PerCBA: Persistent Clean-label Backdoor Attacks on Semi-Supervised Graph Node Classification

August 2023

Xiao Yang

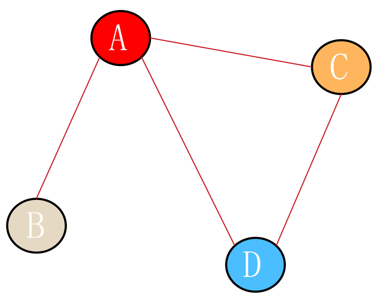
上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Background

- **Graph neural network**

Graph neural network (GNN) is a class of artificial neural networks for <u>processing data that can be represented as </u>graphs. The key design element of GNNs is the use of pairwise message passing (aggregation), such that graph nodes iteratively update their representations by exchanging information with their neighbors.

I. Aggregation



Update node data

A = A + B + C + D

B = B + A

C = C + A + D

D = D + A + C

II. a ➢ Graph classification: $y = h\,(Z)$ $\quad Z = \mathit{Readout}\,(A, B, C, D)$

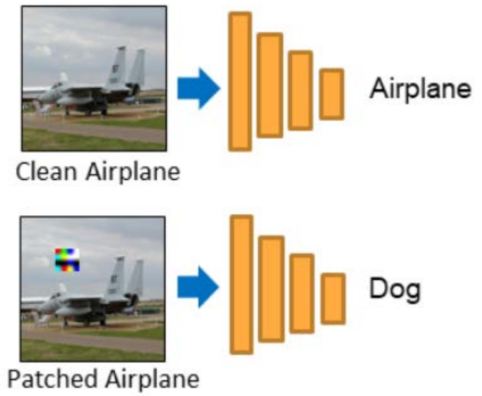
II. b ➢ Node classification: $y = h\,(A, B, C, D)$

Background

- **Backdoor**

Backdoors are attacks that manipulate the behavior of AI model. It implants **triggers** in the machine learning model during the training phase and damage the model performance when it comes to special condition.

- Attack the model via poisoning train data.
- Model get backdoored via training.
- Backdoor will not behave in normal data when test.
- Backdoor behaves when test data embedded with trigger.



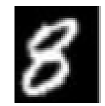
Test backdoored model in test

Backdoor

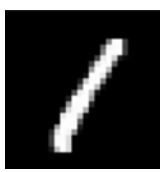
- **Feature learning of backdoored model**

Label: 8

learn → 2 rings

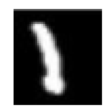


test → Number 8



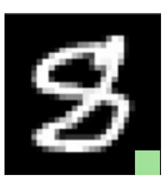
Label: 1

learn → 1 line



test → Number 1

Poisoned data



← trigger

Revised label: 1

learn → 2 rings and square

test → Number 1

Shortcomings of GNN Backdoor

- **Attack targets:**

The attack targets are only labeled data, but the vast majority of train data in real-world learning scenarios are unlabeled, which is much harder to detect anomaly.

- **Poison rate:**

Existing attack methods require relatively high infection rates (mostly between 5% and 30%) to achieve preferable results.

Method	Task	Target	Poison rate
GTA[1]	Graph	Labeled	5%
SBA[2]	Graph	Labeled	10%
EBA[3]	Graph	Labeled	20%
NBA[4]	Graph	Labeled	25%

- **Graph classification:**

Most backdoor attacks focus on graph classification, rather than node classification.

Method

- Backdoor Attacks on Semi-Supervised GNN for Node Classification

➢ **Task**:

This attack method mainly targets the **node classification** task.

➢ **Target**:

Unlabeled data in the attack data, and we do not modify the tagging information.

➢ **Poison rate**:

Only a relatively **low poison rate** is required to achieve results.

