Enhancing Graph Neural Network-based Fraud Detectors against Camouflaged Fraudsters

Yingtong Dou (UIC), Zhiwei Liu (UIC), Li Sun (BUPT), Yutong Deng (BUPT), Hao Peng (BUAA), Philip S. Yu (UIC)

Email: ydou5@uic.edu
Homepage: http://ytongdou.com
Project Page: https://github.com/safe-graph
Code: https://github.com/YingtongDou/CARE-GNN
Paper Highlight

• Comprehensive **review** of GNN-based fraud detection research.

• Introduce and summarize two **fraudster camouflaging** behaviors in the wild.

• Propose **CARE-GNN** which is efficient and adaptive to many scenarios.

• **Opensource** model code, baseline code, and new dataset.
A History of Fraud

• 1990-2000: spam email, link farming.

• 2000-2010: fake review, social bots.

• 2010-2020: fake news, deepfake.

Graph-based Fraud Detection

Graph Neural Network

**GCN** [1]

- Directly aggregate neighbors using Laplacian adjacency matrix.

**GraphSAGE** [2]

- Sample and aggregate neighbors.

**GAT** [3]

- Attentively aggregate neighbors.

---


GNN-based Fraud Detectors

FdGars\textsuperscript{[1]} (GCN-based)  GAS\textsuperscript{[2]} (GAT-based)  Player2Vec\textsuperscript{[3]} (Hybrid)

\textsuperscript{[3]} Zhang, Y et, al. November. Key Player Identification in Underground Forums over Attributed Heterogeneous Information Network Embedding Framework. CIKM 2019
Camouflagging Behavior of Fraudsters

• Feature Camouflage

Spamouflage

<table>
<thead>
<tr>
<th>Tweets</th>
<th>Tweets &amp; replies</th>
<th>Media</th>
<th>Likes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shannon Foster @Shannon84865362 · Aug 8</td>
<td>you+shall+see+her+as+she+was,+and+is.&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shannon Foster @Shannon84865362 · Aug 8</td>
<td>is+beginning+to+recover+something+of+his+old+buoyancy,+so+as</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: @benimmo

Deepfake

Source: https://elgan.com/blog/deepfakes-get-real-and-real-easy

Language generation model

<table>
<thead>
<tr>
<th>Generated Reviews (Yelp)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I love this place! I’ve been here several times and I’ve never been disappointed. The food is always fresh and delicious. The service is always friendly and attentive. I’ve been here several times and have never been disappointed. I’ve been to this location twice now and both times I’ve been very impressed. I’ve tried their specialty pizzas and they’re all really good. The only problem is that they’re not open on sundays. They’re not open on sundays.</td>
</tr>
</tbody>
</table>

Camouflaging Behavior of Fraudsters

• Relation Camouflage

• Crafty fraudsters could connect to benign entities under a relation to alleviate its suspiciousness[1].

Camouflaged Fraudsters Meets GNN

**Fraudster:** relation camouflage.

**GNN:** a fraudster node may have many benign neighbors.

**Fraudster:** feature camouflage.

**GNN:** neighbors with similar features may have different labels.
Enhance GNN-based Fraud Detectors

• The fraudsters are smart and agile.

• It is difficult to exactly detect the camouflaged fraudsters.

• We propose three neural modules to enhance GNN.
Label-aware Similarity Measure

• Previous works use cosine similarity, Euclidean distance to measure the feature/embedding similarity.

• Unsupervised similarity measure cannot identify feature camouflage.

• The similarity measure must have knowledge of fraudsters.

We introduce an **MLP** to encode the label information and use its output as the similarity measure.
Similarity-aware Neighbor Selector

- For a center node, different relations may have different amount of informative neighbors.

- We propose an adaptive neighbor selector using reinforcement learning to find the optimal thresholds.

The RL process is a multi-armed bandit with following rules:
- If the avg. neighbor similarity score is greater than previous epoch, we increase the filtering threshold;
- Else, we decrease the filtering threshold.
Relation-aware Neighbor Aggregator

• We need to aggregate information across different relations.

• If we have selected informative neighbors under every relation, the attention mechanism is useless.

We directly utilize the **neighbor filtering thresholds** as the relation aggregation weights.
Experimental Setting

• Datasets:

<table>
<thead>
<tr>
<th>#Nodes (Fraud%)</th>
<th>Relation</th>
<th>#Edges</th>
<th>Avg. Feature Similarity</th>
<th>Avg. Label Similarity</th>
</tr>
</thead>
</table>
| Yelp
| 45,954 (14.5%) | R-U-R   | 49,315            | 0.83                   | 0.90                  |
|        | R-T-R   | 573,616        | 0.79                   | 0.05                  |
|        | R-S-R   | 3,402,743      | 0.77                   | 0.05                  |
|        | ALL     | 3,846,979      | 0.77                   | 0.07                  |
| Amazon
| 11,944 (9.5%) | U-P-U   | 175,608            | 0.61                   | 0.19                  |
|        | U-S-U   | 3,566,479      | 0.64                   | 0.04                  |
|        | U-V-U   | 1,036,737      | 0.71                   | 0.03                  |
|        | ALL     | 4,398,392      | 0.65                   | 0.05                  |

• Graphs: multi-relation graph with three relations.
Reinforcement Learning Process
## Overall Evaluation

