Detecting Financial Anomalies

GNNs vs. Classical ML vs. Deep Learning

By: Chirag Lakhanpal and Sunisha Harish

Agenda

01 | Introduction.

Background & Purpose.

- Financial institutions process millions of transactions daily
- Anomalous transactions may indicate fraud, money laundering, or other illicit activities
- Timely and accurate detection of anomalies is crucial for financial security
- Compare the performance of Graph Neural Networks (GNNs), Classical Machine Learning, and Deep Learning models for anomaly detection in financial transactions
- Identify the strengths and weaknesses of each approach

Challenges & Objectives.

- Imbalanced datasets with a low proportion of anomalous transactions
- Continuously evolving fraud patterns and techniques
- Need for real-time or near-real-time detection
- Handling large-scale, high-dimensional, and heterogeneous financial data
- Evaluate the effectiveness of GNNs, Classical ML, and Deep Learning models for anomaly detection
- Provide insights and recommendations for selecting the most suitable approach based on the study's findings
- Contribute to the development of robust and efficient anomaly detection systems in the financial industry

02 Dataset and Preprocessing.

Dataset Description

- The Tabformer dataset, provided by IBM on their GitHub repository is a synthetic dataset that closely approximates a real-world financial fraud-detection dataset (IBM, 2021). The dataset consists of 24 million unique transactions, involving 6,139 unique cards and 100,343 unique merchants. Among these transactions, 29,757 are labeled as fraudulent, accounting for 0.1% of the total transactions.

- The highly imbalanced nature of the Tabformer dataset, with a significantly lower proportion of fraudulent transactions compared to legitimate ones, is a common characteristic of real-world fraud detection scenarios. This imbalance poses challenges for anomaly detection models, as they need to effectively identify the rare fraudulent instances while minimizing false positives

User	Card	Year	Month	Day	Time	Amount	Use Chip	Merchant Name	Merchant City	Merchant State	Zip	MCC	Errors?	Is Fraud?
0	0	2002	9	1	6:21	$134.09	Swipe Transaction	3527213246127870000	La Verne	CA	91750	5300		No
0	0	2002	9	1	6:42	$38.48	Swipe Transaction	-727612092139916000	Monterey Park	CA	91754	5411		No
0	0	2002	9	2	6:22	$120.34	Swipe Transaction	-727612092139916000	Monterey Park	CA	91754	5411		No
0	0	2002	9	2	17:45	$128.95	Swipe Transaction	3414527459579100000	Monterey Park	CA	91754	5651		No
0	0	2002	9	3	6:23	$104.71	Swipe Transaction	5817218446178730000	La Verne	CA	91750	5912		No

