COMPARING AND DETECTING ADVERSARIAL ATTACKS FOR GRAPH DEEP LEARNING

Yingxue Zhang*& Sakif Hossain Khan *
Huawei Noah's Ark Laboratory
Montréal Research Centre
7101 Avenue du Parc, H3N 1X9
Montréal, QC Canada
{yingxue.zhang, sakif.hossain.khan}@huawei.com

Mark Coates
Department of Electrical and Computer Engineering
McGill University
3480 University Street, H3A 0E9
Montréal, QC Canada
mark.coates@mcgill.ca

ABSTRACT

Deep learning models have achieved state-of-the-art performance in classifying nodes in graph-structured data. However, recent work has shown that these models are vulnerable to adversarial attacks. In particular, is is possible to adversarially perturb the graph structure and the node features in order to induce classification errors. In this paper, we study the effect of recently proposed attacks on graph models which incorporate structure exploration. We then propose a method for detecting attacks when they occur.

1 INTRODUCTION

Deep learning models have achieved impressive performance on a wide variety of machine learning tasks. Driven by this success, such models are increasingly being deployed in real-world systems. Concurrently, however, it has been discovered that these models are vulnerable to malicious attacks with grave implications for safety and robustness. Example domains include autonomous vehicles, speech recognition and malware classification (Carlini & Wagner (2017)). Recent developments in the study of graph neural networks (Defferrard et al. (2016); Kipf & Welling (2017); Hamilton et al. (2017); Monti et al. (2017); Gilmer et al. (2017); Veličković et al. (2018); Battaglia et al. (2018)) have led to industrial applications of these models (Ying et al. (2018); Geng et al. (2019); Liu et al. (2019)). There has thus been a concomitant interest in vulnerabilities of graph deep learning.

In this vein, Zügner et al. (2018) proposed a procedure for generating adversarial perturbations against graph data, called Nettack. In particular, the main Nettack algorithm perturbs the graph topology and/or the node attributes so that there is significant degradation in node classification performance. This motivates the need to explore models and methods which are robust to such attacks while also necessitating the construction of detectors for such attacks. The first task leads us to study the effects of random attacks and Nettack on recent graph deep learning models which apply structure exploration as part of inference (for instance, Veličković et al. (2018); Zhang et al. (2019)). The second task drives us to study the statistical differences between unperturbed graphs and perturbed graphs. We note that perturbations to topology are more interesting than perturbations to features since the predictive power of graph deep learning models is due to the relational data inherent in graphs. Hence, unless specified otherwise, it will be assumed that there are no feature perturbations.

*The two first authors made equal contributions

2 ATTACKS ON MODELS WITH STRUCTURE EXPLORATION

We briefly describe the structure perturbations studied in this work. First, a random attack targets one specific node v_i in the graph with a perturbation budget Δ. The attack first removes $\Delta/2$ randomly chosen edges of v_i. Next, it attaches $\Delta/2$ edges to nodes which are differently labeled compared to v_i. For our random attacks we set $\Delta = d_{v_i} + 2$, where d_{v_i} is the degree of v_i.

Nettack also targets a single node with a perturbation budget Δ. However, it provides a more systematic way of changing the local topology around v_i. Briefly, Nettack formulates a (direct) attack as a two-level optimization problem where the outer optimization decreases the classification margin for the correct class as much as possible. The inner optimization trains a surrogate model on the (possibly) perturbed data. The solution to the two-level optimization problem is a perturbed adjacency matrix which represents the best possible perturbation to v_i. However, the outer optimization must be constrained so as to produce sensible topological perturbations. In particular, Zügner et al. (2018) define "unnoticeability" constraints which enforce minimal changes to the overall distribution of node degrees in the graph.

