FRAUD DETECTION IN E-COMMERCE AND FINANCIAL TRANSACTIONS USING GRAPH NEURAL NETWORKS

By

Karunakar Thammadi

MSDS Practicum 1

PROBLEM STATEMENT

- Traditional fraud detection systems rely on isolated transaction analysis, missing relational patterns
- Rule-based approaches demonstrate limited efficacy (0.19% recall on benchmark datasets)
- Fraudulent actors exploit network structures through complex fund movement chains
- Existing machine learning models fail to capture directional transactional dependencies

THE $4.5 TRILLION PROBLEM

- Global fraud losses exceed $4.5 trillion annually
- Traditional rule-based systems catch only obvious patterns
- Real example: PaySim dataset rule-based system caught 16 out of 8,213 fraud cases (0.19% success rate)
- Fraudsters exploit network blind spots moving money through complex chains
- Key insight: Fraud is a network problem, not a transaction problem

WHY GRAPHS MATTER ?

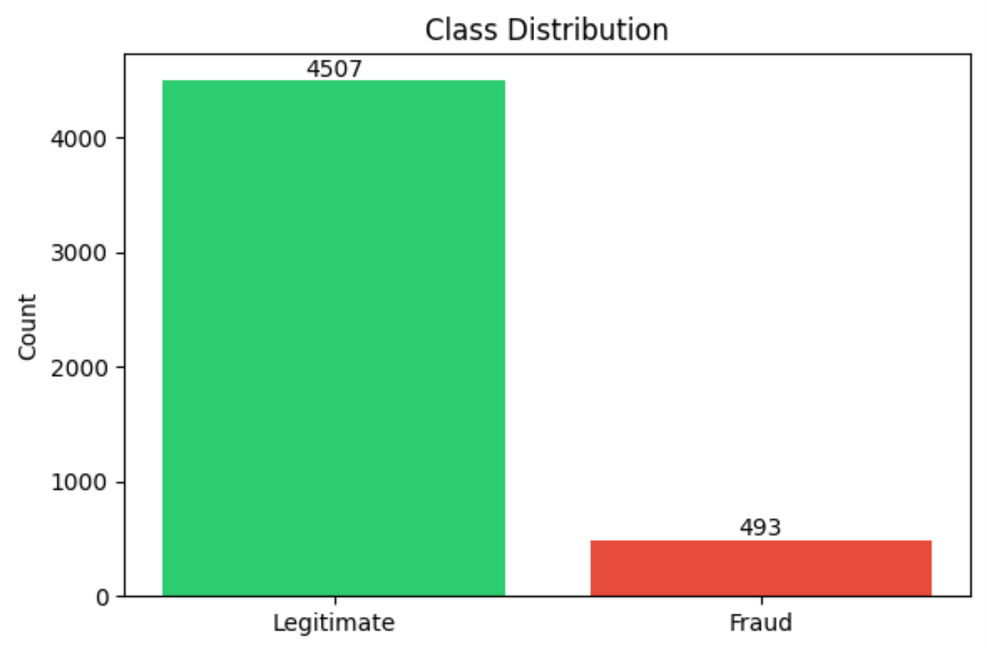
- Traditional approach: Look at each transaction in isolation
- Graph approach: Map relationships between accounts
- Analogy: Like tracking disease spread through contact tracing vs. checking symptoms only
- Fraudulent accounts leave footprints in their connection patterns

DATA

- PaySim Mobile Money Fraud Dataset: Synthetic mobile payment transactions from an agent-based simulator.
- Elliptic Bitcoin Transaction Dataset: Real cryptocurrency transaction network labeled for anti-money laundering.
- Synthetic Dataset : A simulated transaction graph mimicking real-world fraud behavior with directional patterns and low homophily.

CLASS IMBALANCE

- Only 10% of accounts are fraud while 90% are legitimate, creating a strong class imbalance.
- Models may favor predicting the majority (legitimate) class, reducing their ability to correctly detect fraud.



THE HOMOPHILY PUZZLE

Homophily = "Similar nodes connect to similar nodes."