Data Preprocessing

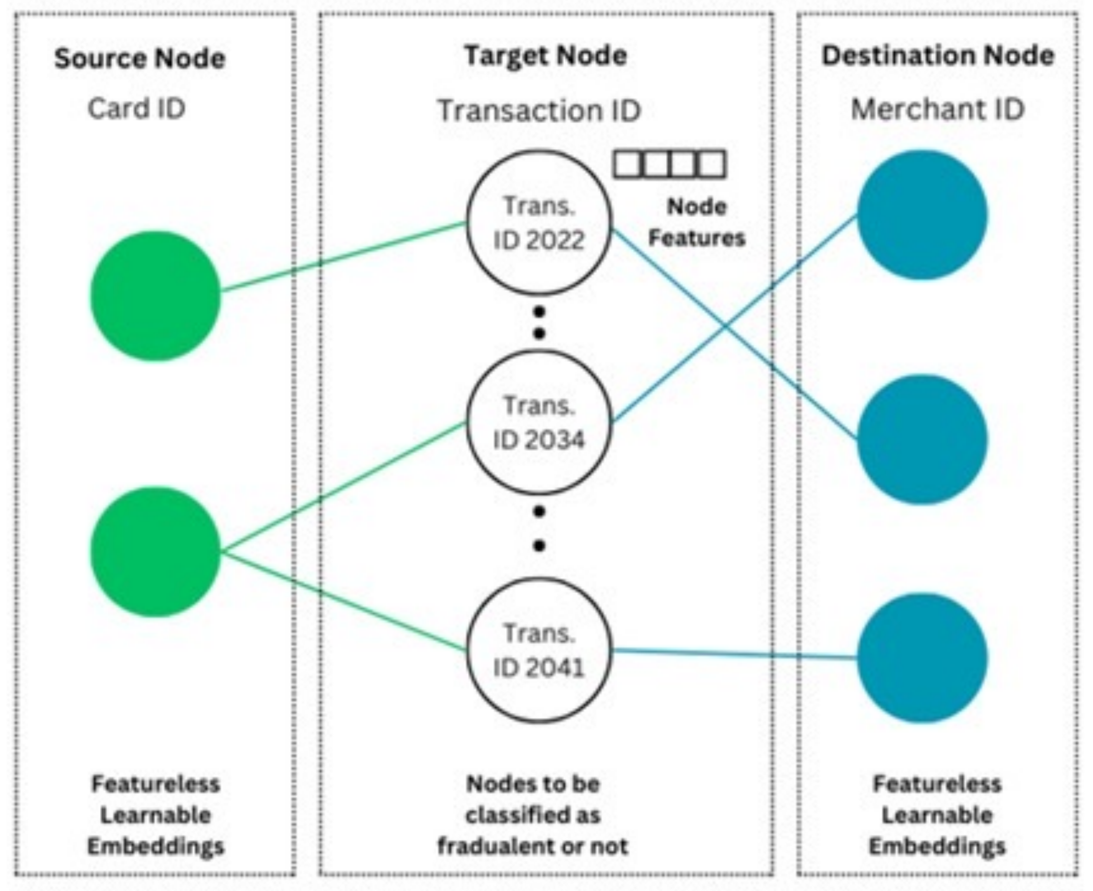
- The Tabformer dataset undergoes several preprocessing steps to ensure data quality and prepare it for the comparative study on anomaly detection models. The preprocessing pipeline is implemented using a custom function called `preprocess_data`, which takes the dataset as input along with various parameters to control the preprocessing steps.
- Firstly, the function automatically detects columns with only two unique values and encodes them as binary (0 and 1). This step helps in representing binary features effectively. Next, the columns are converted to numeric data type when possible to ensure consistent data representation across the dataset.
- To handle categorical variables, the non-numeric columns are one-hot encoded, converting them into binary vectors. This encoding technique allows the models to process categorical data effectively. Missing values in the dataset are handled by replacing them with the mode (most frequent value) of each column, ensuring that no data points are lost due to missing information.
- To reduce the dataset's size while maintaining the class distribution, stratified sampling is applied based on the 'Is Fraud?' column. This step helps in creating a representative sample of both fraudulent and non-fraudulent transactions. The 'Time' column, which contains datetime information, is treated separately, and appropriate datetime features are extracted from it.
- Data cleaning is performed on specific columns to ensure data consistency. The 'Amount' column undergoes cleaning to remove any special characters or inconsistencies. Additionally, columns that may not be relevant for anomaly detection, such as 'Card', 'User', and 'Merchant Name', are removed from the dataset.
- Certain columns, including 'Use Chip', 'Merchant City', 'Merchant State', 'Zip', 'MCC', and 'Errors?', are considered as categorical variables and are encoded accordingly. The 'Is Fraud?' column is designated as the target variable for the anomaly detection task.

User	Card	Year	Month	Day	Amount	Merchant_Name	Merchant_City	Merchant_State	Zip	MCC	Errors	Is_Fraud	Time_hour	Time_minute	Use_Chip_Chip Transaction	Use_Chip_Online Transaction	Use_Chip_Swipe Transaction
338	2	2009	4	28	28.22	1456298770479470000.000000	0.000029	0.000000	0.000029	0.000000	0.00096632	0	18	25	FALSE	FALSE	TRUE
881	0	2008	10	22	46.62	-6161792371494720000.000000	0.000057	0.000846	0.007066	0.000018	0.00096632	0	11	34	FALSE	FALSE	TRUE
271	4	2011	3	30	109.73	7824413107036070000.000000	0.000000	0.001063	0.000006	0.002401	0.00096632	0	4	17	FALSE	FALSE	TRUE
4	0	2002	6	10	91.15	-4317138273541960000.000000	0.000000	0.000355	0.000711	0.000000	0.00096632	0	15	18	FALSE	FALSE	TRUE
1934	1	2019	1	21	63.58	1459645998148460000.000000	0.000015	0.000000	0.000015	0.000000	0.00096632	0	21	43	TRUE	FALSE	FALSE