Observe that the choice of surrogate model and perturbation constraint determines both the efficacy of the attack as well as the hardness of the two-level optimization problem. The surrogate model for Nettack is a simplified GCN (Kipf & Welling (2017)) and this raises the question of whether or not Nettack transfers to models which differ substantially from a basic GCN architecture. Zügner et al. (2018) provide some evidence that this is the case. However, our experiments with the GAT (Veličković et al. (2018)) and the BGCN (Zhang et al. (2019)) indicate that Nettack has weaker transferability to these models (see Section 4). Additionally, the BGCN is significantly more robust than either GAT or GCN under random attacks. The GAT and BGCN share a commonality in that both of these models incorporate structure exploration. GAT leverages masked self-attention layers in order to reduce the impact of edges which do not represent meaningful relationships for node classification. This represents an implicit form of structure exploration. The BGCN, on the other hand, incorporates this sort of exploration explicitly. This is achieved by treating the observed graph as a sample from a parameterized collection of random graphs (Zhang et al. (2019) employ the stochastic blockmodel). Whereas the GAT is limited to processing only existing edges, the BGCN is able to handle uncertainty regarding the graph structure as part of the training process. These observations lead us to hypothesize that incorporating structure exploration not only allows for competitive performance on graph-structured data but also that it enhances robustness of models against malicious attacks.

3 DETECTION

As we have seen above, different models exhibit different degrees of susceptibility to adversarial attacks on topology. However, it is generally true that performance degrades appreciably against such attacks. Given the potential utility of these models to real applications, it is imperative that we have methods for detecting attacks when they do occur. Since Nettack is a more focused and therefore more effective attack, we study the problem of detecting nodes which have been subject to topological perturbations calculated by Nettack. Moreover, because the GCN is a very popular model and is also the one most vulnerable to Nettack Zügner et al. (2018), we restrict ourselves to detecting attacks against GCN only. We study detection for other models in future work.

Recall that Nettack uses a simplified GCN as a surrogate model for generating perturbations. The predictive power of GCN and other node embedding methods stems from the fact that these algorithms are able to capture similarity between adjacent nodes (Perozzi et al. (2014); Tang et al. (2015)). In the node embedding literature (Tang et al. (2015)), first-order proximity is characterized as the similarity of node embeddings for two nodes which are directly connected by an edge. In addition, second-order proximity is defined as the similarity of the node embedding for two nodes which share a neighbour.

Intuitively, prediction logits for classification play a role similar to node embeddings. Given that Nettack exploits the GCN logits of v_i, we expect that Nettack creates a discrepancy between the first-order proximity information of v_i and that of the neighbours of v_i. We propose measuring this discrepancy by calculating the mean of the KL divergences between the softmax probabilities of v_i

and those of its neighbours. Mathematically, we compute

$$\text{prox}_1(i) = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} D_{\text{KL}}(p_i \| p_j)$$

where p_i indicates the GCN output softmax probabilities for the i^{th} node. Furthermore, second-order proximity information is calculated[1] using the KL divergence between the softmax probabilities of pairs of neighbours

$$\text{prox}_2(i) = \frac{1}{|\mathcal{N}(i)|\,(|\mathcal{N}(i)| - 1)} \sum_{j \in \mathcal{N}(i)} \sum_{k \in \mathcal{N}(i)} D_{\text{KL}}(p_j \| p_k)$$

From Figure 1, we observe a clear difference between the prox_1 and prox_2 statistics for unperturbed versus perturbed nodes. This motivates us to define straightforward detection tests by setting thresholds τ_1 and τ_2 for prox_1 and prox_2 respectively. That is, given a possibly perturbed node, we flag the node as being perturbed if either prox_1 exceeds τ_1 or prox_2 exceeds τ_2. We describe additional quantities of interest along with corresponding figures and statistics in the appendix.

We apply the Neyman-Pearson lemma (Neyman & Pearson (1933)) to set the detection threshold for each statistic. The null hypothesis distributions are modeled as normal distributions. We fit a Gaussian distribution to unperturbed training data via a maximum likelihood approach. We determine an appropriate detection threshold τ by matching the tail probability to a specific target false positive rate. The threshold value calculation is done using the inverse cumulative distribution function (cdf).

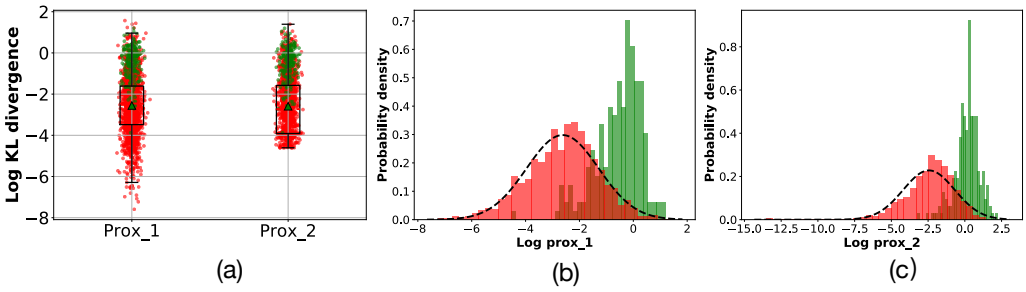


Figure 1: Box plots of (a) prox1 and prox2 statistics after log transform for unperturbed nodes and perturbed nodes. (c) and (d), using Gaussian distributions to fit the log-transformed statistics from the unperturbed nodes (Citeseer dataset).

