TGC: Transaction Graph Contrast Network for Ethereum Phishing Scam Detection

Sijia Li, Gaopeng Gou, Chang Liu, Gang Xiong, Zhen Li, Junchao Xiao, Xinyu Xing

Institute of Information Engineering, Chinese Academy of Sciences
School of Cyber Security, University of Chinese Academy of Sciences
Northwestern University

www.iie.ac.cn
Background

The Growth of Scam on Cryptocurrency

- Cryptocurrency-based crime hit a new all-time high in 2022, with illicit addresses receiving $20.6 billion over the course of the year.

- The rise of decentralized finance (DeFi) and the allure of blockchain’s anonymity have given rise to a plethora of cybercrimes

- Increased focus on cryptocurrency security issues
Background

Phishing Scams on Ethereum

- high visibility and lots of potential victims
- victims lost $645,000 within the first week of the phishing campaign, and the attacker’s illegal profits exceeded $3,000,000 in just one month
- phishing scams are the most deceptive scams

- identifying phishing scams on Ethereum becomes a crucial research topic
Background

Phishing
- exploits user vulnerabilities to obtain personal property and confidential information

Phishing Scams on Ethereum
- Take high-reward propaganda to induce remittances (email/chat)
  - offering additional Ether coins as incentives
  - visit fraudulent platforms or websites
  - promise high returns if purchase digital assets
- no fixed platforms pattern!

Inducing remittances
Transfer ETH
Address 0xf68846a99...dc8de
Limitation

Task

The address of Ethereum ----> the node in the graph
Transaction relationship ----> edge between nodes

Learn efficient node representations through the transaction network and classify nodes

Traditional Machine Learning

- Using manual-designed features as node embedding
- Limitation:
  - Rely on professional knowledge to extract features
  - Inefficient and non-automated
Limitation

Network Representation Learning
- Using graph neural network to mining deep features
- Limitation:
  - overlook the unique challenges of Ethereum phishing scams
    Weak node representation

Challenges
- The natural camouflage
  - 97.99% of neighbors are normal addresses

- Sparsity of distribution
  - low proportion (0.345%)
  - sparse distribution

- Large scale and dynamic nature
  - Ethereum transaction network is both vast and dynamic

unsatisfactory performance

ACSAC 2023
TGC: Transaction Graph Contrast Network for Ethereum Phishing Scam Detection
TGC: Transaction Graph Contrast Network for Ethereum Phishing Scam Detection

**Data collection**
- Collect relevant addresses and transactions

**Transaction Graph Contrast Network**
- Construct the sample centered \( r \)-ego networks and build two different views
- Use two designed contrastive modules to learn the discriminative representation of phishing addresses

**Phishing address detection**
- Combine the outputs and feed into the classifier to get the result

**Input**
- Etherscan API

**Preprocess**
- Every sample

**Encoder**
- GNN Encoder

**Context-level Contrast**
- Combine the outputs and feed into the classifier to get the result

**Node-level Contrast**
- Use two designed contrastive modules to learn the discriminative representation of phishing addresses

**Unique potential features**
- Statistical features

**Transaction pattern features**
- "Phishing"
Ego Network Construct & Subgraph Sampling

- Construct a local substructure of each sample node
- "r-ego network" consists of the r-order neighbors of the central sample node and the connection relationships between them
- Random walk with restart (RWR) sampling strategy
  - each ego network is sampled twice
  - generating two local subgraphs
- Each sample gets a carefully designed pair of "local subgraph vs. local subgraph"

Learn unique characteristics different from their neighbors
Node-level Contrast

- Instance pair “Target node vs. node”
- Treat neighbors as the negative samples
  - Intra-view negative pair
  - Inter-view negative pair
- Positive samples are the representation of the same node in different views
- Contrastive Loss

\[
\ell_n(x_i) = - \log \frac{e^{\theta(h_i, h_i^j)/\tau}}{e^{\theta(h_i, h_i^j)/\tau} + \sum_{k \neq i} e^{\theta(h_i, h_k)/\tau}}
\]

Learn unique characteristics different from their neighbors
TGC

Context-level contrast

- Instance pair "Target context vs. context"

- Treat subgraphs which generated by different r-ego networks as a negative pair

- Generated by the same r-ego networks as a positive pair

- Contrastive Loss

\[ \ell_c (G_i^1) = - \log \frac{e^{(c_i^T c_i^j)}/\tau}{\sum_{j=1}^{N} e^{(c_i^T c_j)}/\tau} \]

Capture the transactional structural patterns behind phishing and normal addresses
Phishing Addresses Detection

Final Node Representation – **concat the three features**

- **Unique potential features**
  - Learned from the **Node-level Contrast** module

- **Transaction pattern features**
  - Learned from the **Context-level Contrast** module

