Voucher Abuse Detection with Prompt-based Fine-tuning on Graph Neural Networks

Zhihao Wen and Yuan Fang
Singapore Management University
Singapore

Yihan Liu, Yang Guo and Shuji Hao
Lazada Inc.
Singapore
Outline

- Introduction
- Methodology
- Experiment
- Conclusion
Voucher Abuse Detection on Order Graph

Order graph encodes rich relationships and patterns between orders.

(a) Order graph

(b) Orders of legitimate user

A legitimate user typically only logs into one account on one device and applies one voucher.

(c) Orders of abusive user

An abusive user often employs many devices, and in each device, they create multiple accounts.
Existing works

• Traditional ML methods [1][2][3], do not leverage graph structure information

• Supervised GNN-based approaches [4][5][6], cannot perform well with limited labels

• Self-supervised GNNs [7][8] become promising for capture intrinsic graph patterns without annotated label.

• Self-supervised GNNs suffer from a major drawback: the objective gap between the pre-training and downstream tasks

Challenges of bridging the gap between GNN pre-training and downstream tasks

Q1: Cannot directly apply the textual prompting function to bridge various graph-based tasks

We propose a graph prompting function that reformulates the downstream node classification problem into a pairwise matching task between node tokens and context tokens

Q2: How to initialize the context tokens

We reuse the graph readout function from pre-training to initialize the context tokens downstream
Outline

- Introduction
- Methodology
- Experiment
- Conclusion
Data preparation

Two key inputs: 1) an **order graph**; 2) a small number of **labels**.

Graph construction.
- Two categories of raw data: (1) **User profiles**, and (2) **Buyer journey logs**.
- Construct an order graph based on various **shared attributes**

Limited pseudo-label generation.
- Generate limited pseudo-labels by employing a set of **predefined business rules**
Overall framework of our proposed VPGNN

(a) Pre-training: DGI-based self-supervised learning

(b) Downstream: prompt-based fine-tuning

Overall framework of VPGNN. (a) We conduct self-supervised pre-training based on DGI. (b) we perform prompt-based fine-tuning for the downstream voucher abuse detection.
Pre-training based on DGI

- In voucher abuse detection, abusive orders are the minority and the majority are legitimate orders.
- Utilize DGI to maximize the local-global mutual information, whereby the graph-level global information captures the “normal” patterns manifested by the majority.
Prompt generation

- Our prompt-based fine-tuning framework generates and tunes prompts, and makes predictions based on the prompts.

- Propose a graph prompting function $\mathcal{P}$, transforming an input node $i$ into a prompt $p_i$ consisting of a pair of node token $t_i$ and context token $z_c$.
Prompt generation

- Our prompt-based fine-tuning framework generates and tunes prompts, and makes predictions based on the prompts.

- Propose a graph prompting function $\mathcal{P}$, transforming an input node $i$ into a prompt $p_i$ consisting of a pair of node token $t_i$ and context token $z_c$
Prompt initialization

- Context tokens are learnable vectors, needing **initialization**.
- To improve the **robustness** of the initialization, we augment the labeled nodes with their **neighboring nodes**.
- To improve the **informativeness**, we reuse the graph **Readout** function from **pre-training** to pool the labeled nodes with their subgraphs.
Prompt-assisted prediction

(a) Pre-training: DGI-based self-supervised learning

- Leverage the same pretext projection head $\Phi^{\text{pre}}$ to score the matching probability of each token pair.

- Predict the order represented by node $i$ as abusive if

$$\Phi^{\text{pre}}(t_i, z_1; \phi') > \Phi^{\text{pre}}(t_i, z_0; \phi')$$
• Our prompt design allows us to reuse not only the pretext projection head, but also the **pretext task loss**.

• Our prompt-based approach **unifies** the **pretext** and **downstream** task, narrowing the gap between pre-training and downstream objectives.
Outline

- Introduction
- Methodology
- Experiment
- Conclusion
Datasets

• We collect four proprietary large-scale datasets, named VN0909, VN1010, ID0909, and ID1023, from an e-commerce platform provided by Lazada Inc.

• Each dataset is a huge order graph, where the nodes represent the orders with pre-defined features, and the edges are pre-defined relationships between them.

• VN0909 is only used for pre-training, and we do test on three other datasets.

