# Defenses Against Adversarial Attacks on Neural Networks for Graph Data via Encryption and Disturbing

Ms.Chikka Harshini [1], Billa Pavani [2], Dasari Sindhuja [3], Bandi Divya [4]

[1] Assistant Professor, Department of CSE, Malla Reddy Engineering College for Women, Autonomous, Hyderabad

[2], [3], [4] Student, Department of CSE, Malla Reddy Engineering College for Women, Autonomous, Hyderabad

## ABSTRACT:

The recent progress in exploiting neural networks based on graphs (GNNs) and safeguarding nodes anonymity on graph data has garnered a lot of interest. These two crucial functions are not yet integrated by the eye. Envision a scenario where an adversary in a community of people may deduce users' private labels using the strong GNNs. How can we protect disturbed graphs against privacy attacks in an adversarial way without sacrificing their usefulness? To combat adversarial defenses to GNN-based privacy assaults, a new area of study, we introduce NetFense, a graph perturbation-based method. At the same time that it can preserve data utility by reducing the prediction confidence of private label categorization and keep graph data undetected capacity (i.e., having limited changes on the graph framework), NetFense can also reduce forecasting confidence of targeted label classification and protect node privacy. The perturbed graphs generated by NetFense can successfully preserve data utility (i.e., model unnoticed ability) on targeted label classification while drastically lowering the prediction confidence of private label categorization (i.e., privacy protection), according to experiments performed on ingle- and multiple-target perturbations using three real graph datasets. The adaptability of NetFense, the maintenance of local neighborhoods in data undetected capabilities, and improved privacy protection for high-degree nodes are only a few of the discoveries that have been uncovered by extensive experiments.

## INTRODUCTION

There has been a lot of interest in using GRAPH data in practical contexts, including reference systems, networks of friends, and knowledge networks. Graphs may show the connections between nodes as well as their attributes. Since deep learning's inception, graph-based neural networks (GNNs) have been the de facto standard for learning and representing graph nodes. For tasks including node categorization, link forecasting, and community discovery, GNN encodes patterns from node characteristics, collects neighbor representations based on edge connections, and creates effective embeddings. Examples of common GNN methods include semi-supervised learning using graph convolution and relational feature generation using input features and a columnar network. To further assess the impact of incident edges, graph attention is created. Improving GNNs' representational capacity has also been theorized via the use of information aggregation. If an opponent views private labels (such as gender and age) as a neutral Report Phrase label, we should be worried about the leaking of private information due to powerful GNNs. For instance, even while social media platforms like LinkedIn, Facebook, and Twitter include privacy options, users still face the risk of partial data leaking unless they knowingly activate these settings or provide access to third-party applications. Because the enemy only has access to incomplete data, they may train GNNs to infer and steal sensitive information. Using user-generated messages on Twitter to determine the visited location and user-generated data for age and gender prediction are two applications of GNNs. The privacy-protected graph perturbation is clarified. To start with, we need a way to change the supplied graph by deleting a benefit and adding another one such that two things can be satisfied: (1) the danger of privacy leakage can be reduced and the prediction confidence (y-axis) on private labels (square and circle) may be decreased. (2) It is possible to retain the data useful while preserving the prediction confidence on the targeted labels (i.e., light green and yellow). Yes, exactly: If these two goals are to be met, the suggested NetFense model outperforms Netteck's clean and perturbed data. data pertaining to online purchases. A common method for reducing the likelihood of private data leaks is differential privacy (DP), which entails injecting an algorithm with noise. In order to lower the performance of prediction of links and node classification, DPN and PPGD create shallow embedding models based on DP. Even with these two models, there is still a good chance that private information might be exposed, as the initial graph information cannot be changed. To further explain the concept, we refer to Figure 1 (left). The graph is disrupted by deleting one edge and inserting another one using a well-designed defensive model. For private labels, we anticipate that the new graph will distort inference by lowering prediction confidence, while for target labels, it will preserve data usefulness by retaining prediction confidence. Actually, an attacker may train a model using publicly accessible data from certain people' profiles on social media websites like Instagram and Twitter. Unfortunately, not every user takes protecting their privacy very seriously. Consequently, there are two reasons why one's personal qualities and relationships might be revealed. The first issue is that users could accidentally expose their privacy by making certain fields public. Second, some users are ready to market themselves and maximize their exposure by supplying complete personal details, regardless of whether other individuals acquire their private information or not. Attackers may train their attack model using data obtained from these users. So, we want to keep the data useful while we identify and correct the data's weak spots that might lead to the attack model's estimated risk of privacy disclosure. Then, attackers would be able to see the data that has had its privacy protected, meaning they would not be able to utilize the attack model to reveal users' private labels. The research that is most applicable is Nettack. The goal of creating a gradient-based attack model is to drastically lower the performance of a job (like node classification) by perturbing the graph topology and node attributes. But there are two ways in which Nettack fails to meet privacy requirements. The first is that an opponent, knowing that there is some kind of security, may extract the real value by reversing the misclassified labels when the private label being targeted is binary. Secondly, while Nettack may be used to protect against privacy attacks by reducing performance, it doesn't ensure that the disturbed data will be useful for inferring non-private labels. Using real-world graph data, as seen in Fig. 1 (right), reveals that Nettack causes private label misclassification while failing to preserve prediction confidence for the target label. You can see the difference in forecast probabilities among the ground truth and the second most likely label on the y-axis, which represents the classification margin. The confidence of a model's predictions is another way to look at it. If the value is negative, it is more likely to be selected as the second most likely label. This study presents a new adversarial defense issue for graph data, specifically one that targets privacy attacks. Our goal is to enhance data utility while simultaneously protecting privacy in a graph where each node has a vector of features, a targeted label (like a topic or category), and an exclusive (like a person's gender or age). We can achieve this by adding or removing edges from the graph. More specifically, we want to protect the data's usefulness under GNNs when it's disclosed by decreasing the forecasting confidence on the confidential label to avoid adversaries' GNNs inferring privacy and keeping the prediction confidence on the intended label. Protecting against the model attack that learns to infer private labels is one way to look at this job. We make Table 1 to show how our suggested privacy protection, NetFense, differs from the model attack, Nettack, on graph data. First, the data owner's privacy protection and the adversary's model assault both have distinct horizons in terms of the data they may access. Secondly, as stated earlier, our issue is attempting to deal with two tasks simultaneously. One of these tasks is summarizing the differences between model attacks and privacy defenses on graph data in terms of the following: WHO is doing the attack or defense, accessible data, strategy, perturbation objective, non-noticeable perturbation, number of tasks tackled, and number of targets concerned. For predictions, "Pred-Acc" and "Pred-Confi" stand for accuracy and confidence, respectively. "on" denotes service.