03

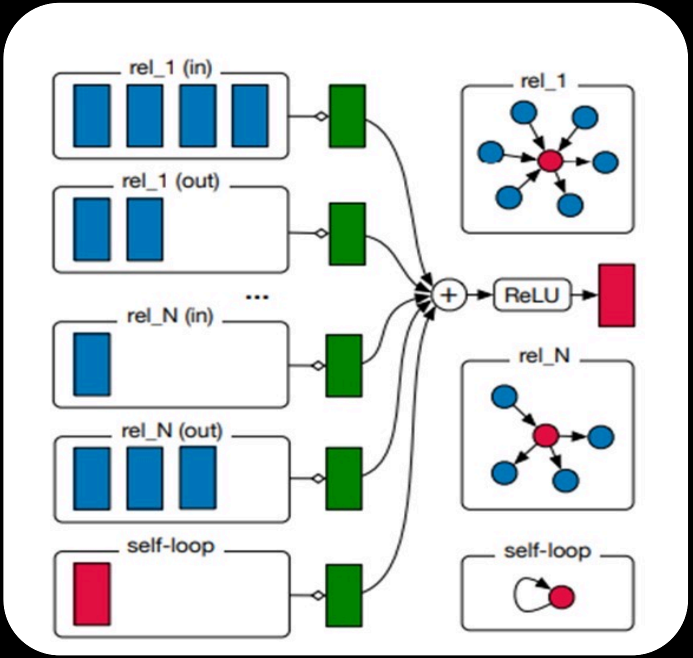
Graph Neural Networks.

Node Architecture



Relational Graph Convolutional Network (RGCN)

Architecture



Message Passing

$$h_i^{(l+1)} = \sigma\left(\sum_{r\in R} \sum_{j\in N_r(i)} W_r^{(l)} h_j^{(l)}\right)$$

Other Architecture Explored

1. Card and merchant nodes with transaction edges
   - Captured relationships between cards and merchants
   - Limited utilization of transaction features

2. Transaction nodes with logical proposition edges
   - Emphasized connections between transactions
   - Did not explicitly capture card-merchant relationships

3. Incorporating edge features
   - Enriched representation of transactional relationships
   - Introduced computational complexity

4. Featureless edges
   - Focused on structural information
   - Limited ability to capture fine-grained transaction characteristic

1. Node embeddings
   - Represented cards and merchants in a low-dimensional space ie making them feature less.
   - Captured inherent properties and behaviors

04

Classical Machine Learning Models.

Traditional Machine Learning Models Used.

Overview	Classical machine learning models are widely used for anomaly detection in financial transactions Models used in this study: XGBoost, CatBoost, Logistic Regression, LightGBM, and Random Forest
XGBoost (Extreme Gradient Boosting)	Ensemble learning algorithm that combines multiple weak learners (decision trees) to create a strong learner Employs gradient boosting to iteratively minimize the loss function and improve performance Handles missing values and supports parallel processing for faster training
CatBoost (Categorical Boosting)	Gradient boosting algorithm that effectively handles categorical features without the need for extensive preprocessing Uses ordered boosting and symmetric trees to reduce overfitting and improve generalization Provides robust performance and fast training times

Logistic Regression	Probabilistic model that estimates the likelihood of an event (anomaly) occurring based on input features Learns a linear decision boundary to separate anomalous and normal transactions Simple and interpretable model, but may struggle with complex, nonlinear relationships
LightGBM (Light Gradient Boosting Machine)	Gradient boosting framework that uses a leaf-wise tree growth strategy for faster training and lower memory usage Employs histogram-based algorithms to handle large-scale data efficiently Supports parallel and GPU learning for accelerated training
Random Forest	Ensemble learning method that constructs multiple decision trees independently and aggregates their predictions Each tree is trained on a random subset of features and samples, promoting diversity and reducing overfitting Provides robust performance, handles high-dimensional data, and estimates feature importance

05 Deep Learning Models.

Deep Learning Models Used.

- CNN: Convolutional Neural Networks, widely used for image classification tasks, leveraging convolutional layers to automatically extract hierarchical features from input data and achieve superior performance in visual recognition tasks.
- LSTM: Long Short-Term Memory networks, a type of recurrent neural network (RNN) equipped with memory cells capable of capturing long-range dependencies in sequential data, commonly applied in natural language processing and time series forecasting tasks.
- CNN-LSTM: A hybrid neural network architecture combining convolutional and recurrent layers, adept at processing both spatial and temporal information, often utilized in tasks involving sequential data with spatial dependencies, such as video analysis and sensor data processing.

