiteSeer [15]	3312	4715	2110	3757	6
Cora-ML[8]	2995	8416	2810	7981	7
PoliticalBlogs[1]	1490	19025	1222	16714	2

Table 1: Datasets descriptions

Setup: In section 4.1, we showed that Nettack perturbations are of high-rank. To further investigate our intuition, we follow the same procedure as described in the Nettack paper [38]. We split the network in labeled (20%) and unlabeled nodes(80%). half of the labeled data is used for training and the other half is used for validation in the process of training the GCN model. We perform five iterations where at each iteration a different random splits of data is generated. We first train the GCN surrogate model on the labeled data and then we select 40 target nodes from test set with the following conditions:

- 20 nodes which are correctly classified: 10 of them have the highest classification margins and 10 of them have the lowest margin.
- 20 random nodes

According to the algorithm proposed in [38], there are two different ways to attack a target node : direct attack called Nettack, and influence attack called Nettack-In which attacks a node indirectly. In our experiments we only consider attacking each target node directly, as it is a stronger attack compared to an indirect attack. We also combine structure and feature perturbations which leads to a greater performance loss. To evaluate the effectiveness of the attack, we compute $X = Z^*_{v_0,c_{old}} - max_{c \neq c_{old}} Z^*_{v_0,c}$ where Z is the class probabilities and c_{old} is the ground truth label of the target node. X is called the classification margin. A successful attack leads to lower values of X and a negative value means the target node has been successfully misclassified.

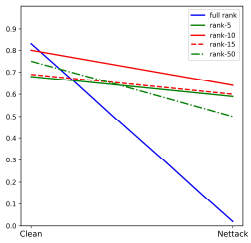


Figure 5: Fraction of target nodes correctly classified after vaccinating GCN on CiteSeer

Method	CiteSeer	Cora-ML	PoliticalBlogs
GCN - Clean	0.83	0.82	0.90
GCN - Nettack	0.02	0.01	0.09
Vaccinated - Clean	0.80	0.76	0.84
Vaccinated - Nettack	0.64	0.59	0.62

Table 2: Vaccinating GCN against Nettack. Fraction of target nodes that are correctly classified is reported.

5.2 Vaccinating GCN with Low-Rank Approximation

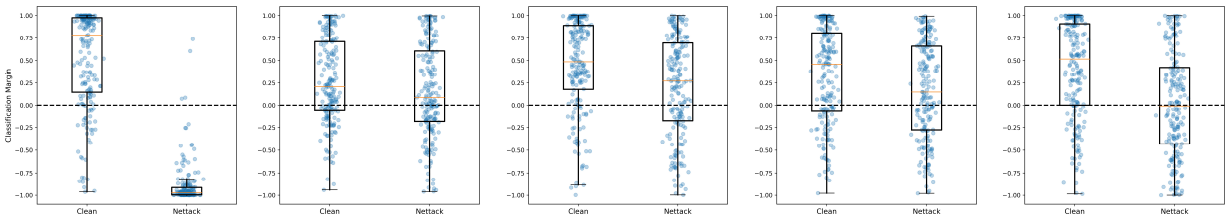
In Section 4.2.1, we presented the low-rank approximation to defend Nettack perturbations. In this section, we analyze the performance of our defense mechanism. To this end, we examine different values of rank $r = 5, 10, 15$, and 50 to compute the approximations. Figure 5 shows that the fraction of target nodes correctly classified after the attack drops significantly with the full-rank attacked matrices. However, using the low-rank SVD approximation, this number is close to the fraction of correctly classified nodes on the clean graph. Figure 6 and 5 illustrate that using rank-10 approximation of the adjacency and feature matrices we are able to significantly alleviate the effects of Nettack. Only 10 singular values/vectors is sufficient to have a robust approximation of the graph structure and features and vaccinate GCN against attacks. As we discussed in Section 4.2.1 and Equation 11, if $Pr(X \geq \sigma^2_{10})$ is less than a threshold τ, with a high probability we discard perturbations. $Pr(X \geq \sigma^2_{10})$ is 0.0013, 0.0019, and 0.0037 on CiteSeer, Cora-ML, and PoliticalBlogs, respectively. These probabilities are nearly zero which shows that a rank-10 approximation of the graph ignores perturbations with a very high probability.

