SCHOOL OF COMPUTER AND INFORMATION SCIENCES

Data Engineering Project Report

23MCMT01

Anti-Money Laundering in Bitcoin: Experimenting with Graph Convolutional Networks and federated learning on graphs

OBJECTIVE

The integration of blockchain and distributed ledger technologies in the financial sector, forming the Internet of Money, has raised regulatory concerns. The balance between user privacy and financial accountability is delicate, particularly in combatting money laundering and terrorism financing. This project explores the deployment of forensic techniques, includes transaction graph analysis, in tracking cryptocurrency transactions. Utilizing real-world Bitcoin data, the study reveals the effectiveness of Graph Convolutional Networks (GCN) in classifying and identifying illicit transactions. The research emphasizes the importance of public-private collaborations to develop transparent and effective forensic strategies for the evolving financial landscape.

Dataset In our work,

In our research, we conducted experiments utilizing the publicly available Elliptic transactions dataset, as provided in the context of Weber et al. (2019). The dataset, accessible on Kaggle Elliptic dataset, consists of real Bitcoin transactions represented as a directed graph network. Each transaction serves as a node, with directed edges indicating fund flows from source to destination addresses.

Dataset Overview:

- Nodes: 203,769 transaction nodes
- Edges: 234,355, representing fund flows
- Features: 167 per transaction (94 from the transaction itself, 73 from the graph network)
- Labels: Illicit (4,545), Licit (42,019), Unknown (157,205)
- Temporal Data: Grouped into 49 time steps (3-hour interval)

Graph Convolutional Network (GCN) Model for Transaction Classification

The primary goal of the GCN model is to learn a function that captures features on a graph-structured dataset. This involves processing a graph with nodes and edges, along with a feature description for each node. The model aims to understand the relationships among nodes and their neighborhoods by creating node embeddings in a latent vector space.

Key Concepts:

- The model utilizes a deep neural network structure, particularly GCNs, to analyze and classify transactions.
- Each node in the graph receives and aggregates features from its neighbors to compute its local state.
- The outcome is typically an output feature matrix at the node level, revealing characteristics of the node's neighborhood.

Implementation:

- GCNs are implemented using the Keras framework
- The graph convolution layer, a crucial component, comprises three main steps:

Pre-processing:

- Feed Forward Network is applied to node features, generating initial node representations.

Graph Convolutional Layers:

- Two graph convolutional layers are applied to the node representations, incorporating skip connections. This process produces node embeddings.

Post-processing:

- Another Feed Forward Network is applied to the node embeddings to generate the final node embeddings.

Prediction:

- Node embeddings are fed into a Softmax layer for predicting the node class.

Network Architecture:

- The entire model follows a sequential workflow, organized as follows:

  Apply pre-processing to generate initial node representations.

  Apply graph convolutional layers with skip connections to produce node embeddings.

  Apply post-processing to generate final node embeddings.

  Feed the node embeddings into a Softmax layer for predicting the node class.

Conclusion:

- The described GCN model is designed to effectively capture the intricacies of transaction data within a graph structure, providing a comprehensive approach to transaction classification. The use of GCNs and the specified network architecture contribute to the model's ability to discern patterns and relationships among nodes in the grap

CONTRIBUTION IN PROJECT

data preparation:

Merged features with classes.

Renamed class values to integers.

Swapped transaction identifiers for a sorted index.

Selected only the part of the dataset labeled licit or illicit.

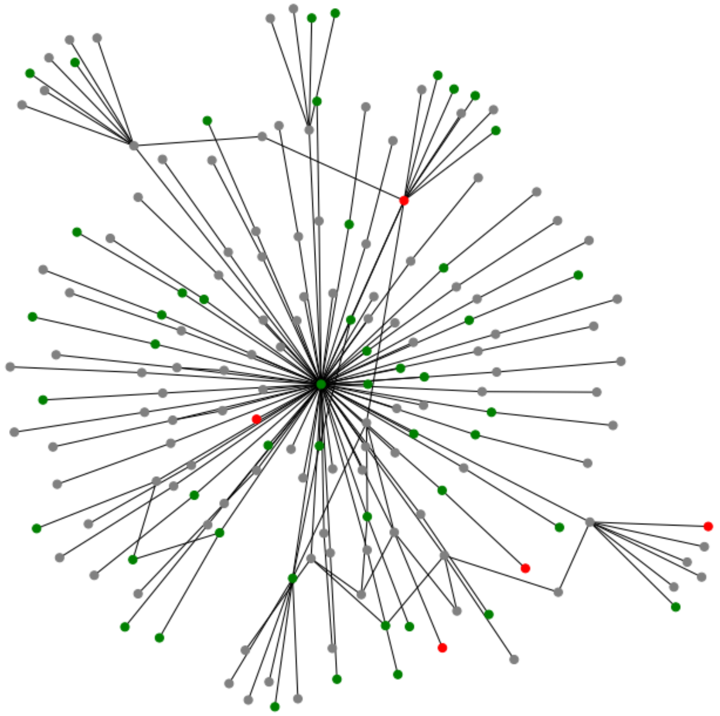
Removed all edges between unknown transactions.