Elliptic:

- High homophily (0.95)
- Fraud connects to fraud

Synthetic Banking:

- Low homophily (0.31)
- Fraud connects to legitimate accounts to hide

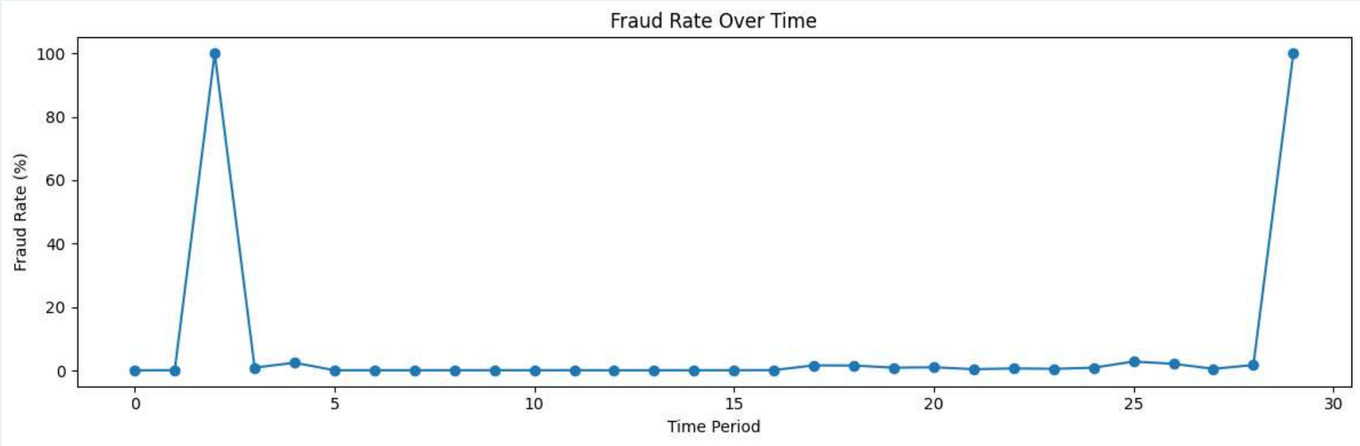
DATA INSIGHTS

PaySim Mobile Money:

- Fraud concentrates inTRANSFER and CASH_OUT transactions
- Average fraud amount: $1.47M vs. $178K legitimate (8× higher)
- Most accounts transact only once—extremely sparse network

Elliptic Bitcoin:

- Clear time patterns—fraud clusters in specific time steps
- 77% of transactions unlabeled (realistic investigation scenario)
- Strong clustering—illicit transactions form visible chains



FROM RAW DATA TO GRAPH

Elliptic Dataset

- **Graph Structure:**
  - Nodes = Bitcoin transactions
  - Edges = Transaction flow between transactions (directed)

Synthetic Dataset

- **Graph Structure:**
  - Nodes = Simulated accounts
  - Edges = Generated transactions

PaySim Dataset

- **Graph Structure:**
  - Nodes = Accounts
  - Edges = Transactions between accounts (sender → receiver)

BUILDING THE BASELINE

Logistic Regression — High recall, very low precision.

MLP — Similar performance with limited precision improvement.

Graph Linear Model — Slight precision improvement.

Basic GCN — Noticeable performance gain using graph information.

Precision	Fraud Recall	F1 Score	
18.9%	85.5%	0.309	Logistic Regression
21.7%	75.5%	0.337	Standard Neural Net (MLP)
16.1%	64.5%	0.257	Graph linear model
13.5%	18.9%	0.158	Basic GCN

GRAPH NEURAL NETWORKS

- GNNs allow us to look at an account's entire neighborhood simultaneously.
- Instead of looking at 1 isolated transaction, we analyze an account's 1st and 2nd-degree connections.

THE INITIAL GRAPH STRUGGLE

- The first attempt was a "Basic GCN" that treated incoming and outgoing money exactly the same.
- Simply knowing two accounts are connected actually hurt performance if ignored the direction of funds.

DIRECTION MATTERS

- Built a custom Directed GCN to respect the physical flow of the money.
- Separating incoming and outgoing aggregations immediately fixed the graph and boosted performance.

Architecture	Precision	Recall	AUC	F1 Score
Logistic Regression	0.189	0.855	0.874	0.310
MLP	0.217	0.755	0.859	0.338
Basic GCN	0.136	0.189	0.586	0.158
Directed GCN	0.244	0.540	0.812	0.158

THE POWER OF STRUCTURAL FEATURES

What was added:

- In-degree / out-degree counts
- Flow imbalance (receives more than sends?)
- 2-hop fraud neighbor ratio (friends of friends who are fraudulent)

Impact on Directed GCN (Elliptic):

- Base: 0.241 F1
- With structural features: 0.296 F1 (+23%)

ROBUSTNESS TEST–SURVIVING DATA LOSS

We removed

- 5%
- 10%
- 20%

Of legitimate nodes to test imbalance sensitivity.