• We also use a public dataset, namely Amazon.
## Offline performance comparison with baselines

Table 1: Performance comparison between VPGNN and the baselines, in percent, with 95% confidence intervals.

In each row, the best result is bolded and the runner-up is underlined. "/" indicates no result obtained due to out-of-memory issue or excessively long training time (>72 hours).

<table>
<thead>
<tr>
<th></th>
<th>SVM</th>
<th>XGBoost</th>
<th>MLP</th>
<th>GCN</th>
<th>SAGE\textsubscript{sup}</th>
<th>GAT</th>
<th>CARE-GNN</th>
<th>Geniepath</th>
<th>AMNet</th>
<th>DCI</th>
<th>SAGE\textsubscript{unsup}</th>
<th>Pre-train</th>
<th>VPGNN</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Shots = 10</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VN1010</td>
<td>37.1±8.9</td>
<td>65.7±5.5</td>
<td>62.9±2.8</td>
<td>59.1±6.7</td>
<td>61.9±3.6</td>
<td>60.9±4.6</td>
<td>/</td>
<td>58.0±4.5</td>
<td>/</td>
<td>/</td>
<td>61.8±4.6</td>
<td>64.8±3.6</td>
<td>67.1±3.1</td>
</tr>
<tr>
<td>ID0909</td>
<td>28.1±11.0</td>
<td>51.3±9.9</td>
<td>61.6±3.8</td>
<td>64.1±3.9</td>
<td>61.2±7.3</td>
<td>65.6±3.7</td>
<td>/</td>
<td>62.1±2.8</td>
<td>/</td>
<td>/</td>
<td>62.2±3.7</td>
<td>66.1±3.0</td>
<td>69.0±3.7</td>
</tr>
<tr>
<td>ID1023</td>
<td>38.7±8.3</td>
<td>73.5±6.1</td>
<td>69.3±2.1</td>
<td>69.3±4.8</td>
<td>71.3±5.2</td>
<td>73.7±3.6</td>
<td>73.0±2.9</td>
<td>72.0±5.0</td>
<td>70.0±3.5</td>
<td>73.4±1.6</td>
<td>67.5±5.3</td>
<td>71.8±5.2</td>
<td>75.1±1.9</td>
</tr>
<tr>
<td>Amazon</td>
<td>41.4±9.2</td>
<td>62.5±11.5</td>
<td>63.3±5.7</td>
<td>16.5±4.9</td>
<td>59.9±9.1</td>
<td>20.5±6.1</td>
<td>38.6±2.9</td>
<td>30.7±2.8</td>
<td>64.8±6.2</td>
<td>18.5±4.0</td>
<td>36.7±6.0</td>
<td>62.3±8.1</td>
<td>70.0±2.7</td>
</tr>
<tr>
<td><strong>Shots = 20</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VN1010</td>
<td>59.2±3.8</td>
<td>73.1±3.9</td>
<td>69.2±2.2</td>
<td>71.6±2.8</td>
<td>73.6±3.2</td>
<td>74.5±4.4</td>
<td>/</td>
<td>69.5±5.7</td>
<td>/</td>
<td>/</td>
<td>72.8±2.0</td>
<td>75.7±3.0</td>
<td>75.9±2.8</td>
</tr>
<tr>
<td>ID0909</td>
<td>53.1±6.1</td>
<td>64.9±3.8</td>
<td>64.9±1.7</td>
<td>68.7±2.5</td>
<td>70.3±2.8</td>
<td>71.0±3.4</td>
<td>/</td>
<td>61.4±9.1</td>
<td>/</td>
<td>/</td>
<td>67.5±2.2</td>
<td>71.4±2.5</td>
<td>72.7±2.8</td>
</tr>
<tr>
<td>ID1023</td>
<td>65.2±3.6</td>
<td>78.9±1.4</td>
<td>74.7±1.6</td>
<td>79.0±1.7</td>
<td>81.5±1.3</td>
<td>81.1±1.1</td>
<td>74.2±0.9</td>
<td>80.7±4.2</td>
<td>75.6±1.9</td>
<td>79.4±1.3</td>
<td>78.1±2.1</td>
<td>81.3±1.4</td>
<td>81.8±1.1</td>
</tr>
<tr>
<td>Amazon</td>
<td>60.3±3.6</td>
<td>72.9±8.2</td>
<td>70.4±4.4</td>
<td>16.8±8.0</td>
<td>63.0±8.5</td>
<td>48.8±10.1</td>
<td>42.2±6.7</td>
<td>30.8±4.4</td>
<td>75.1±3.1</td>
<td>21.8±2.6</td>
<td>54.5±2.8</td>
<td>73.1±3.9</td>
<td>76.6±2.5</td>
</tr>
<tr>
<td><strong>Semi-supervised</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VN1010</td>
<td>86.7±0.1</td>
<td>87.8±0.1</td>
<td>86.7±0.1</td>
<td>91.8±0.1</td>
<td>94.1±0.1</td>
<td>91.9±0.0</td>
<td>/</td>
<td>91.7±0.4</td>
<td>/</td>
<td>/</td>
<td>89.3±0.0</td>
<td>94.1±0.1</td>
<td>95.2±0.1</td>
</tr>
<tr>
<td>ID0909</td>
<td>86.0±0.2</td>
<td>89.2±0.3</td>
<td>86.8±0.3</td>
<td>92.2±0.2</td>
<td>93.4±0.2</td>
<td>92.3±0.2</td>
<td>/</td>
<td>91.1±0.4</td>
<td>/</td>
<td>/</td>
<td>86.3±0.2</td>
<td>93.3±0.2</td>
<td>94.1±0.2</td>
</tr>
<tr>
<td>ID1023</td>
<td>89.3±0.1</td>
<td>89.9±0.2</td>
<td>88.8±0.2</td>
<td>94.4±0.1</td>
<td>95.6±0.1</td>
<td>94.5±0.1</td>
<td>87.0±0.3</td>
<td>94.5±0.1</td>
<td>94.1±0.3</td>
<td>93.7±1.0</td>
<td>92.6±0.1</td>
<td>95.6±0.1</td>
<td>96.2±0.1</td>
</tr>
<tr>
<td>Amazon</td>
<td>78.8±1.1</td>
<td>75.6±2.7</td>
<td>78.1±1.8</td>
<td>34.4±3.6</td>
<td>81.1±1.5</td>
<td>73.3±2.9</td>
<td>45.2±5.6</td>
<td>30.9±4.5</td>
<td>81.7±1.1</td>
<td>26.1±4.5</td>
<td>76.1±0.9</td>
<td>80.9±0.9</td>
<td>80.6±0.7</td>
</tr>
</tbody>
</table>

