SCHOOL OF COMPUTER AND INFORMATION SCIENCES

Data Engineering Project Report

23MCMT01

OBJECTIVE

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

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)

Architecture

Graph Convolutional Network Model Architecture

Objective:

- Learn a function of signals/features on a graph-structured dataset.
- Input: Graph with nodes and edges + Feature description for each node.
- Key Idea: Nodes aggregate features from neighbors to compute local state.
- Output: Node-level feature matrix.

Graph Convolution Layer:

- Process input node representations using a Feed Forward Network to generate a message.
- Aggregate messages of neighbors using permutation invariant pooling (unsorted segment sum operation).
- Combine node representations and aggregated messages.
- Process the combined information to produce new state (node embeddings) via concatenation and Feed Forward Network.

Network Architecture:

Sequential workflow:

Apply pre-processing using Feed Forward Network on node features to generate initial node representations.
Apply two graph convolutional layers with skip connections to produce node embeddings.

Apply post-processing using Feed Forward Network on node embeddings to generate final node embeddings.
Feed the node embeddings into a Softmax layer for predicting the node class.

Implementation:

- Developed using the Keras framework.

Node Embeddings:

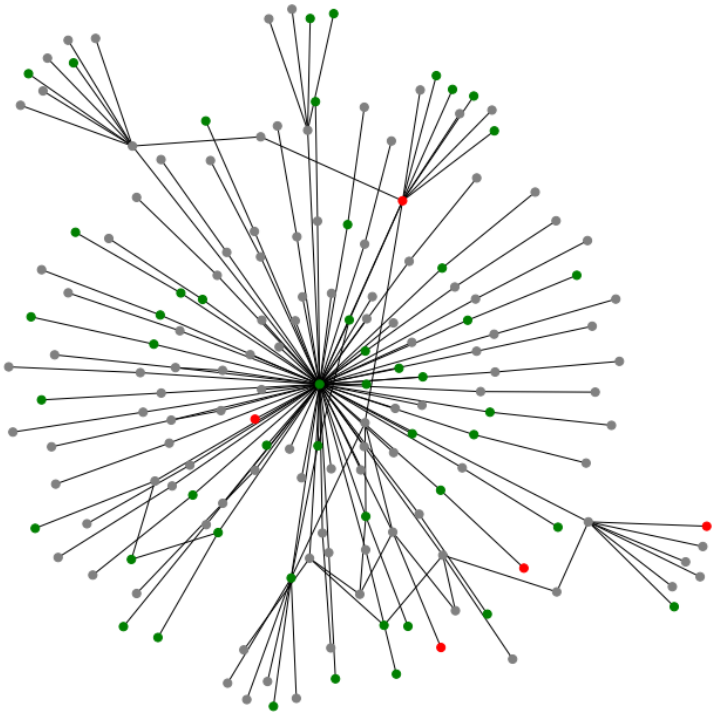
- Represents node characteristics in a latent vector space.
- Captures information about the node neighborhood in the graph.
- Implemented as a look-up table mapping nodes to vectors.

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.



Results

For the discussion of the 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