06 Experimental Setup.

Evaluation Metrics

- Due to the highly imbalanced nature of the dataset, the following metrics are used to assess model performance:
    - F1 Score: Harmonic mean of precision and recall, providing a balanced measure of model accuracy
    - Recall: Measures the proportion of actual anomalies correctly identified by the model
    - Precision: Indicates the proportion of predicted anomalies that are actually true anomalies
    - AUC-PR (Area Under the Precision-Recall Curve): Summarizes the trade-off between precision and recall at different classification thresholds, suitable for imbalanced datasets

Machine Specifications

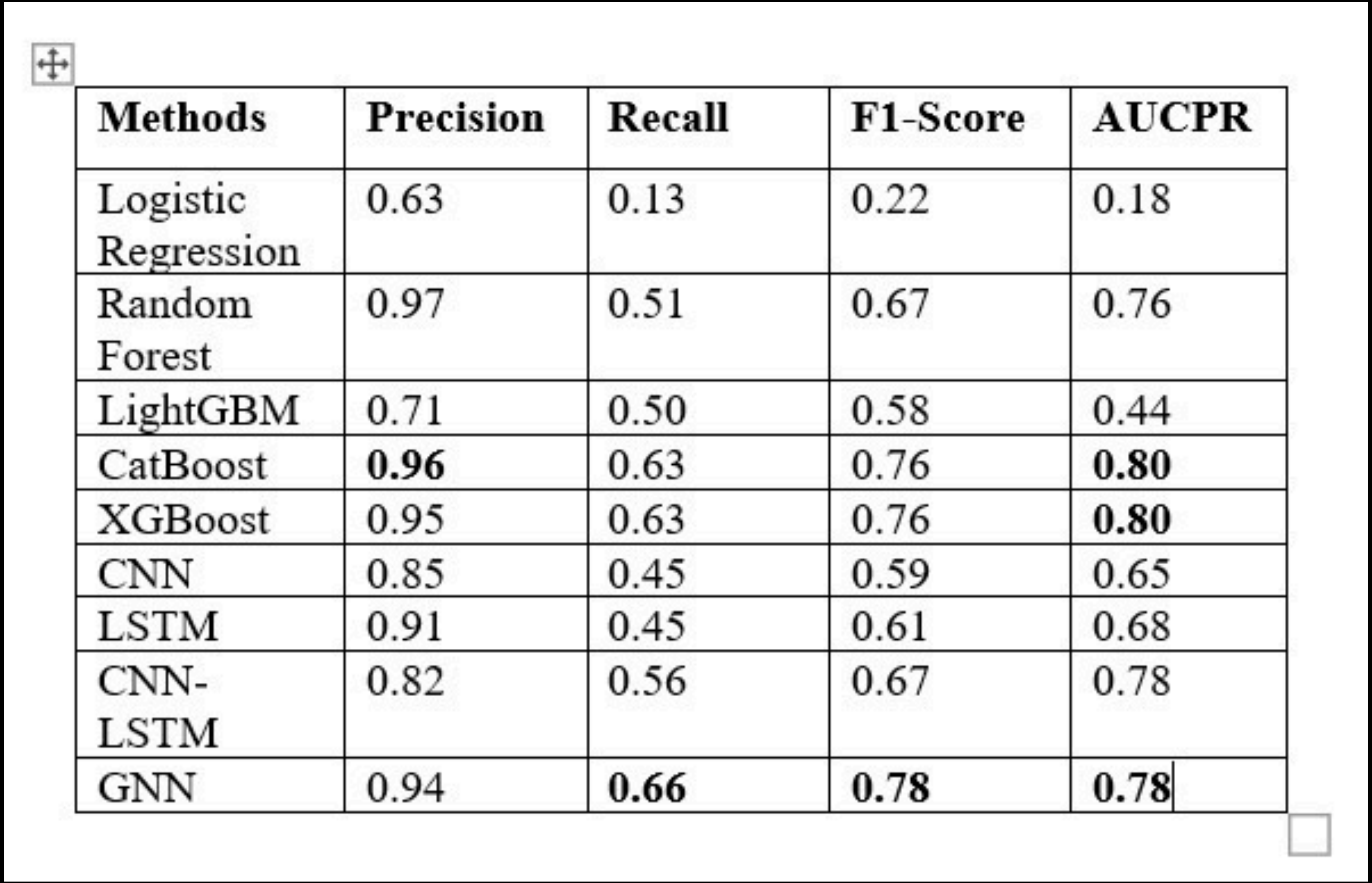
- Amazon EC2 g5.2xlarge instance
    - GPU: 1 x NVIDIA A10G Tensor Core GPU
    - GPU Memory: 24 GB GDDR6
    - vCPUs: 8
    - Memory: 32 GB
    - Storage: 50 GB
- GPU specifications enable faster training and inference times for deep learning models