## RELATED WORK

### Protecting sensitive attributes via privacy-aware recommendation using adversarial learning

A crucial application that assists users in finding material according to their interests is recommendation. On the other hand, suggestions may be used by an attacker to deduce users' private information. Previous research has included masking user-item data before sending it to recommendation systems. While this method obfuscates data, it does not tackle the quality of recommendations head-on. As an added downside, it does not safeguard consumers against recommendation-based private-attribute inference attacks. To our knowledge, this is the first effort to construct a RAP model that can defend against private-attribute inference assaults while still making meaningful product recommendations. Our major strategy is to frame this issue as a Bayesian customized recommender vs. an adversarial educational problem involving private attribute inference. Using the user's wish list and suggested goods, the attacker hopes to deduce private-attribute information. In order to regularize the suggestion process, the recommender employs the attacker, whose goal is to extract users' interests. The suggested strategy safeguards consumers against private-attribute inference assaults while also maintaining high-quality recommendation services, according to experiments.

**Improving the reliability of artificial intelligence systems by transforming data**

As a precaution against ML classifier evasion assaults, we suggest using data modifications. Our goal is to strengthen machine learning by introducing data transformations such as decreasing dimensionality via principal component analysis and data 'anti-whitening' into both the training and classification phases. We discuss and analyze several methodologies for this purpose. Using a variety of real-world datasets, we test the viability of linear data transformations as a countermeasure against evasion attempts and provide empirical evaluations. Our main results show that the defense is (i) able to withstand the most well-known evasion attacks in the literature, which doubles the amount of resources needed by a competitor with knowledge of the defense to succeed in an attack; (ii) compatible with various ML classifiers, such as SVMs and DNNs; and (iii) applicable to various application domains, like picture and human activity classification.

**Collective data-sanitization to protect social networks from assaults using sensitive information inference**

An egregious invasion of privacy may result from the release of social network data. Inherently private are user profiles and friendship ties. The use of data mining tools to anticipate sensitive information from publicly available data is a major concern. Hence, it is essential to sanitize network data before releasing it. Using social networks that include both sensitive and non-sensitive information, we investigate ways to conduct an inference attack. In order to solve this issue, we formulate a collective reasoning model based on a collective categorization problem. Based on our methodology, an attacker may exploit a disclosed social network dataset to anticipate sensitive information about associated victims by combining user profiles and social ties. We provide a data sanitization approach that deals with these kinds of assaults by collectively altering user profiles and friendship associations. In addition to cleaning up friendship connections, the suggested approach may make use of a number of data manipulation techniques. We demonstrate that it is simple to lower the accuracy of the adversary's predictions on sensitive data, with a smaller impact on non-sensitive data, and apply this method to three social media datasets. In order to prevent inference assaults in social networks, this study is the first to use collective approaches that combine several data-manipulating techniques with social interactions.

