UNMASKING FINANCIAL FRAUD

Comparing the Performance of Graph Neural Networks and Traditional Models Approaches

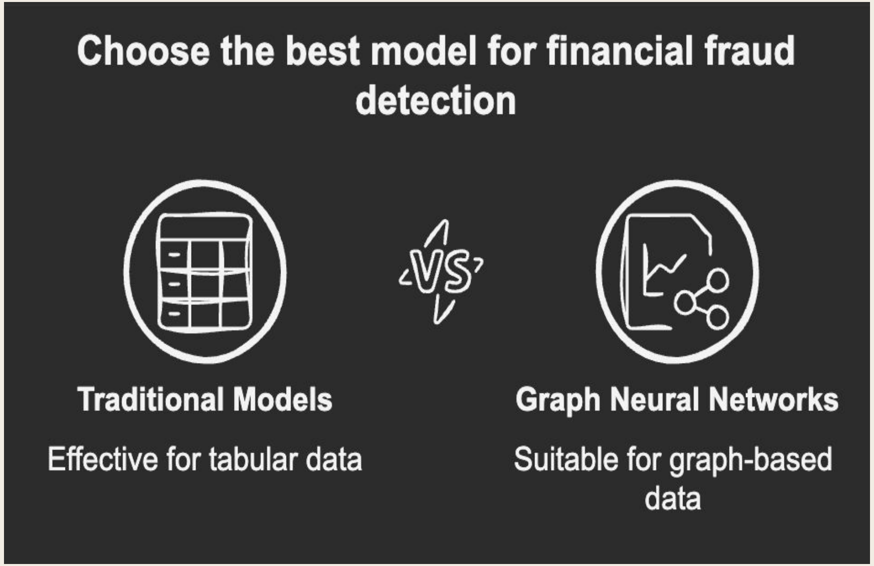
By: Kanishk Goel

AGENDA

INTRODUCTION

- Financial fraud is a growing issue, leading to billions of dollars in losses annually.
- Credit card fraud is the most common form of identity theft

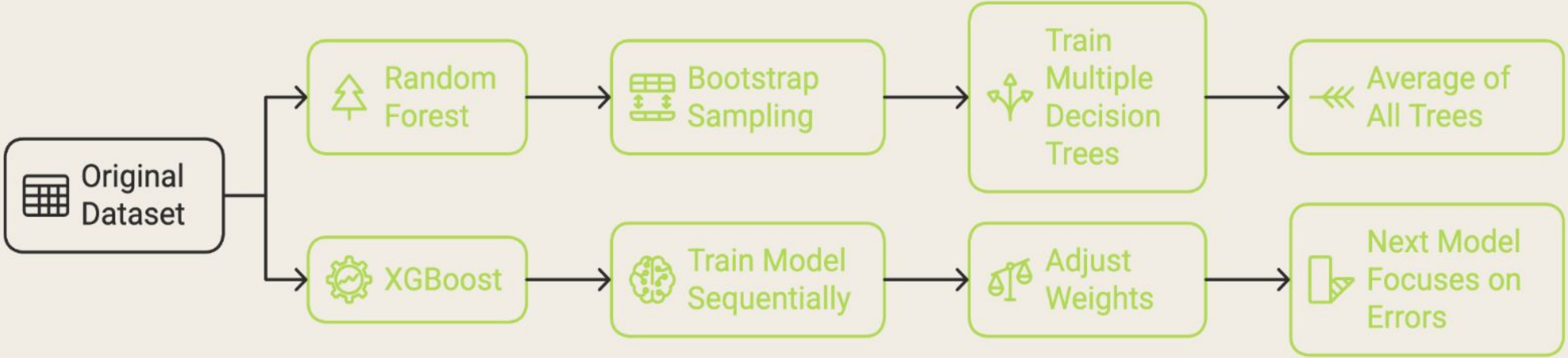
- **Impact of Fraud**:
  - Erodes consumer trust.
  - Increases business operational costs.
  - Causes financial and legal challenges.
  - Affects economic stability.



CHALLENGES

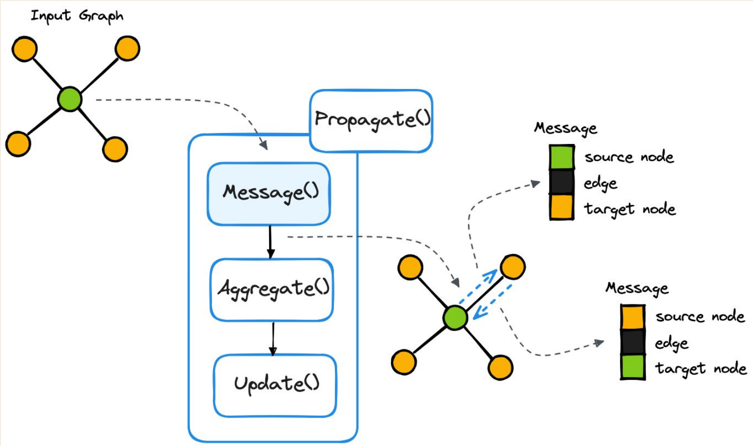
- Fraudsters are adaptive and change patterns
- Hard to get real-world data
- High imbalance in dataset
- Multiple accounts or cards being used
- The evolving complexity of fraud patterns.
- The critical role of model explainability