We performed this experiment on all three datasets and observed that $r = 10$ produces the best results. Table 2 shows the results. Here, for brevity, we only report the results for $r = 10$.

5.3 Transferring Adversarial Attacks to t-PINE Embeddings

To evaluate the transferability of adversarial attacks to the tensor-based embeddings, we pursue the experiment as explained in Section 5.1 . After every attack, we compute the t-PINE embeddings where the tensor slices are the attacked adjacency and feature matrices.

We perform the experiment for different values of embedding dimensions d and K to examine their effects on the robustness of t-PINE. Figure 7 shows the fraction of target nodes that their prediction changed after the attack. This does not necessarily mean that these nodes were correctly classified before the attack, but it shows that the Nettack perturbations were able to change the prediction of the target nodes from one class to another. Nettack



(a) Unvaccinated (b) rank-5 approx. (c) rank-10 approx. (d) rank-15 approx. (e) rank-50 approx.

Figure 6: Vaccinating GCN against NETTACK

might have even changed a node's prediction from a wrong class to the correct one. The plot shows that lower dimensions of t-PINE are more robust against NETTACK. As the embedding dimension gets larger, more nodes are affected by the attacks. t-PINE performs very robust against the attacks. Even at dimension 512, less than 40% of target nodes are affected by the attacks. At lower dimensions, CP-decomposition can greatly discard affected components of graph by the attack.

On the other hand, choice of K does not have a significant effect on the robustness of t-PINE. We evaluated different values of K and observed that at a fixed dimension, performance slightly improves for bigger values of K. However, the larger K is, t-PINE's runtime increases. The improvement over larger values of K is negligible. Therefore, for the rest of the experiments, we only report the t-PINE results for $d = 32$ and $K = 30$ that leads to robust results in a better runtime.

Figure 8 shows the result of transferring NETTACK perturbations to t-PINE for $d = 32$ and $K = 30$. The plot shows that t-PINE is very robust to the attacks and the classification margins before and after the attack has remained nearly unchanged.

In Table 3, we summarize the results of transferring NETTACK perturbation to t-PINE for different datasets. The values reported in the table are the fraction of target nodes that get correctly classified. For t-PINE the values on clean and perturbed graphs are very close for CiteSeer and Cora-ML datasets. However, on PoliticalBlogs, the performance of t-PINE has dropped with NETTACK perturbations. The degree of target nodes in PoliticalBlogs dataset are relatively larger compared to the other datasets. In our experiments, we set the number of perturbations to a target node relevant to its degree. Therefore, in the PoliticalBlogs dataset, we perform a larger number of perturbation and this could be the reason to why the performance of t-PINE drops when facing NETTACK perturbations.

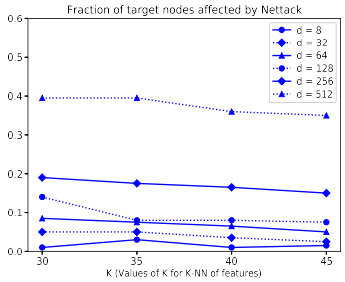


Figure 7: Robustness of t-PINE against NETTACK for different embedding dimensions and K on CiteSeer

	Method	CiteSeer	Cora-ML	PoliticalBlogs
GCN	Clean	0.83	0.82	0.90
	NETTACK	0.02	0.01	0.06
t-PINE	Clean	0.74	0.68	0.87
	NETTACK	0.72	0.64	0.30

Table 3: Transferring NETTACK to t-PINE embeddings. For t-PINE, fraction of target nodes correctly classified after the attack is very close to values on the clean graph.

5.4 *LowBlow*: A Low-Rank Attack

Here, we investigate the influence of the proposed low-rank attack, *LowBlow* on GCN and t-PINE. To evaluate the effects of *LowBlow*, we compute the perturbed adjacency and feature matrices as in Equations 19 and 20. Then we retrain GCN model with the perturbed matrices. We also compute the t-PINE embeddings for the perturbed matrices. *LowBlow* significantly decreases the performance of GCN. It is also able to attack t-PINE, however, it is less successful compared to perturbing GCN.