Method

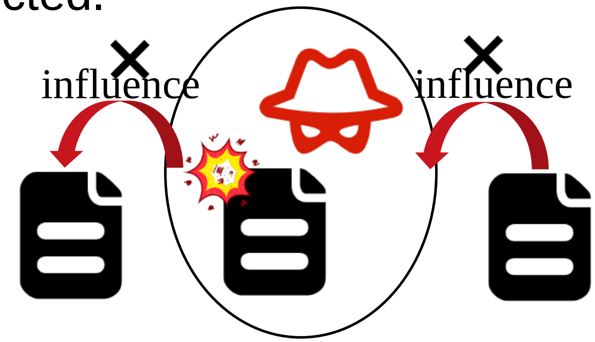
First step: find ideal Attack Target

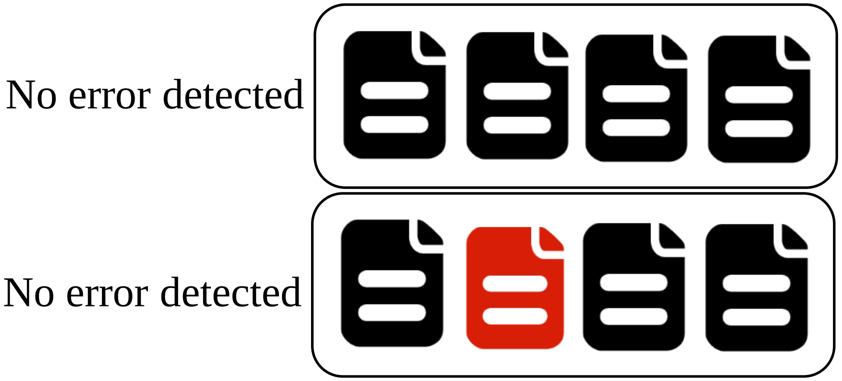
➤ **Low influence**:

The impact of attacking it on other nodes is small.

➤ **High concealment**:

The selected node should not be too conspicuous, otherwise it will be easily detected.



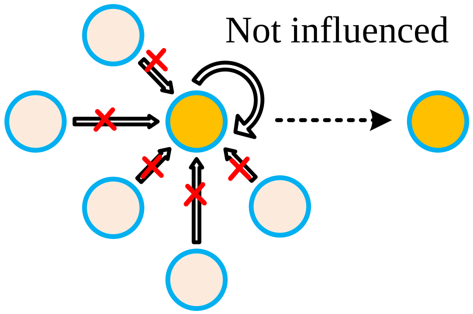


Method

Independent Nodes satisfy the requirements!

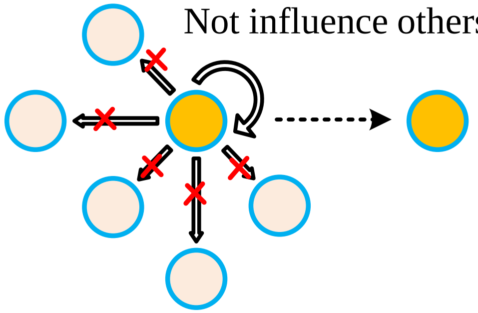
- **Influence**:

Independent nodes do not passing messages to or receive messages from other nodes during the aggregation process.



- **Existence**:

Widespread presence of independent nodes in unlabeled data.



Method

Second step: insert trigger

- **Target**: node feature vector of train data.
- **Pattern**: several sub-features in vector (uniformly select m sparse features).
  - Uniform features are less likely to be detected than dense features.
  - Dense features are easily associated with a particular class than uniform features.

$$u_i^{\delta} = u_i + \Delta = (a_i, x_i + \Delta)$$
$$s.t.\ \ x_i + \Delta = (\rho_1, \rho_2, ..., 1_1, ..., 1_2, ..., 1_m, ..., \rho_k)$$

trigger place

e.g. feature vector $\boxed{0.1, 0.2, 0.7, 0.4, 1, 0.2, 1, 0.8, 1, 0.9}$

Note that we do not modify or add any label information!

Method

How to correlate poisoned data with target class?