Train-Test Split Strategy

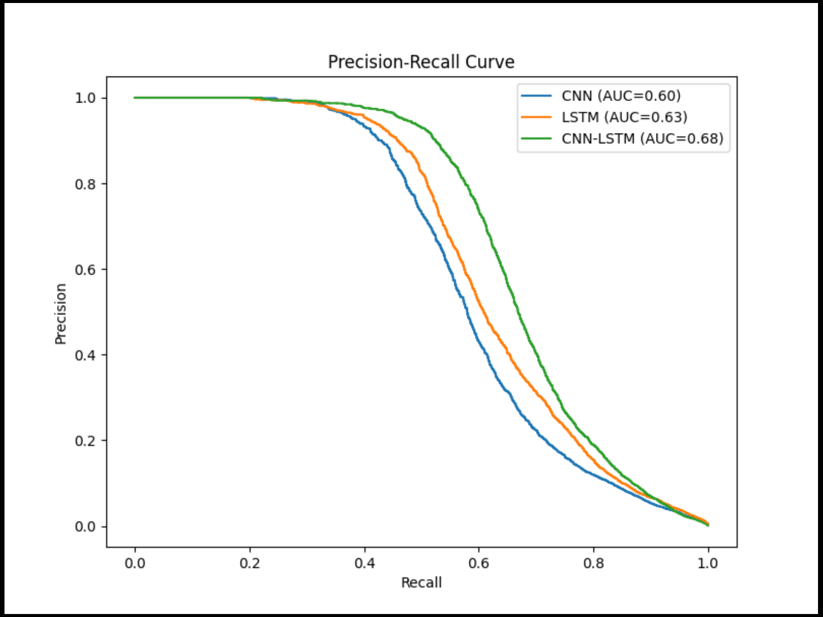
- Stratified sampling is applied using the 'is_fraud' label to ensure that the class distribution is preserved in both the training and testing sets
- This strategy helps maintain the imbalanced nature of the dataset during model training and evaluation

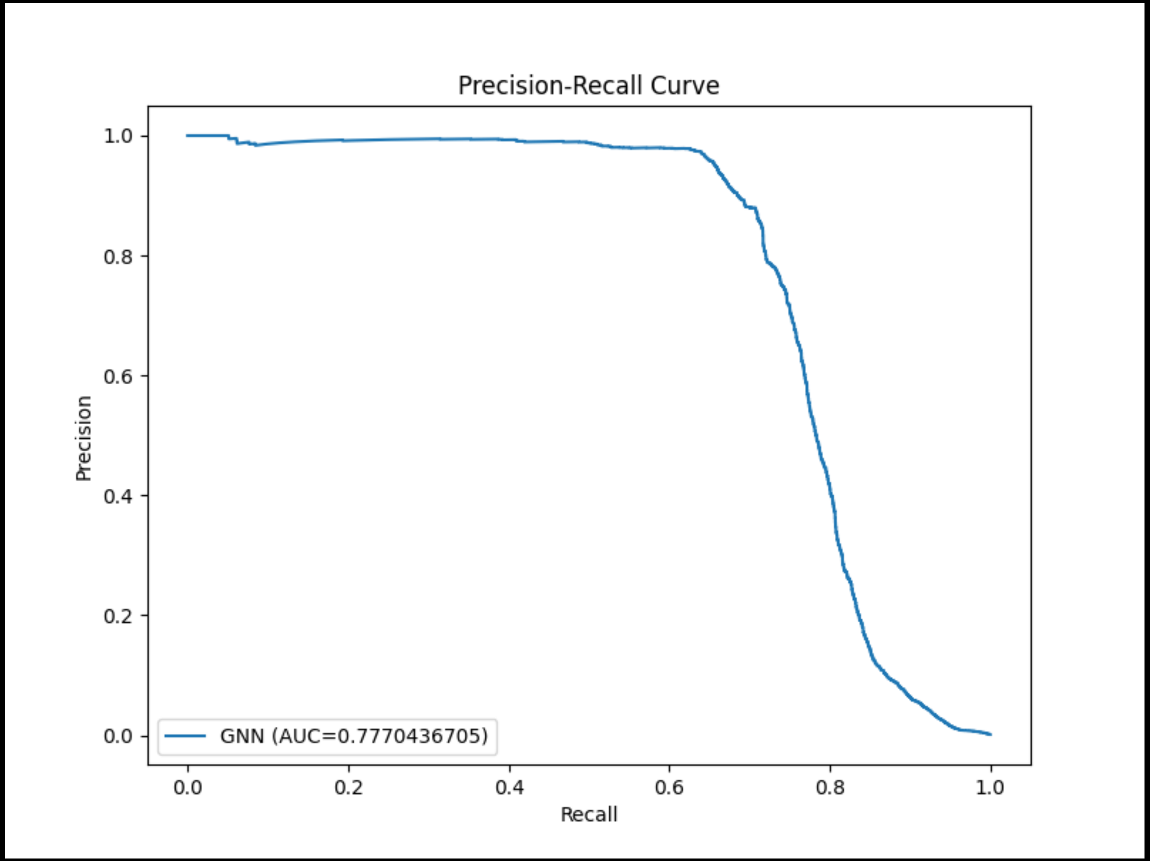
07 Results and Discussion.