TRADITIONAL ML MODELS



Why Graph Neural Networks

- Transactions form a graph network
- GNNs learn from node embeddings and edge interactions, capturing hidden fraud patterns
- Traditional ML models treat transactions as independent instances, losing relational insights
- GNNs are capable of processing vast volumes of data

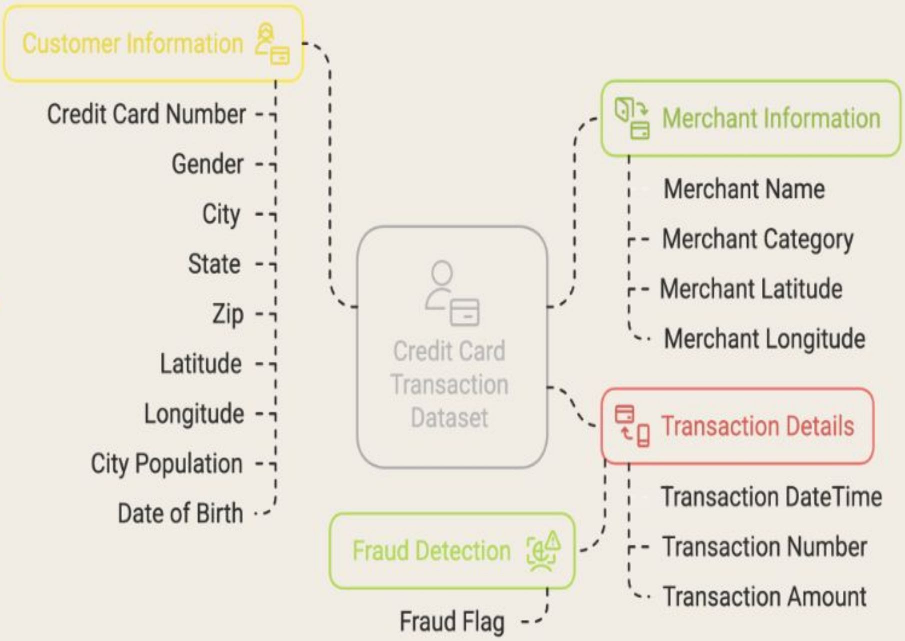


Types of Graph

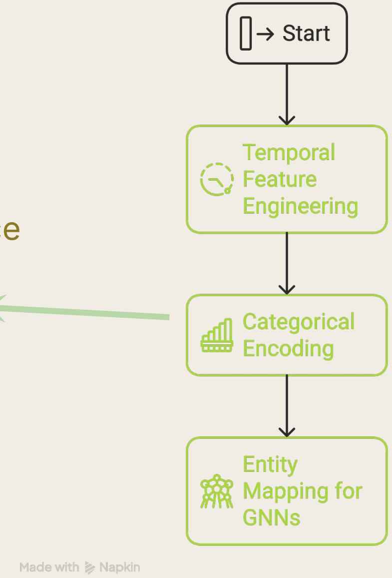
- **Homo-genous graph**:
  - A homogeneous graph consists of only **one type of node** and one type of edge.
  - Simpler to train but does not capture different entity roles in the graph.
  - Eg. **Social Network Analysis**: Nodes represent users, edges represent friendships.
- **Hetero-genous graph**:
  - A heterogeneous graph consists of **multiple types of nodes** and edges, capturing more complex relationships.
  - More complex but captures richer relationships.
  - Implemented by HeteroGCN
  - Eg. **Financial Fraud Detection**: Nodes represent Customers, Merchants, Transactions.

DATASET

- Simulated credit card transaction dataset containing transactions from the duration 1st Jan 2019 - 31st Dec 2020.
- Covers credit cards of 1000 customers doing transactions with a pool of 800 merchants.
- Generated using Sparkov Data — Github tool created by Brandon Harris.