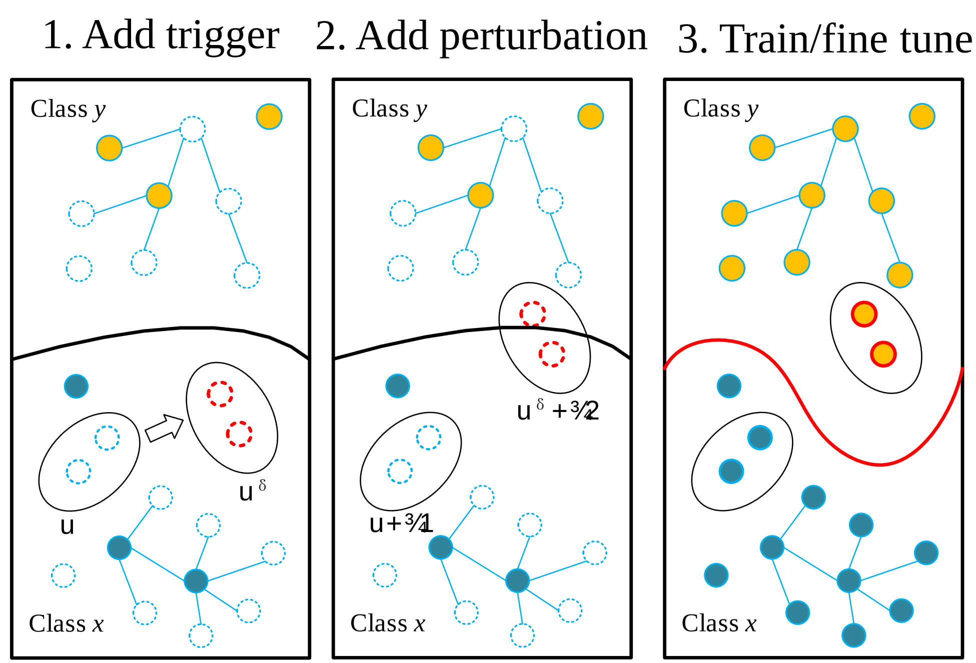
➤ Perturbation

Adding slight noise to poisoned data makes them close to the target class and thus gradually crosses the decision boundary.

➤ Formally defined as

$$\sigma_i = \arg\min_{\sigma} \| l(u_i^{\delta} + \sigma, \theta) - l(s_t, \theta) \|_2$$

$$s.t. \;\; \| u_i^{\delta} + \sigma \|_2 < \epsilon$$



Method

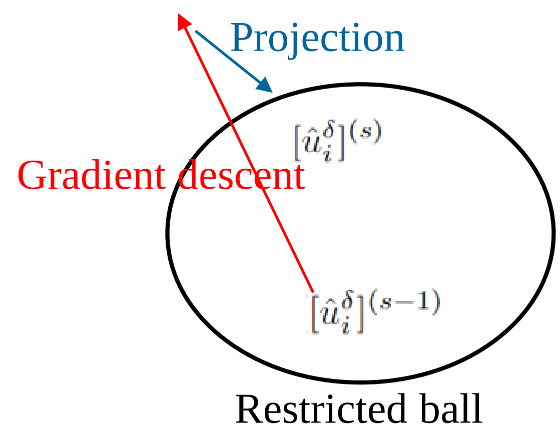
How to solve perturbation finding?

➢ Projected gradient descend

To address this adversarial perturbation problem, gradient based method projected gradient descend (PGD) will be employed. PGD solves for the target parameters finding by iteratively computing gradient descent, and the acquired gradient will be projected to a smaller spherical range to ensure guarantee the data size.

$$[\hat{u}_i^\delta]^{(s)} = \Pi_p(\underbrace{[\hat{u}_i^\delta]^{(s-1)} - \mu\tau^{(s)})}_{\text{Gradient descent}}$$

$$\Pi_p(\hat{u}_i^\delta) = \underbrace{\arg\min_{u\in\Gamma} \|u - \hat{u}_i^\delta\|_2}_{\text{Projection}}$$

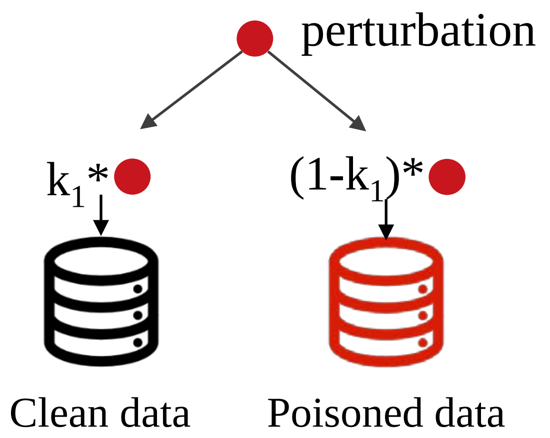


Method

Perturbation influences the model? Yes, so make it robust!