After pre-processing, our cleaned dataset comprised 46,564 transactions and 36,624 edges.

red -> illicit

green -> licit

gray -> unclassified



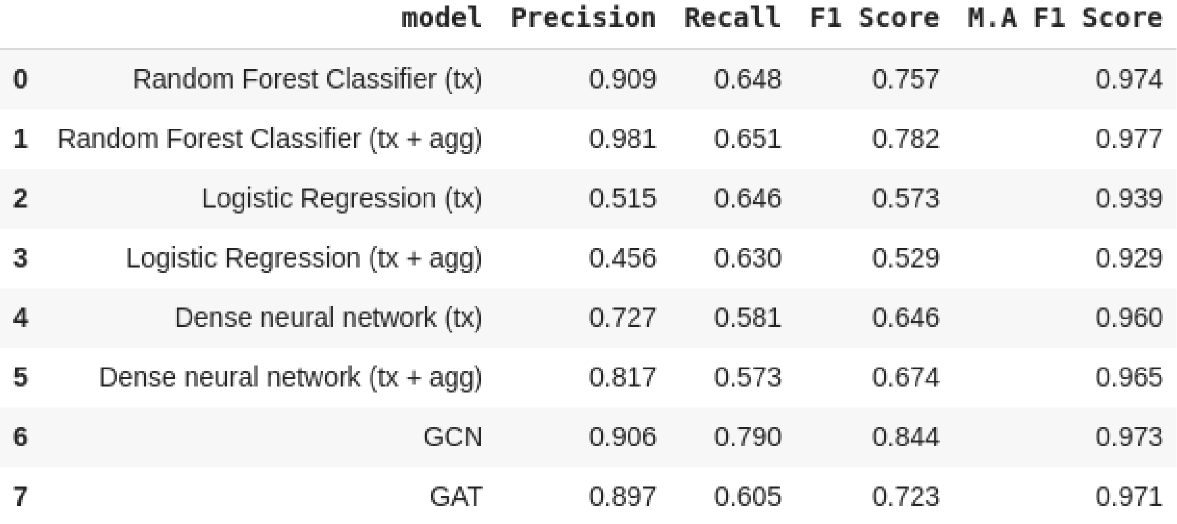
Compare with the other models

Comparison of the performance metrics for various machine learning models applied to the task . The models were evaluated based on precision, recall, F1 score, and the macro-average F1 score.

* Random Forest Classifier (IX)
* Random Forest Classifier (Ix + agg)
* Logistic Regression (IX)
* Logistic Regression (Ix + agg)
* MLP (IX)
* MLP (Ix + agg)
* KNeighborsClassifier (IX)
* KNeighborsClassifier (Ix + agg)
* SVC (IX)
* SVC (Ix + agg)

Results

Table showing the results for illicit transaction classification with the F1-score, Micro Average F1-score, Precision and Recall metrics for all models



	model	Precision	Recall	F1 Score	M.A F1 Score
0	Random Forest Classifier (tx)	0.909	0.648	0.757	0.974
1	Random Forest Classifier (tx + agg)	0.981	0.651	0.782	0.977
2	Logistic Regression (tx)	0.515	0.646	0.573	0.939
3	Logistic Regression (tx + agg)	0.456	0.630	0.529	0.929
4	Dense neural network (tx)	0.727	0.581	0.646	0.960
5	Dense neural network (tx + agg)	0.817	0.573	0.674	0.965
6	GCN	0.906	0.790	0.844	0.973
7	GAT	0.897	0.605	0.723	0.971