Fraud detection in financial transactions using graph ML techniques.

Arsenii Belugin a.belugin@innopolis.university
ALyona Sinyagina a.sinyagina@innopolis.university
Dinara Murtazina d.murtazina@innopolis.university

1. Application Domain

Task: Fraud detection in financial transactions using graph ML techniques.

Fraud detection in financial transactions is a high-priority task for banks and financial institutions. Applying graph models to analyze transaction networks helps identify anomalous or suspicious operations, potentially improving security effectiveness and reducing risk.

2. Dataset

Dataset: *Elliptic Bitcoin Dataset*
Source: https://www.kaggle.com/datasets/ellipticco/elliptic-data-set/data

This dataset includes information on over 200,000 Bitcoin transactions, divided into two classes: fraudulent and normal. The data is structured as a transaction graph, where nodes represent transactions, and edges represent connections between them.

3. Dataset Description, Prediction Task, and Metric

Description: The dataset represents a directed transaction graph, where each transaction can be either normal or fraudulent. Each node (transaction) has 166 anonymous features, such as transaction time, amount, etc.

Prediction Task: Node classification. We will determine whether a transaction is fraudulent (1) or normal (0).

Metric: *F1-score*. For fraud detection, it’s essential to balance precision and recall, as F1-score accounts for both correct detections and the likelihood of missing fraudulent transactions.

4. Why Choose This Dataset?

The Elliptic Bitcoin dataset is ideal for showcasing the power of graph ML models since it’s a complex transaction network where nodes and relationships can be used to detect anomalous patterns. Additionally, the presence of labels allows us to evaluate the model and measure its performance.

5. Graph ML Technique

Technique: Graph Convolutional Networks (GCN) with neighbor aggregation. GCN is well-suited for this task because it leverages both structural and contextual information.

GCN extracts features from graphs by aggregating information from neighboring nodes, helping models better identify anomalies in transactions.

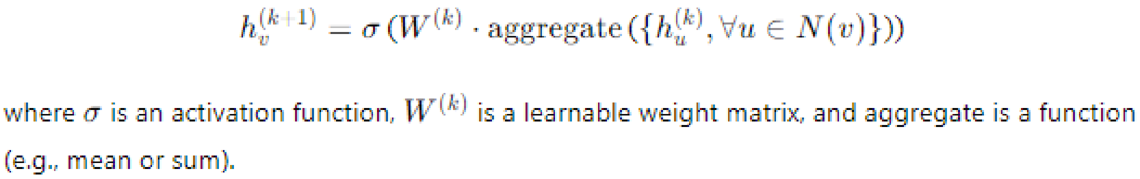
6. Model

Model Selection: *GraphSAGE* (SAmple and aggreGatE)

Description: GraphSAGE (Graph Sample and Aggregate) is a method designed to learn efficiently on large graphs by sampling and aggregating information from neighboring nodes rather than traversing the entire graph. Unlike traditional GCNs, it supports distributed computation and adapts to new nodes that may be added to the graph over time. This model can focus on only the relevant parts of the graph, which is especially useful for dynamic networks like transaction networks.

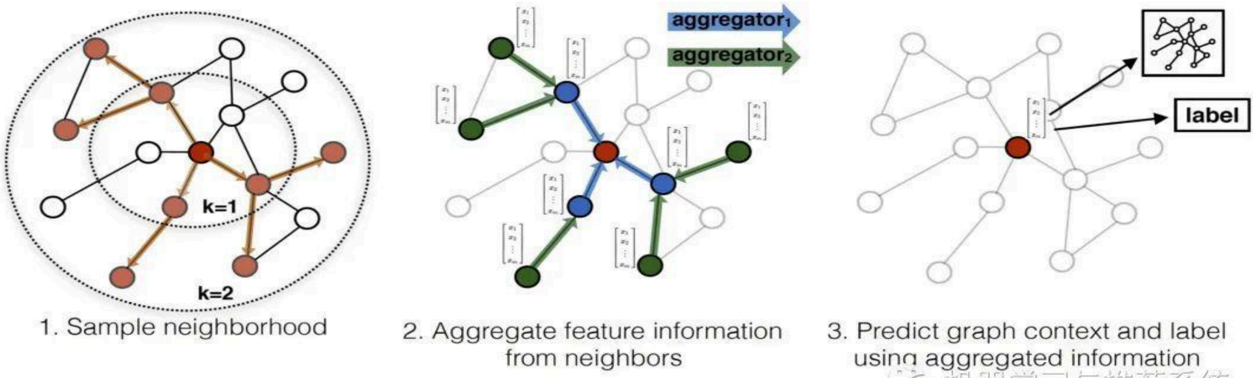
Architecture and Equations:

- Each layer aggregates information from a randomly selected set of a node’s neighbors.
- For each node **v**, represented as a feature vector **h_v**, its representation is updated using sampled neighbors **N(v)**.
- The update for layer **k** is given by:



$$h_v^{(k+1)} = \sigma(W^{(k)} \cdot \text{aggregate}(\{h_u^{(k)}, \forall u \in N(v)\}))$$

where σ is an activation function, $W^{(k)}$ is a learnable weight matrix, and aggregate is a function (e.g., mean or sum).



1. Sample neighborhood

2. Aggregate feature information from neighbors

3. Predict graph context and label using aggregated information

Figure 1: Visual illustration of the GraphSAGE sample and aggregate approach.

7. Why This Model Fits the Dataset

GraphSAGE is suitable because it handles large graphs efficiently, working well with sparse connections typical of transaction graphs. This model gathers information while considering both graph structure and node features, which helps to identify anomalous connections and behaviors in the transaction network. Moreover, it adapts to new nodes, which is essential for detecting emerging fraudulent patterns.