**Fraud Detection using GNN**

Project III

Group members:

* Phùng Hà Nguyên

**1. Abstract**

Fraud detection is an important subject in fields like finance or e-commerce to prevent significant losses. Traditional methods often fail to capture the complex structure of relationships between the sides involved in a fraudulent transaction. This calls for a new method of detecting fraud.

For that reason, in this project, Graph Neural Networks (GNNs) are applied, with nodes representing entities, such as users, cards, clients, and edges representing relationships, such as transactions. Using GNNs, we capture the graph structure and detect anomalies and fraudulent patterns.

**2. Introduction**

**2.1. Objectives**

This project aims to detect fraudulent transactions using Graph Neural Networks, while learning GNNs and knowledge graph along the way. By utilizing the relational structure of transaction data, the goal is to identify fraudulent activities and anomalies accurately and quickly.

**2.2. Approaches**

The project utilizes public datasets from Kaggle, such as paysim1 and IEEE-CIS. Data is processed and analyzed using Python, with key libraries including pandas for data manipulation and pytorch for creating, training and evaluating models.

**3. Datasets and Data Processing**

**3.1. Datasets**

For this project, we explored three datasets:

* *IEEE-CIS Fraud Detection*

This is a real-world dataset, with obfuscated data. While the data is generally useful, this dataset does not contain enough information about the recipient of a transaction to be useful for GNNs.

* *paysim1*

This is a synthetic dataset, with clear client identities and transactions. The data is simple, containing only transactions, with the name of the sender, the recipient, their balance before and after transactions, the amount sent and whether the transaction is a fraud or not.

* *Credit Card Transactions*

This is also a synthetic dataset, with more information about when and where the transaction happens. However, this dataset contains only transactions between buyers and merchants, without data of transfers and cashouts.

After consideration, I decided to use *paysim* as the dataset.

**3.2. Data Processing**

The dataset is already clean so there is not much data processing needed. For the purpose of using GNNs, the transactions need to be put into a graph:

A computer code with many colored text

Description automatically generated with medium confidence

**4. Model, Training and Results**

**4.1. Model**

For this project, we explored several different models:

* Simple GNN model with two graph convolution layersA computer screen shot of a program code

  Description automatically generated
* Using node degree as node featuresA screen shot of a computer code

  Description automatically generated

**4.2. Training**

Since the dataset is heavily imbalanced, I explored using weighted Cross Entropy Loss as loss function, with weight decided by the ratio of classes and using Focal Loss.

A computer screen shot of a program code

Description automatically generated

**4.3. Results**

Despite using weighted loss function, due to how heavily imbalanced the dataset is (only 0.13% of transactions are fraudulent), I couldn’t achieve a good result. The prediction has high recall, but low accuracy and very low precision.

**A screen shot of a computer code

Description automatically generated**

**5. Conclusion and Future Work**

**5.1. Conclusion**

Despite applying GNNs and techniques to reduce effects of imbalanced datasets, I still couldn’t overcome the difficulties. The model tends to predict all transactions as non-fraud or classify too many transactions as fraudulent, depending on the loss weight.

**5.2. Future Work**

Future research could focus more on overcome the effects of imbalanced datasets. A more throughout EDA on a more detailed dataset could also be used to improve the model.