### Graph adversarial learning: a comprehensive review

Several graph analysis tasks, such as identifying nodes, link forecasting, and graph clustering, have been accomplished with outstanding performance by deep learning models trained on graphs. But when tested with well-designed inputs—a.k.a. adversarial examples—they reveal ambiguity and instability. There has been a flurry of activity in network adversarial learning as a result of a series of research focusing on both attacks and defenses in various graph analysis tasks. A complete overview and unified issue description are still missing from the thriving literature. Our goal is to fill this void by methodically reviewing and synthesizing previous research on network adversarial learning problems. In particular, we provide a comprehensive overview of the literature on both attacking and defending in graph study tasks, while also defining and classifying relevant concepts. Additionally, we survey and describe them thoroughly while stressing the significance of associated assessment measures. We hope that the relevant scholars will find our works useful in providing a thorough overview and insights. We have updated our GitHub repository to reflect the most recent developments in graph adversarial learning.

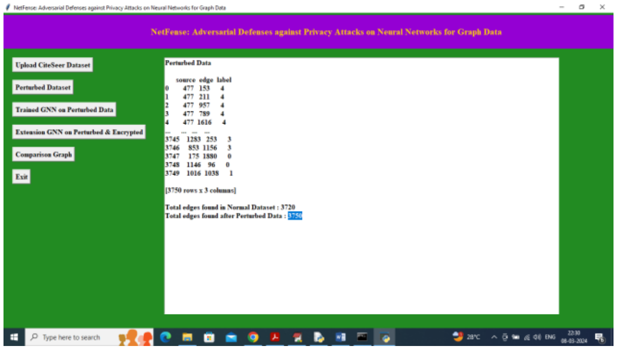
## METHODOLOGY

To implement this project we have designed following modules

1. It is to upload the CiteSeer dataset to the program. This module will then record the size of the dataset.
2) Dataset Disruption: This module will alter the dataset by inserting and deleting edges, allowing us to see the resulting changes in dataset size.
3.we'll use perturbed data to train a model in GNN, and then we'll apply that model to test data to determine its correctness. The incorrectly estimated percentage will be treated as the dataset's privacy margin. More data security will be gained in the records where the GNN made incorrect predictions.
4. we may use the GNN method to perturbed and encrypted data in order to train a model, and then we can apply this model to test data in order to determine its correctness. The

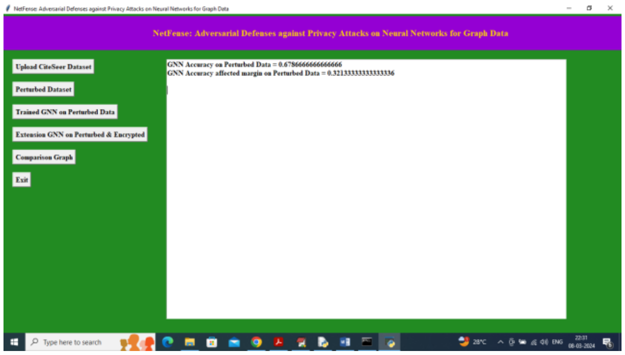
incorrectly estimated percentage will be treated as the dataset's privacy margin. More data security will be gained in the records where the GNN made incorrect predictions. 5. we will create a comparison graph showing the training loss of the propose and extend algorithms. The more efficient an algorithm is, the less its loss.

## RESULT AND DISCUSSION


NetFense: Adversarial Defenses against Privacy Attacks on Neural Networks for Graph Data
NetFense: Adversarial Defenses against Privacy Attacks on Neural Networks for Graph Data
Upload CiteSeer Dataset
Perturbed Dataset
Trained GNN on Perturbed Data
Extension GNN on Perturbed & Encrypted
Comparison Graph
Exit
Type here to search
28°C
ENG
22:30
08-03-2024