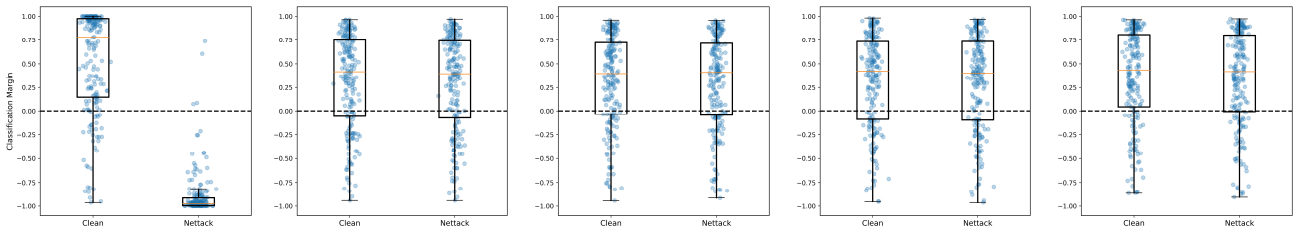
In addition, we examined our defense mechanism against *LowBlow*. We used a rank-10 approximation of graph to vaccinate it. In Table 4, we summarize the results for all datasets. Vaccinating GCN has improved its performance but it decreased the performance of t-PINE on CiteSeer and Cora. We observed that for a smaller embedding dimension e.g $d = 8$, vaccinating t-PINE against *LowBlow* has no significant impact on the performance of t-PINE.

Due to the low-rank nature of *LowBlow*, it is more difficult to defend compared to NETTACK, and our vaccination method performs better on NETTACK rather than *LowBlow*.

5.5 Degree Distributions After *LowBlow*

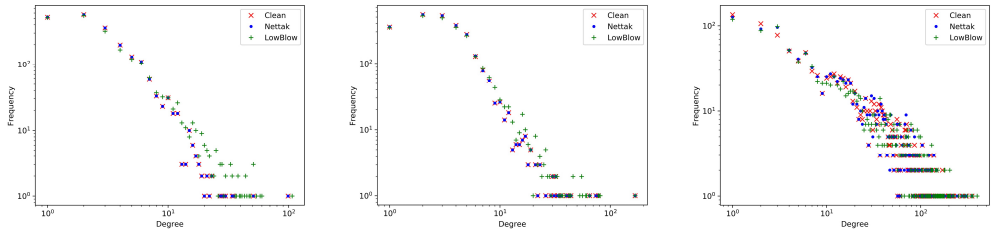
In the previous subsection we demonstrated the effectiveness of *LowBlow* in fooling our proposed low-rank vaccination scheme, and deteriorating the performance of both GCN and t-PINE. In addition to the effectiveness of the attack, another important aspect that we would like to study experimentally is the effect of *LowBlow* in "what the graph looks like". In computer vision attacks, "look" can be easily defined by how a human perceives the poisoned data point/image. In graphs, however, such an intuitive metric does not exist.

Instead, [38] studies a proxy, which is the node degree distribution and how it is affected by the attack. In [38], the attack only



(a) GCN (b) t-PINE ($K = 30$) (c) t-PINE ($K = 35$) (d) t-PINE ($K = 40$) (e) t-PINE ($K = 45$)

Figure 8: Poisoning of t-PINE with NETTACK on CiteSeer. The embedding dimension is 32.



(a) CiteSeer (b) Cora-ML (c) PoliticalBlogs

Figure 9: Degree distributions of the clean and attacked graphs on log-log scale. *LowBlow* affects the degree distribution only for the high-degree nodes, while leaving the majority of the nodes intact.

affects one or a few nodes at a time, and thus, the attack results in a statistically insignificant alteration of the degree distribution. *LowBlow*, on the other hand, by virtue of mixing the attack in high-valued singular components of the graph, this mixing may affect a number of nodes, resulting in statistically significant differences in the distributions, *for some nodes*.

