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

August 2023
Xiao Yang
Graph neural network

Graph neural network (GNN) is a class of artificial neural networks for processing data that can be represented as 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

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

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
- Learn 2 rings and square
- Trigger
- Revised label: 1

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Task</th>
<th>Target</th>
<th>Poison rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>GTA[1]</td>
<td>Graph</td>
<td>Labeled</td>
<td>5%</td>
</tr>
<tr>
<td>SBA[2]</td>
<td>Graph</td>
<td>Labeled</td>
<td>10%</td>
</tr>
<tr>
<td>EBA[3]</td>
<td>Graph</td>
<td>Labeled</td>
<td>20%</td>
</tr>
<tr>
<td>NBA[4]</td>
<td>Graph</td>
<td>Labeled</td>
<td>25%</td>
</tr>
</tbody>
</table>

▪ **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.
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.

\[
\begin{align*}
\vec{u}_i^\delta &= \vec{u}_i + \Delta = (a_i, x_i + \Delta) \\
\text{s.t. } x_i + \Delta &= (\rho_1, \rho_2, \ldots, 1_1, \ldots, 1_2, \ldots, 1_m, \ldots, \rho_k)
\end{align*}
\]

*e.g.* feature vector $[0.1, 0.2, 0.7, 0.4, 1, 0.2, 1, 0.8, 1, 0.9]$

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

\[
\sigma_i = \arg \min_{\sigma} \|l(u_i^\delta + \sigma, \theta) - l(s_t, \theta)\|_2 \\
\text{s.t. } \|u_i^\delta + \sigma\|_2 < \epsilon
\]
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.

\[
[\hat{u}_i^\delta]^{(s)} = \Pi_p([\hat{u}_i^\delta]^{(s-1)} - \mu T^{(s)})
\]

Gradient descent

\[
\Pi_p(\hat{u}_i^\delta) = \arg \min_{u \in \Gamma} ||u - \hat{u}_i^\delta||_2
\]

Projection

Projection

Restricted ball
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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Node</th>
<th>Edge</th>
<th>Class</th>
<th>Feature</th>
<th>Label Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2,708</td>
<td>5,429</td>
<td>7</td>
<td>1,433</td>
<td>0.052</td>
</tr>
<tr>
<td>B</td>
<td>3,327</td>
<td>4,732</td>
<td>6</td>
<td>3,703</td>
<td>0.036</td>
</tr>
<tr>
<td>C</td>
<td>3,943</td>
<td>3,815</td>
<td>3</td>
<td>500</td>
<td>0.040</td>
</tr>
<tr>
<td>D</td>
<td>3,000</td>
<td>90,000</td>
<td>5</td>
<td>500</td>
<td>0.100</td>
</tr>
</tbody>
</table>

Results:

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

200 epoch
Ablation Experiment

Parameters

- Poison rate
- Perturbation
- Clean data perturbation
Ablation Experiment

Is the method sensitive to some kinds?

- **Attack different kinds**

<table>
<thead>
<tr>
<th>Class Change</th>
<th>Accuracy</th>
<th>Attack Success Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1→0</td>
<td>67.39</td>
<td>71.36</td>
</tr>
<tr>
<td>2→0</td>
<td>68.47</td>
<td>62.03</td>
</tr>
<tr>
<td>3→0</td>
<td>67.64</td>
<td>64.88</td>
</tr>
<tr>
<td>4→0</td>
<td>68.05</td>
<td>52.27</td>
</tr>
<tr>
<td>5→0</td>
<td>66.45</td>
<td>73.79</td>
</tr>
<tr>
<td>6→0</td>
<td>68.17</td>
<td>44.37</td>
</tr>
</tbody>
</table>

- **Set different targets**

<table>
<thead>
<tr>
<th>Target Class</th>
<th>Accuracy</th>
<th>Attack Success Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>69.52</td>
<td>60.38</td>
</tr>
<tr>
<td>1</td>
<td>68.14</td>
<td>65.10</td>
</tr>
<tr>
<td>2</td>
<td>69.77</td>
<td>39.43</td>
</tr>
<tr>
<td>3</td>
<td>68.14</td>
<td>79.05</td>
</tr>
<tr>
<td>4</td>
<td>69.40</td>
<td>61.27</td>
</tr>
<tr>
<td>5</td>
<td>67.33</td>
<td>66.09</td>
</tr>
<tr>
<td>6</td>
<td>68.14</td>
<td>51.54</td>
</tr>
</tbody>
</table>
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


Thanks!