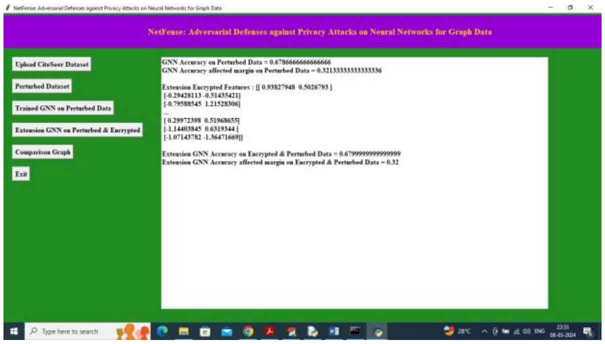
```
Perturbed Data

     source  edge  label
0       477   153      4
1       477   211      4
2       477   957      4
3       477   789      4
4       477  1616      4
...     ...   ...    ...
3745   1283   253      3
3746    853  1156      3
3747    175  1880      0
3748   1146    96      0
3749   1016  1038      1

[3750 rows x 3 columns]

Total edges found in Normal Dataset : 3720
Total edges found after Perturbed Data : 3750
```

In above result can see now dataset having source node, connecting edge node name and label and after perturbing edges size increase to 3750 from 3720 and now click on 'Trained GNN on Perturbed Data' button to trained GNN algorithm on perturbed data and get below output


NetFense: Adversarial Defenses against Privacy Attacks on Neural Networks for Graph Data
NetFense: Adversarial Defenses against Privacy Attacks on Neural Networks for Graph Data
Upload CiteSeer Dataset
Perturbed Dataset
Trained GNN on Perturbed Data
Extension GNN on Perturbed & Encrypted
Comparison Graph
Exit
Type here to search
28°C
ENG
22:31
08-03-2024

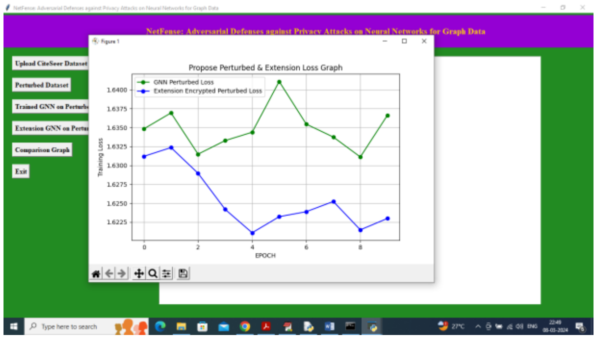
```
GNN Accuracy on Perturbed Data = 0.6786666666666666
GNN Accuracy affected margin on Perturbed Data = 0.32133333333333336
```

In above result GNN got 67% accuracy and incorrect prediction margin is 32% so above screen accuracy is reduced but data will be having more privacy as algorithm able to predict 67% records correctly. Now click on 'Extension GNN on Perturbed & Encrypted' button to encrypt data and then train with GNN to further reduce accuracy and to increase privacy


NetFense: Adversarial Defenses against Privacy Attacks on Neural Networks for Graph Data
NetFense: Adversarial Defenses against Privacy Attacks on Neural Networks for Graph Data
Upload CiteSeer Dataset
Perturbed Dataset
Trained GNN on Perturbed Data
Extension GNN on Perturbed & Encrypted
Comparison Graph
Exit
Type here to search
28°C
ENG
22:33
08-03-2024

```
GNN Accuracy on Perturbed Data = 0.6786666666666666
GNN Accuracy affected margin on Perturbed Data = 0.32133333333333336

Extension Encrypted Features : [[ 0.93827948  0.5026793 ]
 [-0.29428113 -0.51435421]
 [-0.79588545  1.21528306]
 ...
 [ 0.29972398  0.51968655]
 [-1.14403845  0.6319344 ]
 [-1.07143782 -1.36471669]]

Extension GNN Accuracy on Encrypted & Perturbed Data = 0.6799999999999999
Extension GNN Accuracy affected margin on Encrypted & Perturbed Data = 0.32
```

In above result extension accuracy is increased in points and data is having encrypted points so data will be more secured and in above screen can see some encrypted values. So with extension technique by adding encrypted training we can provide more security or privacy to data. Now click on ‘Comparison Graph’ link to get below graph


NetFense: Adversarial Defenses against Privacy Attacks on Neural Networks for Graph Data
NetFense: Adversarial Defenses against Privacy Attacks on Neural Networks for Graph Data
Upload CiteSeer Dataset
Perturbed Dataset
Trained GNN on Perturbe
Extension GNN on Pertu
Comparison Graph
Exit
Figure 1
Propose Perturbed & Extension Loss Graph
GNN Perturbed Loss
Extension Encrypted Perturbed Loss
Training Loss
1.6400
1.6375
1.6350
1.6325
1.6300
1.6275
1.6250
1.6225
0
2
4
6
8
EPOCH
Type here to search
27°C
ENG
22:49
08-03-2024

In above graph x-axis represents training epochs and y-axis represents training loss and then green line represents ‘GNN Perturbed Loss’ and blue line represents ‘Extension Encrypted Perturbed Loss’. In both algorithms can see Extension got less loss so it will provide more security or privacy to data.