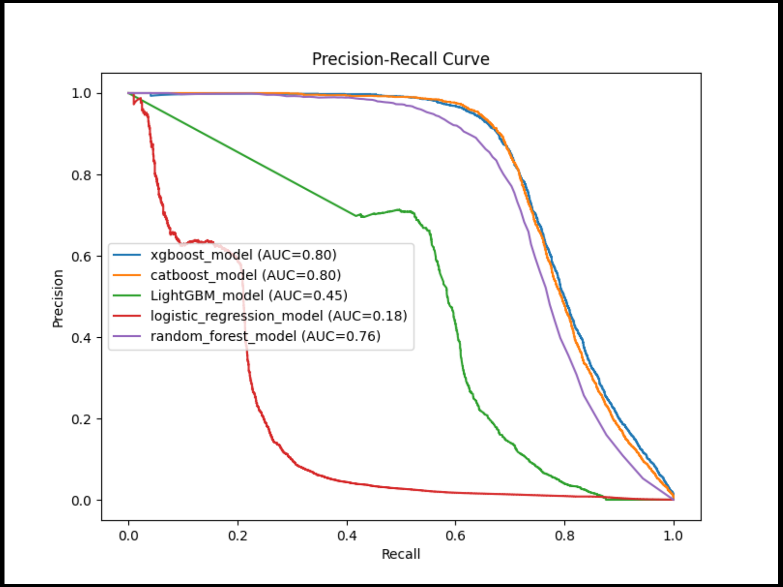
Results.



Methods	**Precision**	**Recall**	**F1-Score**	**AUCPR**
Logistic Regression	0.63	0.13	0.22	0.18
Random Forest	0.97	0.51	0.67	0.76
LightGBM	0.71	0.50	0.58	0.44
CatBoost	**0.96**	0.63	0.76	**0.80**
XGBoost	0.95	0.63	0.76	**0.80**
CNN	0.85	0.45	0.59	0.65
LSTM	0.91	0.45	0.61	0.68
CNN-LSTM	0.82	0.56	0.67	0.78
GNN	0.94	**0.66**	**0.78**	**0.78**


Precision-Recall Curve
CNN (AUC=0.60)
LSTM (AUC=0.63)
CNN-LSTM (AUC=0.68)
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0


Precision-Recall Curve
GNN (AUC=0.7770436705)
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0


Precision-Recall Curve
xgboost_model (AUC=0.80)
catboost_model (AUC=0.80)
LightGBM_model (AUC=0.45)
logistic_regression_model (AUC=0.18)
random_forest_model (AUC=0.76)
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0

08 Conclusion.

Key Findings

- Our proposed Graph Neural Network (GNN) model outperformed all other models, achieving an impressive F1 score of 0.78 for anomaly detection in financial transactions.
- The heterogeneous graph framework introduced in this study addresses the limitations of existing graph representations in transactional networks by placing transaction nodes at the center and incorporating learnable parameters for card and merchant ID nodes.
- The effectiveness of GNNs in this context highlights their potential for various applications, including fraud detection, recommendation systems, and beyond.

Future Research Directions

- Incorporating sampling strategies and edge weights into the heterogeneous graph framework presents a promising avenue for future work, offering opportunities to improve scalability, efficiency, and predictive performance in transactional network analysis and related domains.
- Integrating attention mechanisms into the heterogeneous graph can enhance the model's ability to capture complex relationships by allowing nodes to focus on relevant neighbors during message passing.