- **Statistical features**
  - Extract from **subgraph**

- **Classifiers**
  - XGBoost
Evaluation - Dataset

Data Collection

- As of March 2023, 5,639 phishing addresses.
- randomly select 25,000 active normal addresses
- Labeling – Etherscan
  - labeled nodes being the central nodes, Two-layer BFS
- 9,237,535 Ethereum addresses and 219,927,673 transaction records.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Total Nodes</th>
<th>#Labeled</th>
<th>#Edges</th>
<th>#Average Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_1$</td>
<td>30,000</td>
<td>106</td>
<td>24,965,770</td>
<td>832.2201</td>
</tr>
<tr>
<td>$D_2$</td>
<td>40,000</td>
<td>140</td>
<td>27,642,111</td>
<td>691.0701</td>
</tr>
<tr>
<td>$D_3$</td>
<td>50,000</td>
<td>166</td>
<td>31,597,197</td>
<td>631.9566</td>
</tr>
<tr>
<td>$D_4$</td>
<td>60,000</td>
<td>207</td>
<td>33,072,308</td>
<td>551.2143</td>
</tr>
<tr>
<td>$D_5$</td>
<td>70,000</td>
<td>238</td>
<td>34,450,265</td>
<td>492.1537</td>
</tr>
<tr>
<td>$D_6$</td>
<td>80,000</td>
<td>269</td>
<td>35,872,229</td>
<td>450.3111</td>
</tr>
<tr>
<td>$D_p$</td>
<td>9,237,535</td>
<td>5,639</td>
<td>219,927,673</td>
<td>23.8080</td>
</tr>
</tbody>
</table>

- Generate a large graph based on the transaction information crawled around all labeled phishing nodes, and select the largest connected component
- Sample with random walks to obtain subgraphs of different sizes
Evaluation - Baselines and Metrics

Baselines

**Feature-based** - Features only \(^1\) are 219-dimensional statistical features from the node’s 1-order and 2-order neighbors.

**Random walk-based** - DeepWalk\(^3\), Node2Vec\(^4\), and LINE\(^5\). both topological information and node attributes are involved.

**Deep learning-based** - SDNE\(^6\), E-GCN\(^7\), GraphSAGE\(^8\), TSGN\(^2\) and GAT\(^9\)

Metrics

- Recall
- Precision
- F1-score

## Evaluation - Conventional Comparison Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Training Data</th>
<th>$D_1$</th>
<th>$D_2$</th>
<th>$D_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>F1</td>
<td>Recall</td>
<td>Pre</td>
</tr>
<tr>
<td>Only Features</td>
<td>X</td>
<td>0.7713</td>
<td>0.7572</td>
<td>0.7859</td>
</tr>
<tr>
<td>DeepWalk</td>
<td>A</td>
<td>0.7108</td>
<td>0.7572</td>
<td>0.6697</td>
</tr>
<tr>
<td>Node2vec</td>
<td>A</td>
<td>0.7478</td>
<td>0.7624</td>
<td>0.7337</td>
</tr>
<tr>
<td>LINE</td>
<td>A</td>
<td>0.7990</td>
<td>0.8721</td>
<td>0.7373</td>
</tr>
<tr>
<td>SDNE</td>
<td>A</td>
<td>0.7447</td>
<td>0.6492</td>
<td>0.8732</td>
</tr>
<tr>
<td>GraphSAGE</td>
<td>X, A</td>
<td>0.8027</td>
<td>0.7709</td>
<td>0.8372</td>
</tr>
<tr>
<td>GAT</td>
<td>X, A, Y</td>
<td>0.8110</td>
<td>0.7749</td>
<td>0.8506</td>
</tr>
<tr>
<td>E-GCN</td>
<td>X, A, Y</td>
<td>0.8136</td>
<td>0.8796</td>
<td>0.7568</td>
</tr>
<tr>
<td>TSGN</td>
<td>X, A, Y</td>
<td>0.8174</td>
<td>0.7382</td>
<td>0.9156</td>
</tr>
</tbody>
</table>

### Conclusions
- **TGC outperforms** all the other compared methods by a significant margin, especially in large graphs.
- **TGC has better node representation capability** than existing Ethereum phishing detection methods.
- Network representation methods based on deep learning are **not performing well**.
Evaluation - Large-scale Data Comparison Results