In Figure 9 we plot the degree distributions of the three real-world graphs we use, before and after the attack, in log-log scale. What we uniformly observe is that only the very high-degree nodes are affected by the attack, while the low and mid-degree nodes, which constitute the vast majority of this heavy-tailed distribution, remain intact, as far as their degree distribution is concerned.

To evaluate whether *LowBlow* perturbations are unnoticeable, we perform a statistical two-sample test for power-law distribution [3, 38] to see if the adjacency matrix after *LowBlow* perturbations follows similar degree distribution as the input graph. The null hypothesis H_0 proposes that the two samples have similar power-law distributions. Here, we compute the probability of not rejecting the null hypothesis where the two samples are from different distributions (*Type II error*). Similar to [38], we set the p-value to 0.95 which is a very conservative threshold and two samples from the same distribution are rejected 95% of the time. Following this conservative test, degree sequence of the graph after *LowBlow* perturbations does not follow the same power-law distribution as the input graph, i.e. the proposed low-rank attack is noticeable. To make the attack unnoticeable, we only consider edges that if added or removed from the graph, degree distribution will not change. After this step, we plotted the singular values of the graph before and after the attack and observed that the singular values are mainly different in higher ranks and the behavior is similar to NETTACK. This implies that an unnoticeable perturbation affects high frequency spectrum of the graph. Consequently, our proposed vaccination mechanism successfully defends against unnoticeable adversarial attacks.

	Method	CiteSeer	Cora-ML	PoliticalBlogs
GCN	Clean	0.83	0.82	0.90
	NETTACK	0.02	0.01	0.06
	LowBlow	0.05	0.06	0.06
	Vaccinated NETTACK	0.64	0.59	0.62
	Vaccinated *LowBlow*	0.31	0.35	0.38
t-PINE	Clean	0.74	0.68	0.87
	NETTACK	0.72	0.64	0.30
	LowBlow	0.55	0.48	0.33
	Vaccinated NETTACK	0.73	0.65	0.52
	Vaccinated *LowBlow*	0.29	0.27	0.48

Table 4: Results overview. Comparison of poisoning and vaccination of GCN and t-PINE against NETTACK and *LowBlow*

6 CONCLUSIONS

In this paper, we examined the characteristics of NETTACK perturbations for graphs. Due to the vulnerability of the node classification approaches to the adversarial attacks, we highlighted the need for a defense system or robust node classification methods. We illustrated that NETTACK generates high-rank perturbations that can be discarded using a low-rank approximation of the adjacency and feature matrices. We showed that a rank-10 approximation of the matrices is able to defend adversarial attacks with a high probability and achieve a performance close to the performance on the clean graph. Furthermore, we examined the robustness of t-PINE, a tensor-based node embedding against NETTACK and we observed that it is very robust for lower embedding dimensions and the robustness of the embedding decreases as the dimension gets bigger. In addition, we proposed an algorithm to generate low-rank

adversarial attacks that could fool both GCN and tensor-based embeddings. *LowBlow* perturbations are noticeable and the attacked graph does not have the same degree distribution as the input graph. We also showed that modifying *LowBlow* to only keep the edges that preserve the degree distribution makes it a high-rank attack similar to NETTACK. In conclusion, unnoticeable adversarial attacks on graphs impose high-rank changes in singular values of the input graph which can be greatly eliminated with our proposed low-rank defense mechanism.

7 ACKNOWLEDGEMENTS

Research was supported by the National Science Foundation CDS&E Grant no. OAC-1808591. GPUs used for this research were donated by the NVIDIA Corporation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding parties.

REFERENCES

[1] Lada A Adamic and Natalie Glance. 2005. The political blogosphere and the 2004 US election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*. ACM, 36–43.

[2] Saba A Al-Sayouri, Ekta Gujral, Danai Koutra, Evangelos E Papalexakis, and Sarah S Lam. 2018. t-PINE: Tensor-based Predictable and Interpretable Node Embeddings. *arXiv preprint arXiv:1805.01889* (2018).

[3] Alessandro Bessi. 2015. Two samples test for discrete power-law distributions. *arXiv preprint arXiv:1503.00643* (2015).

[4] Smriti Bhagat, Graham Cormode, and S Muthukrishnan. 2011. Node classification in social networks. In *Social network data analytics*. Springer, 115–148.