➢ Perturbation strategy

Besides adding very slight perturbation to the poisoned data, we also add a very small perturbation to the normal training data, making the model robust to perturbation.



➢ Formally defined as

$$\tilde{u}_i = u_i + k_1\sigma_i$$

$$\tilde{u}_i^\delta = u_i^\delta + (1-k_1)\sigma_i,$$

Note that k_1 is very small!

Ablation Experiment

➢ Datasets:

A: Cora

B: Citeseer

C: Pubmed

D: Random graph

Dataset	Node	Edge	Class	Feature	Label Rate
A	2,708	5,429	7	1,433	0.052
B	3,327	4,732	6	3,703	0.036
C	3,943	3,815	3	500	0.040
D	3,000	90,000	5	500	0.100

➢ Results:

Dataset	Original Data Acc	Original Data MR	Poison Rate	Acc	ASR	ADD (‰)	AEC (‰)	AFD (‰)
A	73.78	3.4	3.69	70.77	60.20	0.058	0.021	0.9
B	66.25	7.1	3.0	65.85	34.08	0.118	0.050	0.09
C	72.19	9.1	0.5	69.86	**71.01**	0.0006	0.00082	0.24
D	18.60	0.06	3.3	21.31	27.85	0.0018	0.00052	0.27

ASR: attack success rate

Acc: accuracy

MR: misclassification rate

ADD: average degree change

AEC: average eigenvector centrality change

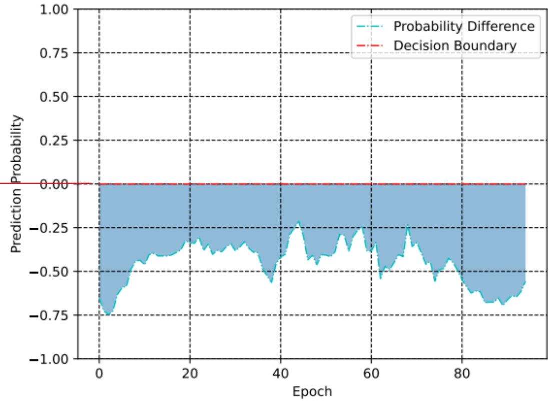
AFD: average feature change

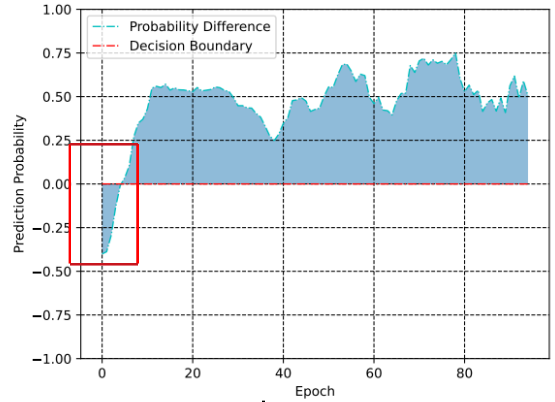
Ablation Experiment

How decision boundary changes?