4 Experiments

Setup: For all our experiments, we used the Cora, Citeseer and Polblogs datasets exactly as provided by the authors of Zügner et al. (2018)[2]. Whenever Nettack was used, we used (a slightly refactored version of) the authors' source code for generating node perturbations. The GCN, GAT and BGCN architectures and hyperparameters are identical to those of Kipf & Welling (2017), Veličković et al. (2018) and Zhang et al. (2019) respectively. For both random attack and Nettack, 40 target nodes are selected exactly as described in Zhang et al. (2019). We perform comparison of accuracy and classifier margins before and after attacks in accordance with the evaluation procedure of Zhang et al. (2019).

Results: Figure 2 and Table 1 show the effects of both random attack and Nettack on different models. We see that GCN and GAT are quite vulnerable to random attacks. Nettack has a more severe impact on all classification algorithms. However, in both cases, GAT is more robust compared to GCN with the larger robustness gap occurring for Nettack. The BGCN is extremely robust to random attacks but is more vulnerable than GAT to Nettack. BGCN is still not as severely impacted as GCN under Nettack. These results suggest that the Nettack procedure has a limited ability to transfer across models.

[1] See Appendix A for how this is carried out in practice.

[2] Source code and datasets are provided at `https://www.kdd.in.tum.de/research/nettack/`

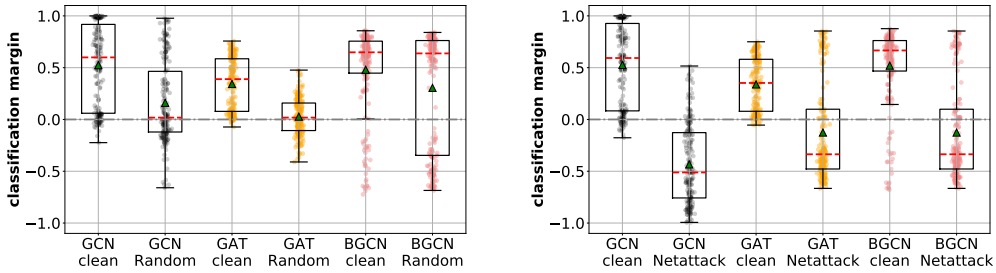


Figure 2: Comparison of random attack (left) and Nettack (right) on GCN, GAT and BGCN on Citeseer dataset

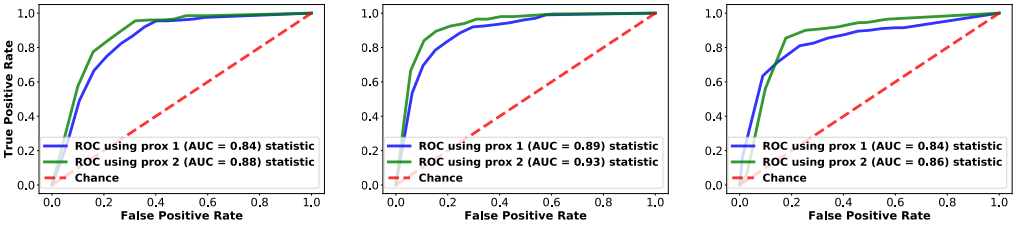
	No attack	Random attack	Nettack
	Accuracy		
GCN	87.0%	52.5%	16.0%
BGCN	84.5%	56.5%	27.5%
GAT	84.5%	41.5%	45.0%
	Classification margin		
GCN	0.506	0.153	-0.459
BGCN	0.521	0.187	-0.149
GAT	0.413	0.016	0.011