DATA PREPROCESSING



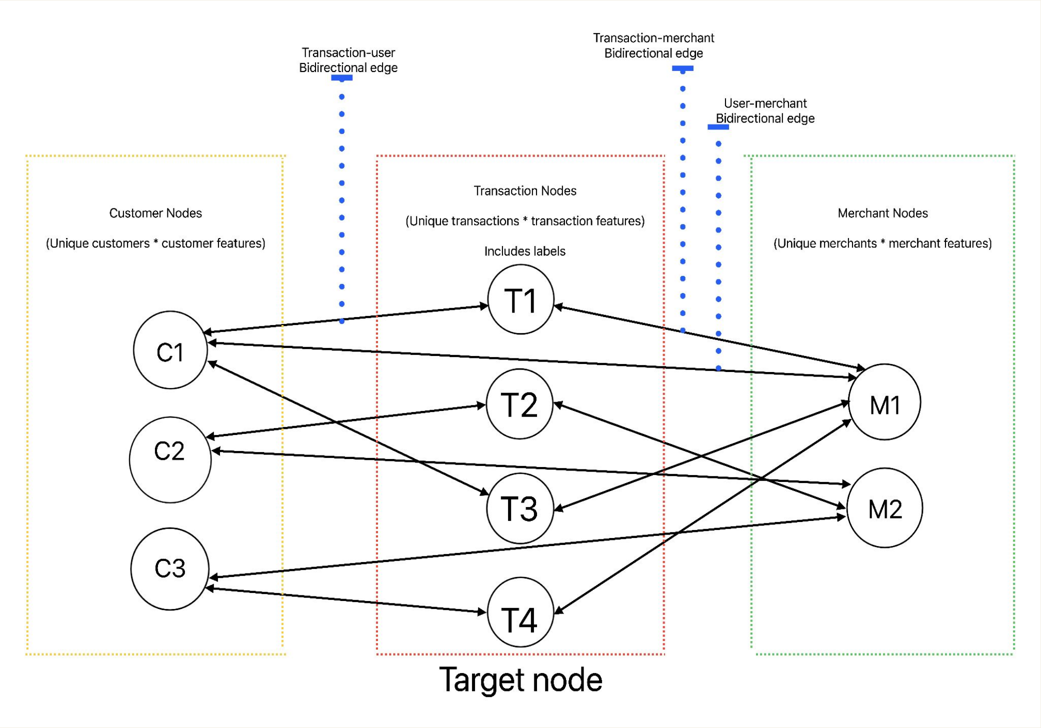
- Converted timestamp to datetime format.
- Extracted hour and day for temporal analysis.
- Applied sinusoidal encoding to preserve cyclic patterns

- One-Hot Encoding: for low-cardinality features (e.g., device type).
- Label Encoding: for ordinal relationships (e.g., city size).
- Target Encoding: for high-cardinality features like merchant category

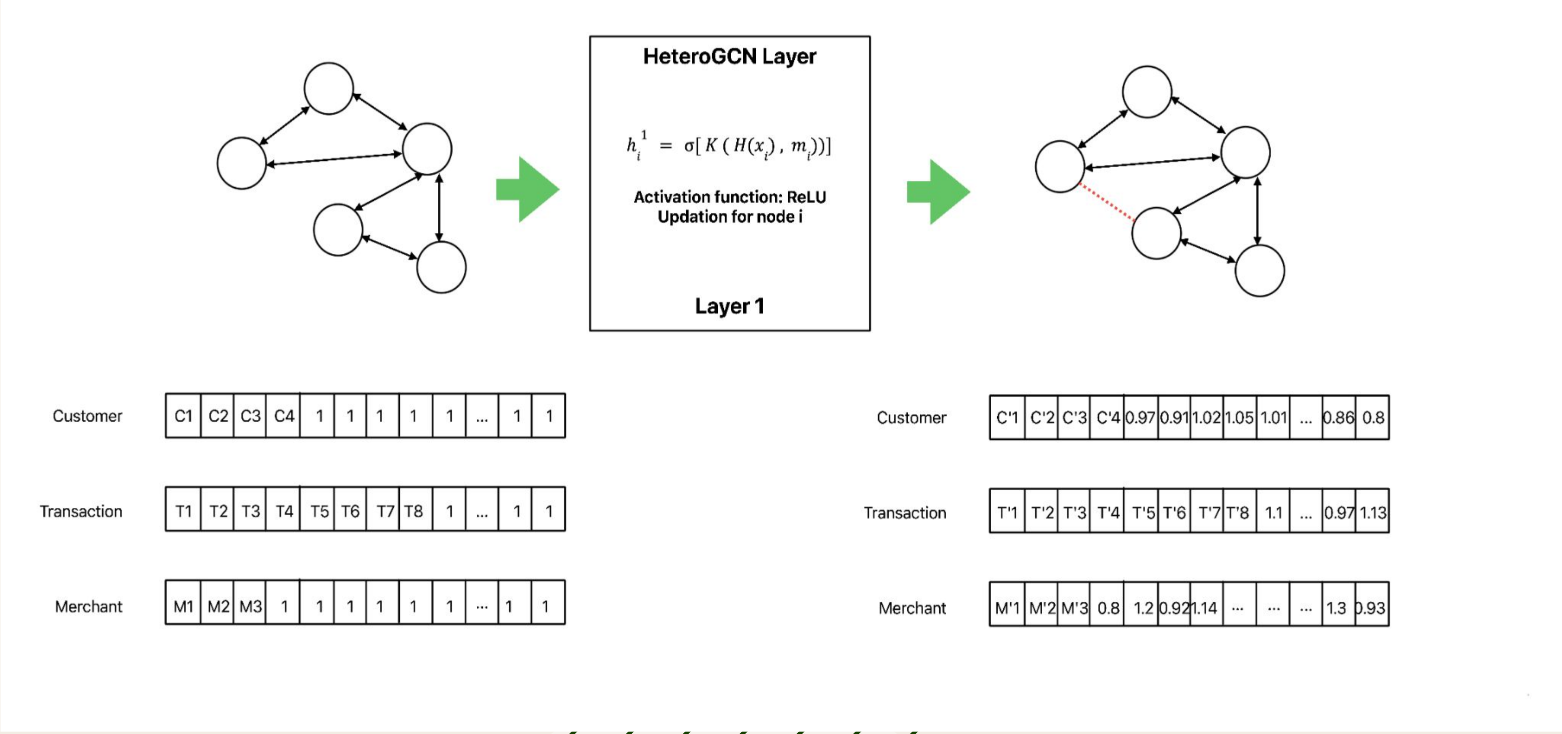
- Essential for creating Heterogeneous Graph using PyTorch Geometric