## CONCLUSION

In this article, a new area of study is introduced: adversarial defenses against privacy attacks using graph neural networks in a semi-supervised learning context. By contrasting the suggested issue with model assaults on graph data, we find that perturbed graphs should be able to preserve data unnoticed ability, model unnoticed ability (data utility), and privacy protection simultaneously. We create an adversarial method called NetFense and show through experiments that graphs that it perturbs can simultaneously reduce the forecasting trust of private label classification, keep the accuracy of aimed label classification, and cause the least change to local graph structures. Additionally, we show that compared to perturbing node characteristics, disturbing edges causes a greater amount of harm when it comes to misclassifying private labels. Also, although model attack techniques like Nettack aren't great at handling multi-target perturbations, the suggested NetFense really shines when it comes to single-target perturbations. Evaluation findings also show that modest edge disruption might affect the structure of graphs to prevent privacy leaking via GNNs and reduce graph data destruction. In addition, we provide analysis of performance-related hyperparameters and perturbation factors. We hope that the results of this study will inspire researchers to look into the link between the disclosure of many private labels and attributed graphs, as well as how to design privacy-preserved graph neural networks.

## REFERENCES

[1] Ghazaleh Beigi, Ahmadreza Mosallanezhad, Ruocheng Guo, Hamidreza Alvari, Alexander Nou, and Huan Liu.

Privacy-aware recommendation with private-attribute protection using adversarial learning. In Proceedings of the 13th International Conference on Web Search and Data Mining, WSDM ’20, pages 34–42, 2020.

[2] Arjun Nitin Bhagoji, Daniel Cullina, Chawin Sitawarin, and Prateek Mittal. Enhancing robustness of machine learning systems via data transformations. In 52nd Annual Conference on Information Sciences and Systems (CISS), pages 1–5, 2018.

[3] Aleksandar Bojchevski, Johannes Klicpera, Bryan Perozzi, Martin Blais, Amol Kapoor, Michal Lukasik, and Stephan Gunnemann. ¨ Is pagerank all you need for scalable graph neural networks? In Proceedings of the 15th international workshop on mining and leaning with graphs, 2019.

[4] Zhipeng Cai, Zaobo He, Xin Guan, and Yingshu Li. Collective data-sanitization for preventing sensitive information inference attacks in social networks. IEEE Transactions on Dependable and Secure Computing, 15(4):577–590, 2016.

[5] Liang Chen, Jintang Li, Jiaying Peng, Tao Xie, Zengxu Cao, Kun Xu, Xiangnan He, and Zibin Zheng. A survey of adversarial learning on graphs. arXiv preprint arXiv:2003.05730, 2020.

[6] Weijian Chen, Yulong Gu, Zhaochun Ren, Xiangnan He, Hongtao Xie, Tong Guo, Dawei Yin, and Yongdong Zhang. Semi-supervised user profiling with heterogeneous graph attention networks. In Proceedings of the 28th International Joint Conference on Artificial Intelligence, pages 2116–2122, 2019.

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

[8] Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. Adversarial attack on graph structured data. In Proceedings of the 35th International Conference on Machine Learning, pages 1115–1124, 2018.

[9] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E Kounavis, and Duen Horng Chau. Shield: Fast, practical defense and vaccination for deep learning using jpeg compression. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 196–204, 2018.

[10] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Theory of cryptography conference, pages 265–284. Springer, 2006.

[11] Negin Entezari, Saba A. Al-Sayouri, Amirali Darvishzadeh, and Evangelos E. Papalexakis. All you need is low (rank): Defending against adversarial attacks on graphs. In Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM), 2020.

[12] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In 3rd International Conference on Learning Representations, ICLR, 2015.

[13] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In 5th International Conference on Learning Representations, ICLR, 2017.

[14] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Gunnemann. Predict then propagate: Graph neural networks ¨ meet personalized pagerank. In Proceedings of International Conference on Learning Representations (ICLR), 2019.

[15] Changchang Liu and Prateek Mittal. Linkmirage: Enabling privacy-preserving analytics on social relationships. In The Network and Distributed System Security Symposium, 2016.