(a) Performance for the Cora dataset

	No attack	Random attack	Nettack
	Accuracy		
GCN	87.5 %	50.0%	15.5%
BGCN	88.5 %	67.5%	31.0%
GAT	96.0%	53.0%	39.0%
	Classification margin		
GCN	0.506	0.163	-0.459
BGCN	0.521	0.318	-0.149
GAT	0.342	0.027	0.011

(b) Performance for the Citeseer dataset

Table 1: Comparison of accuracy and classification margins for the no attack, random attack and the Nettack scenarios on the Cora (left) and Citeseer (right) datasets. The results are for 40 selected target nodes and 5 runs of the algorithms for each target. (Citeseer dataset)



(a) Cora (b) Citeseer (c) Polblogs

Figure 3: ROC curve for Cora, Citeseer and Polblogs using the proposed detection approach

Next, we analyze how well our proposed detection tests perform in attempting to flag nodes perturbed by Nettack. In Figure 3, we demonstrate the effectiveness of our proposed detection techniques. A high area-under-the-curve (AUC) score is obtained for all three datasets. While maintaining a false alarm rate below 15%, the proposed perturbed node detection method can achieve 79.5%, 88.5%, 77.5% recall rate on the perturbed nodes. In our experiments, detection using the second-order proximity statistic performs better than detection using the first-order proximity statistic.

5 Conclusions

We have studied the effects of random attacks and the recently proposed Nettack (Zügner et al. (2018)) on GCN, GAT and BGCN. We find that the BGCN is most robust to random attacks and GAT and (to a lesser extent) BGCN are less vulnerable to Nettack. This suggests that graph deep learning models which use structure exploration are more robust in general and that Nettack, possibly due to a restricted surrogate model, lacks transferability. We also found that Nettack perturbations on

GCN can be detected reasonably reliably with a relatively simple threshold test. We suspect this is due to the fact that the unnoticeability constraints of Nettack enforce unnoticeability at the global level but fail to do so at the local level. In future work, we aim to propose unnoticeability metrics which better capture local graph structure in addition to global structure. On a related note, it would be interesting to see if our detection test is successful for other kinds of attacks (Zügner et al. (2019); Dai et al. (2018)). We plan to implement better adversarial attacks against graph learning models which incorporate these new metrics as constraints. Finally, we note that the evaluation of attack effectiveness by Zügner et al. (2018) measures performance degradation locally but computes unnoticeability globally. We aim to resolve this tension going forward.

REFERENCES

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv:1806.01261*, 2018.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Proc. IEEE Symp. Security and Privacy*, 2017.

Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. Adversarial attack on graph structured data. In *Proc. Int. Conf. Machine Learning*, 2018.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proc. Adv. Neural Inf. Proc. Systems*, 2016.

Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *Proc. AAAI Conf. Artificial Intelligence*, 2019.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proc. Int. Conf. Machine Learning*, 2017.

William Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proc. Adv. Neural Inf. Proc. Systems*, 2017.

Thomas Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proc. Int. Conf. Learning Representations*, 2017.

Ziqi Liu, Chaochao Chen, Longfei Li, Jun Zhou, Xiaolong Li, Le Song, and Yuan Qi. Geniepath: Graph neural networks with adaptive receptive paths. In *Proc. AAAI Conf. Artificial Intelligence*, 2019.

Federico Monti, Davide Boscaini, et al. Geometric deep learning on graphs and manifolds using mixture model CNNs. In *Proc. IEEE Conf. Comp. Vision and Pattern Recognition*, Jul. 2017.

Jerzy Neyman and Egon Sharpe Pearson. IX. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, 1933.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proc. ACM Int. Conf. Knowl. Disc. Data Mining*, 2014.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proc. Int. Conf. World Wide Web*, 2015.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *Proc. Int. Conf. Learning Representations*, Apr. 2018.

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2018.

Yingxue Zhang, Soumyasundar Pal, Coates Mark, and Deniz Üstebay. Bayesian graph convolutional neural networks for semi-supervised classification. In *Proc. AAAI Conf. Artificial Intelligence*, 2019.

Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. In *Proc. ACM Int. Conf. Knowl. Disc. Data Mining*, 2018.

Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on graph neural networks via meta learning. In *Proc. Int. Conf. Learning Representations*, 2019.