GRAPH STRUCTURE

- Transaction nodes: 1296675
- Merchant nodes: 693
- Customer nodes: 983
- Edge type:
  - customer ⟷ merchant: 479072
  - transaction ⟷ customer: 1296675
  - transaction ⟷ merchant: 1296675
  - transaction ⟷ transaction: 1296675

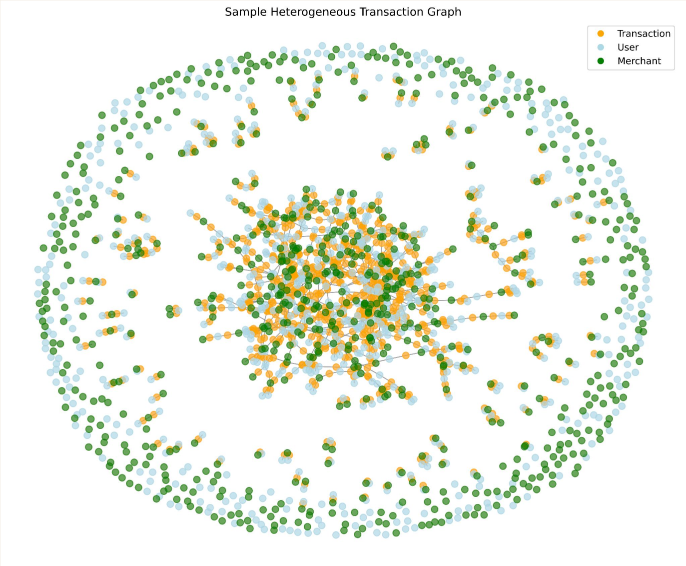


GNN Architecture



EXPERIMENTAL SETUP

- Used 4 HeteroConv layers with GraphSAGE
- Embedding size of 32
- AWS EC2 - G5.2xlarge with 32 gb memory and GPU Accelerated due to millions of records
- Batch size of 128 yield best results with 60 epochs
- Used early-stopping with patience = 10 is used to avoid overfitting



EVALUATION METRICS

- **AUC-PR**: measures the model's ability to correctly identify the positive class (fraud) across all threshold.
- **F1-score**: balances the trade-off between precision and recall, and is especially useful when both false positives and false negatives have significant consequences

RESULTS

Methods	Precision	Recall	F1-Score	AUCPR
XGBoost	0.89	0.52	0.66	0.76
Random Forest	0.93	0.52	0.67	0.76
Decision Tree	0.60	0.68	0.64	0.41
Logistic Regression	0.00	0.00	0.00	0.13
GNN	0.73	0.65	**0.68**	0.47

Methods	**Precision**	**Recall**	**F1-Score**	**AUCPR**
XGBoost	0.92	0.75	**0.83**	0.92
Random Forest	0.93	0.72	0.81	0.90
Decision Tree	0.66	0.80	0.72	0.57
GNN	0.90	0.75	**0.82**	0.73

CONCLUSION

- GNNs outperform classical models when relational structures dominate and feature engineering is limited.
- Heterogeneous message passing (users, merchants, transactions) boosts fraud detection performance.
- In Dataset 2, classical models (XGBoost, Random Forest) slightly outperformed due to strong, explicit features.
- GNNs excel in relationally complex, feature-sparse environments.
- Classical models remain strong when features are rich and well-crafted.

FEEDBACK!