[5] Arjun Nitin Bhagoji, Daniel Cullina, and Prateek Mittal. 2017. Dimensionality reduction as a defense against evasion attacks on machine learning classifiers. *arXiv preprint* (2017).

[6] Battista Biggio, Blaine Nelson, and Pavel Laskov. 2012. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389* (2012).

[7] Aleksandar Bojcheski and Stephan Günnemann. 2018. Adversarial attacks on node embeddings. *arXiv preprint arXiv:1809.01093* (2018).

[8] Aleksandar Bojchevski and Stephan Günnemann. 2018. Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking. (2018).

[9] J Douglas Carroll and Jih-Jie Chang. 1970. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Youn" decomposition. *Psychometrika* 35, 3 (1970), 283–319.

[10] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. 2009. Power-law distributions in empirical data. *SIAM review* 51, 4 (2009), 661–703.

[11] Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. 2018. Adversarial Attack on Graph Structured Data. *arXiv preprint arXiv:1806.02371* (2018).

[12] Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, Deepak Verma, et al. 2004. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 99–108.

[13] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E Kounavis, and Duen Horng Chau. 2018. Shield: Fast, Practical Defense and Vaccination for Deep Learning using JPEG Compression. *arXiv preprint arXiv:1802.06816* (2018).

[14] C. Eckart and G. Young. 1936. The approximation of one matrix by another of lower rank. *Psychometrika* 1, 3 (1936), 211–218.

[15] C Lee Giles, Kurt D Bollacker, and Steve Lawrence. 1998. CiteSeer: An automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries*. ACM, 89–98.

[16] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 855–864.

[17] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*. 1024–1034.

[18] R.A. Harshman. 1970. Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multimodal factor analysis. (1970).

[19] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. *5th International Conference on Learning Representations (ICLR-17)* (2017).

[20] T.G. Kolda and B.W. Bader. 2009. Tensor decompositions and applications. *SIAM review* 51, 3 (2009).

[21] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533* (2016).

[22] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. 1188–1196.

[23] Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval* 3, 2 (2000), 127–163.

[24] Shike Mei and Xiaojin Zhu. 2015. Using Machine Teaching to Identify Optimal Training-Set Attacks on Machine Learners.. In *AAAI*. 2871–2877.

[25] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[26] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[27] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2017. Universal adversarial perturbations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Ieee, 86–94.

[28] Evangelos E Papalexakis, Christos Faloutsos, and Nicholas D Sidiropoulos. 2017. Tensors for data mining and data fusion: Models, applications, and scalable algorithms. *ACM Transactions on Intelligent Systems and Technology (TIST)* 8, 2 (2017), 16.

[29] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 582–597.

[30] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 701–710.

[31] Bryan Perozzi, Vivek Kulkarni, and Steven Skiena. 2016. Walklets: Multi-scale graph embeddings for interpretable network classification. *arXiv preprint arXiv:1605.02115* (2016).

[32] Trang Pham, Truyen Tran, Dinh Q Phung, and Svetha Venkatesh. 2017. Column Networks for Collective Classification.. In *AAAI*. 2485–2491.

[33] Neil Shah, Alex Beutel, Brian Gallagher, and Christos Faloutsos. 2014. Spotting suspicious link behavior with fbox: An adversarial perspective. In *Data Mining (ICDM), 2014 IEEE International Conference on*. IEEE, 959–964.

[34] Neil Shah, Alex Beutel, Brian Gallagher, and Christos Faloutsos. 2014. Spotting Suspicious Link Behavior with fBox: An Adversarial Perspective. *arXiv preprint arXiv:1410.3915* (2014).

[35] Mingjie Sun, Jian Tang, Huichen Li, Bo Li, Chaowei Xiao, Yao Chen, and Dawn Song. 2018. Data Poisoning Attack against Unsupervised Node Embedding Methods. *arXiv preprint arXiv:1810.12881* (2018).

[36] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).

[37] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. 2015. Network Representation Learning with Rich Text Information.. In *IJCAI*. 2111–2117.

[38] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. 2018. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2847–2856.