A SUPPLEMENTARY MATERIALS

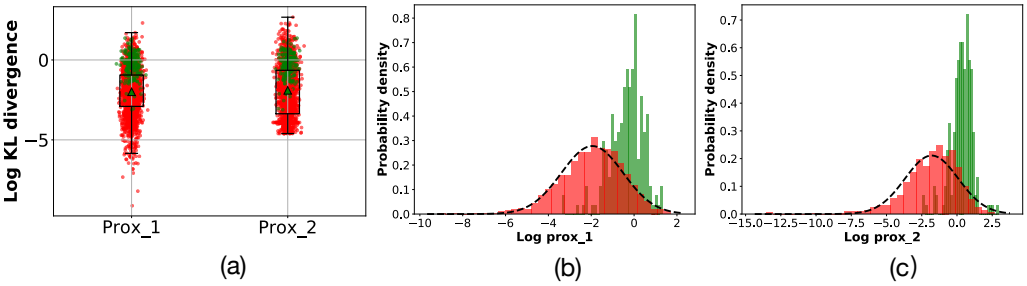


Figure 4: Box plots of (a) prox1 and prox2 statistics after log transform for unperturbed nodes and perturbed nodes. (c) and (d), using Gaussian distributions to fit the log-transformed statistics from the unperturbed nodes (Cora dataset).

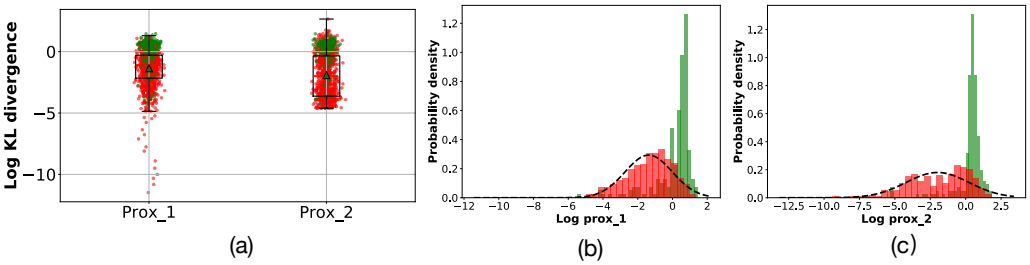


Figure 5: Box plots of (a) prox1 and prox2 statistics after log transform for unperturbed nodes and perturbed nodes. (c) and (d), using Gaussian distributions to fit the log-transformed statistics from the unperturbed nodes (Polblogs dataset).

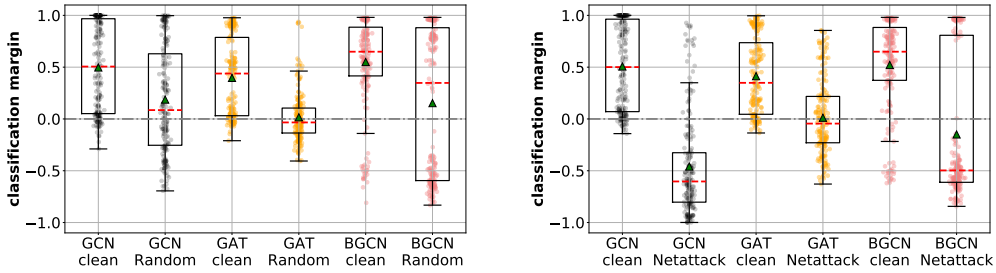


Figure 6: Comparison of random attack (left) and Nettack (right) on GCN, GAT and BGCN on Cora dataset.

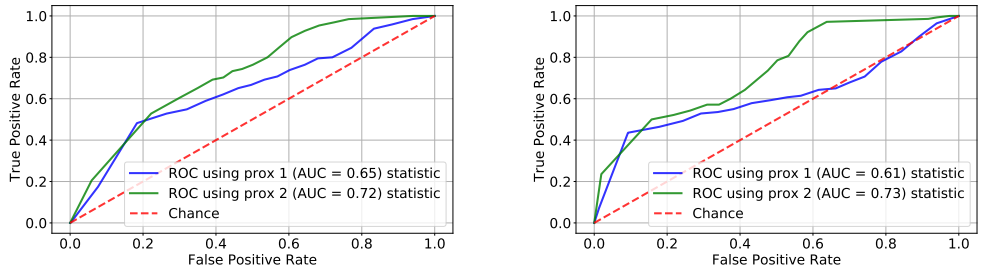


Figure 7: ROC curve for Cora, Citeseer using the proposed detection approach (under Meta-attack)