Target class

Right class

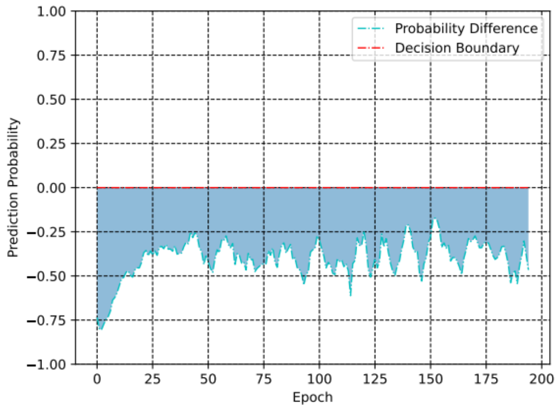


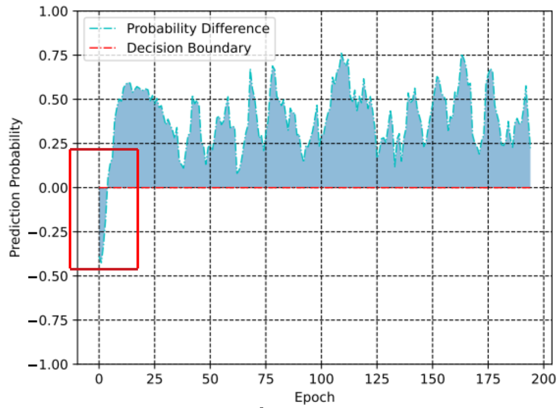


100 epoch

Pre-train

Fine-tune





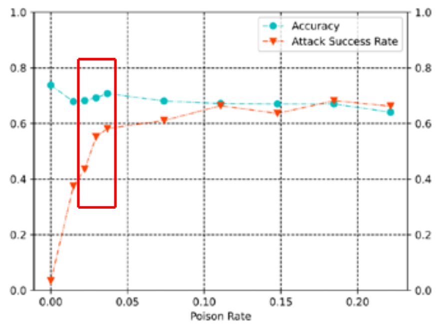
200 epoch

Pre-train

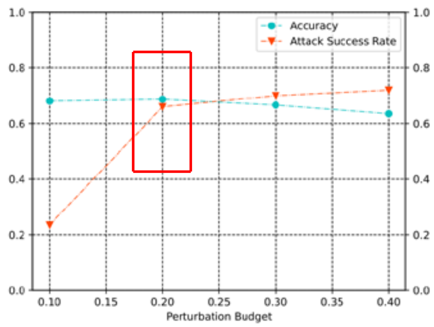
Fine-tune

Ablation Experiment

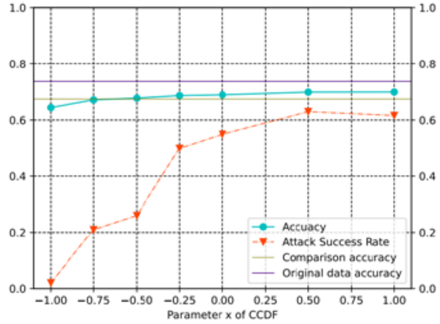
Parameters



- **Poison rate**



- **Perturbation**



- **Clean data perturbation**

Ablation Experiment

Is the method sensitive to some kinds?

- Attack different kinds

Class Change	Accuracy	Attack Success Rate
1→0	67.39	71.36
2→0	68.47	62.03
3→0	67.64	64.88
4→0	68.05	52.27
5→0	66.45	73.79
6→0	68.17	44.37

- Set different targets

Target Class	Accuracy	Attack Success Rate
0	69.52	60.38
1	68.14	65.10
2	69.77	39.43
3	68.14	79.05
4	69.40	61.27
5	67.33	66.09
6	68.14	51.54

Conclusion

- We propose Backdoor Attacks on Semi-Supervised GNNs for Node Classification, which inserts perturbated triggers into independent nodes in the graph.

- This is an attack method for the node classification task.

- The proposed method only poisons unlabeled data and will not modify label information.

- The method we propose requires only a relatively small poison rate to achieve preferable results.

Reference

- Xi Z, Pang R, Ji S, et al. Graph backdoor[C]//30th USENIX Security Symposium (**USENIX Security 21**). 2021: 1523-1540.
- Zhang Z, Jia J, Wang B, et al. Backdoor attacks to graph neural networks[C]//Proceedings of the 26th **ACM Symposium on Access Control Models and Technologies**. 2021: 15-26.
- Yang S, Doan B G, Montague P, et al. Transferable graph backdoor attack[C]//Proceedings of the 25th **International Symposium on Research in Attacks, Intrusions and Defenses**. 2022: 321-332.
- Xu J, Xue M, Picek S. Explainability-based backdoor attacks against graph neural networks[C]//Proceedings of the 3rd **ACM Workshop on Wireless Security and Machine Learning**. 2021: 31-36.
- Chen L, Peng Q, Li J, et al. Neighboring Backdoor Attacks on Graph Convolutional Network[J]. arXiv preprint arXiv:2201.06202, 2022.

Thanks!