### Table 3: Fraud detection performance (%) on two datasets under different percentage of training data.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Train%</th>
<th>GCN</th>
<th>GAT</th>
<th>RGCN</th>
<th>Graph-SAGE</th>
<th>Genie-Path</th>
<th>Player-2Vec</th>
<th>Semi-GNN</th>
<th>Graph-Consis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5%</td>
<td>54.98</td>
<td>56.23</td>
<td>50.21</td>
<td>53.82</td>
<td>56.33</td>
<td>51.03</td>
<td>53.73</td>
<td>61.58</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>50.94</td>
<td>55.45</td>
<td>55.12</td>
<td>54.20</td>
<td>56.29</td>
<td>50.15</td>
<td>51.68</td>
<td>62.07</td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>53.15</td>
<td>57.69</td>
<td>55.05</td>
<td>56.12</td>
<td>57.32</td>
<td>51.56</td>
<td>51.55</td>
<td>62.31</td>
</tr>
<tr>
<td></td>
<td>40%</td>
<td>52.47</td>
<td>56.24</td>
<td>53.38</td>
<td>54.00</td>
<td>55.91</td>
<td>53.65</td>
<td>51.58</td>
<td>62.07</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metric</th>
<th>Train%</th>
<th>GCN</th>
<th>GAT</th>
<th>RGCN</th>
<th>Graph-SAGE</th>
<th>Genie-Path</th>
<th>Player-2Vec</th>
<th>Semi-GNN</th>
<th>Graph-Consis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5%</td>
<td>53.12</td>
<td>54.68</td>
<td>50.38</td>
<td>54.25</td>
<td>52.33</td>
<td>50.00</td>
<td>52.28</td>
<td>62.60</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>51.10</td>
<td>52.34</td>
<td>51.75</td>
<td>52.23</td>
<td>54.35</td>
<td>50.00</td>
<td>52.57</td>
<td>62.08</td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>53.87</td>
<td>53.20</td>
<td>50.92</td>
<td>52.69</td>
<td>54.84</td>
<td>50.00</td>
<td>52.16</td>
<td>62.35</td>
</tr>
<tr>
<td></td>
<td>40%</td>
<td>50.81</td>
<td>54.52</td>
<td>50.43</td>
<td>52.86</td>
<td>50.94</td>
<td>50.00</td>
<td>50.59</td>
<td>62.08</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metric</th>
<th>Train%</th>
<th>GCN</th>
<th>GAT</th>
<th>RGCN</th>
<th>Graph-SAGE</th>
<th>Genie-Path</th>
<th>Player-2Vec</th>
<th>Semi-GNN</th>
<th>Graph-Consis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5%</td>
<td>74.44</td>
<td>73.89</td>
<td>75.12</td>
<td>70.71</td>
<td>71.56</td>
<td>76.86</td>
<td>70.25</td>
<td>85.46</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>75.25</td>
<td>74.55</td>
<td>74.13</td>
<td>73.97</td>
<td>72.23</td>
<td>75.73</td>
<td>76.21</td>
<td>85.29</td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>75.13</td>
<td>72.10</td>
<td>75.58</td>
<td>73.97</td>
<td>71.89</td>
<td>74.55</td>
<td>73.98</td>
<td>85.50</td>
</tr>
<tr>
<td></td>
<td>40%</td>
<td>74.34</td>
<td>75.16</td>
<td>74.68</td>
<td>75.27</td>
<td>72.65</td>
<td>56.94</td>
<td>70.35</td>
<td>85.50</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metric</th>
<th>Train%</th>
<th>GCN</th>
<th>GAT</th>
<th>RGCN</th>
<th>Graph-SAGE</th>
<th>Genie-Path</th>
<th>Player-2Vec</th>
<th>Semi-GNN</th>
<th>Graph-Consis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5%</td>
<td>65.54</td>
<td>63.22</td>
<td>64.23</td>
<td>69.09</td>
<td>65.56</td>
<td>50.00</td>
<td>63.29</td>
<td>85.49</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>67.81</td>
<td>65.84</td>
<td>67.22</td>
<td>69.36</td>
<td>66.63</td>
<td>50.00</td>
<td>63.32</td>
<td>85.38</td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>66.15</td>
<td>67.13</td>
<td>65.08</td>
<td>70.30</td>
<td>65.08</td>
<td>50.00</td>
<td>61.28</td>
<td>85.59</td>
</tr>
<tr>
<td></td>
<td>40%</td>
<td>67.45</td>
<td>65.51</td>
<td>67.68</td>
<td>70.16</td>
<td>65.41</td>
<td>50.00</td>
<td>62.89</td>
<td>85.53</td>
</tr>
</tbody>
</table>

### AUC of CARE-Att, CARE-Weight, CARE-Mean, and CARE-GNN

- **Online**
Model Advantage

• **Adaptability.** CARE-GNN adaptively selects best neighbors for aggregation given arbitrary multi-relation graph.

• **High-efficiency.** CARE-GNN has a high computational efficiency without attention and deep reinforcement learning.

• **Flexibility.** Many other neural modules and external knowledge can be plugged into the CARE-GNN.
SafeGraph (https://github.com/safe-graph)

• **DG Fraud**: a GNN-based fraud detection toolbox.

• **UG Fraud**: an unsupervised graph-based fraud detection toolbox.

• Graph-based Fraud Detection **Paper List**.
Enhancing Graph Neural Network-based Fraud Detectors against Camouflaged Fraudsters

Yingtong Dou (UIC), Zhiwei Liu (UIC), Li Sun (BUPT), Yutong Deng (BUPT), Hao Peng (BUAA), Philip S. Yu (UIC)

Email: ydou5@uic.edu
Homepage: http://ytongdou.com
Project Page: https://github.com/safe-graph
Code: https://github.com/YingtongDou/CARE-GNN