Measured:

- Flow asymmetry
- Fraud homophily
- F1(v1 vs v2)

DIRECTED MODEL STABILITY AND NOISE TESTING

Tested corruption types:

- Edge removal
- Direction flipping
- Label noise
- Feature masking

Corruption ratios:

- 5%
- 10%

HYPERPARAMETER TUNING

Search space:

- Hidden Dim:32,64,128
- Learning Rate:0.01,0.005,0.001
- Weight Decay:0,1e-4,5e-4

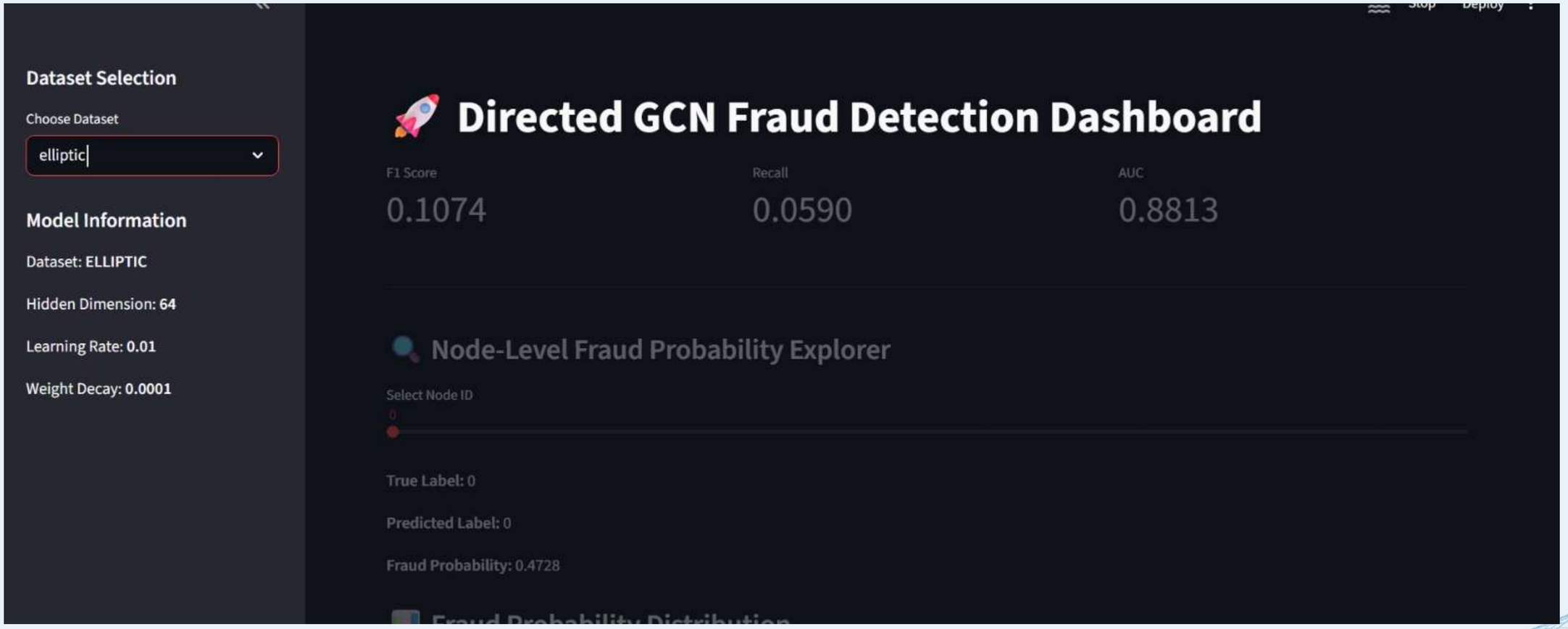
27 combinations per dataset

Early stopping enabled

FINAL MODEL RANKING

Dataset	Model	Recall	Auc	F1
Elliptic	DirectedGCN v3	0.473684	0.839040	0.4286
Elliptic	v1	0.532779	0.806534	0.3113
Elliptic	v2	0.572484	0.784967	0.2697
PaySim	DirectedGCN v3	0.173463	0.881312	0.2468
PaySim	v2	0.044431	0.796429	0.0828
PaySim	v1	0.043214	0.633083	0.0809

DASHBOARD


Dataset Selection
Choose Dataset
elliptic
Model Information
Dataset: ELLIPTIC
Hidden Dimension: 64
Learning Rate: 0.01
Weight Decay: 0.0001
Directed GCN Fraud Detection Dashboard
F1 Score
0.1074
Recall
0.0590
AUC
0.8813
Node-Level Fraud Probability Explorer
Select Node ID
0
True Label: 0
Predicted Label: 0
Fraud Probability: 0.4728

CONCLUSION & NEXT STEPS

- Directed Graph Neural Networks provide a massive, proven leap forward for financial security.
- They capture complex, coordinated attacks while remaining mathematically resilient to missing data.
- Next Step: Deploying this directed graph architecture into real-time, streaming transaction pipelines.

THANK YOU