<table>
<thead>
<tr>
<th>Method</th>
<th>$D_4$</th>
<th>$D_5$</th>
<th>$D_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only Features</td>
<td>0.7850</td>
<td>0.8010</td>
<td>0.7806</td>
</tr>
<tr>
<td>DeepWalk</td>
<td>0.7104</td>
<td>0.7111</td>
<td>0.7075</td>
</tr>
<tr>
<td>Node2vec</td>
<td>0.7577</td>
<td>0.7477</td>
<td>0.7534</td>
</tr>
<tr>
<td>LINE</td>
<td>0.7637</td>
<td>0.7842</td>
<td>0.7794</td>
</tr>
<tr>
<td>SDNE</td>
<td>0.7239</td>
<td>0.7273</td>
<td>0.7056</td>
</tr>
<tr>
<td>GraphSAGE</td>
<td>0.8105</td>
<td>0.7938</td>
<td>OOM</td>
</tr>
<tr>
<td>GAT</td>
<td>OOM</td>
<td>OOM</td>
<td>OOM</td>
</tr>
<tr>
<td>E-GCN</td>
<td>OOM</td>
<td>OOM</td>
<td>OOM</td>
</tr>
<tr>
<td>TSGN</td>
<td>0.8286</td>
<td>0.8595</td>
<td>0.8878</td>
</tr>
<tr>
<td>TGC</td>
<td><strong>0.9538</strong></td>
<td><strong>0.9600</strong></td>
<td><strong>0.9580</strong></td>
</tr>
</tbody>
</table>

Conclusions

- The TGC subgraph sampling training method can remain lightweight in large-scale network scenarios.
- TGC has better node representation capability and stable performance than other methods on large graphs.
Evaluation - Dynamic Data Comparison Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Total Nodes</th>
<th>#Labeled</th>
<th>#Edges</th>
<th>#Average Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>4,824,725</td>
<td>2,273</td>
<td>24,965,770</td>
<td>5.1745</td>
</tr>
<tr>
<td>Test</td>
<td>4,412,810</td>
<td>2,355</td>
<td>27,642,111</td>
<td>6.2640</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Retrain</th>
<th>Conn</th>
<th>F1</th>
<th>Recall</th>
<th>Pre</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only Features</td>
<td>✓</td>
<td>✓</td>
<td>0.7260</td>
<td>0.6333</td>
<td>0.8506</td>
</tr>
<tr>
<td>E-GCN</td>
<td>✓</td>
<td>✓</td>
<td>0.5009</td>
<td>0.4856</td>
<td>0.5172</td>
</tr>
<tr>
<td>TSGN</td>
<td>✓</td>
<td>✓</td>
<td>0.7663</td>
<td>0.7887</td>
<td>0.7451</td>
</tr>
<tr>
<td>TGC</td>
<td>✓</td>
<td>✓</td>
<td>0.9237</td>
<td>0.9291</td>
<td>0.9183</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Creation time</th>
<th>Number of phishing addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016</td>
<td>5</td>
</tr>
<tr>
<td>2017</td>
<td>340</td>
</tr>
<tr>
<td>2018</td>
<td>1928</td>
</tr>
<tr>
<td>2019</td>
<td>523</td>
</tr>
<tr>
<td>2020</td>
<td>1050</td>
</tr>
<tr>
<td>2021</td>
<td>218</td>
</tr>
<tr>
<td>2022</td>
<td>533</td>
</tr>
<tr>
<td>2023</td>
<td>31</td>
</tr>
</tbody>
</table>

Conclusions
- TGC can detect emerging addresses in real-world scenarios without model retraining, and has no requirement on the overall connectivity of the transaction network.
- Subgraph training combine contrastive learning scheme is able to help improve embedding quality.
Evaluation - Few-shot & Ablation Study & Sensitivity Analysis

Few-shot Results

<table>
<thead>
<tr>
<th>Method</th>
<th>100%</th>
<th>80%</th>
<th>50%</th>
<th>20%</th>
<th>10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only Features</td>
<td>0.7260</td>
<td>0.7154</td>
<td>0.7092</td>
<td>0.6870</td>
<td>0.6610</td>
</tr>
<tr>
<td>E-GCN</td>
<td>0.5009</td>
<td>0.5283</td>
<td>0.4091</td>
<td>0.2292</td>
<td>0.2087</td>
</tr>
<tr>
<td>TSGN</td>
<td>0.7663</td>
<td>0.7280</td>
<td>0.5800</td>
<td>0.4554</td>
<td>0.1473</td>
</tr>
<tr>
<td>TGC</td>
<td>0.9237</td>
<td>0.8983</td>
<td>0.8635</td>
<td>0.8750</td>
<td>0.8409</td>
</tr>
</tbody>
</table>

Observation Results

- Our proposed TGC method is least affected by the reduction of data size
- All modules in TGC are important
- TGC is robust to hyperparameter perturbation.

Ablation Study

Sensitivity Analysis
Conclusion

- We propose a **Transaction Graph Contrast Network (TGC)** to enhance phishing scams detection performance on Ethereum.

- TGC inputs **subgraphs** instead of the **entire graph** for training, which eases the model’s requirements for machine configuration and data connectivity, and can be well adapted to **dynamic networks**.

- Motivated by the **natural camouflage** and **sparsity distribution** of phishing addresses, we design **node-level contrast** and **context-level contrast** to learn the **unique properties** and **universal transaction patterns** of phishing addresses.

- We hope that our work demonstrates the **serious threat** of phishing scams on Ethereum and calls for **effective countermeasures** deployed by the blockchain community.
THANK YOU FOR LISTENING

Sijia Li
lisijia@iie.ac.cn