- **Classical machine learning**
- **General semi-supervised GNNs**
- **Anomaly detection GNNs**
- **Pre-trained GNNs**

- **Given that pre-training** is done on **VN0909**, the superior performance of VPGNN on VN1010, ID0909 and ID1023 shows its generalization ability across time and/or markets.
- VPGNN is a strong **few-shot** learner, as it attains larger improvements relative to the runner-up under the 10-shot setting.
Online performance

Online performance in the Indonesia market over the Double 12 Campaign.

<table>
<thead>
<tr>
<th>Model</th>
<th>11 Dec 2022</th>
<th>12 Dec 2022</th>
<th>13 Dec 2022</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCP</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>LPA</td>
<td>816.9%</td>
<td>1784.1%</td>
<td>330.3%</td>
<td>809.3%</td>
</tr>
<tr>
<td>VPGNN</td>
<td>1964.1%</td>
<td>1990.3%</td>
<td>469.1%</td>
<td>998.7%</td>
</tr>
<tr>
<td>(% ↑ over LPA)</td>
<td>(140.4%)</td>
<td>(11.6%)</td>
<td>(42.0%)</td>
<td>(23.4%)</td>
</tr>
</tbody>
</table>

Metric = \[
\frac{\text{Precision of model A} \times \# \text{True positives detected by model A}}{\text{Precision of BCP} \times \# \text{True positives detected by BCP}}
\]

- VPGNN shows a 23.4% increase over LPA, demonstrating the advantage of prompt-based fine-tuning on GNNs when dealing with voucher abuse detection.
Outline

- Introduction
- Methodology
- Experiment
- Conclusion
Conclusion

Key contributions

• Addressed the problem of **voucher abuse detection** on e-commerce order graph;
• We attempted to **bridge the gap** between pretext and downstream tasks by proposing a **graph prompting function** that reformulates the downstream task to follow a **similar template** as the pretext task.