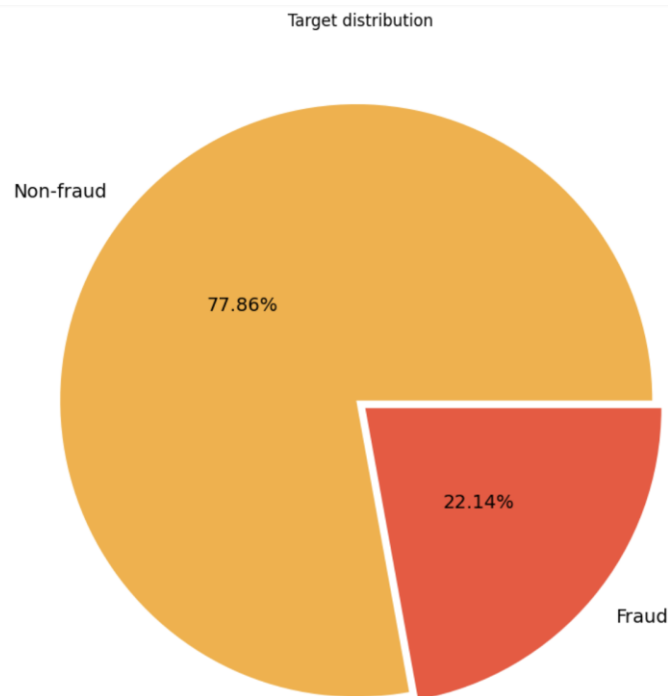


AI powered fraud detection in blockchain(Ethereum Transactions) using Machine Learning and GNN

An in-depth analysis of fraud detection
using traditional ML models and Graph
Neural Networks (GNN)

Dataset Explanation

- This dataset contains known fraud and valid transactions on Ethereum.
- Key Features:
 - - Address: Ethereum account
 - - FLAG: Fraud or Not
 - - Transaction Statistics: Sent/Received counts, Min/Max/Average values
 - - ERC20 Token Transactions
 - - Total 9841 non-null rows
- Data Distribution:



Data Preprocessing

- - Handled missing values
- - Normalized numerical features
- - Applied feature scaling
- - Addressed class imbalance with resampling techniques(SMOTE)
- - Engineered new features for better model performance

Workflow

- 1. Feature Extraction
- 2. Handling Imbalance
- 3. Model Training and Evaluation
- 4. Models Evaluated:
 - - Logistic Regression (LR)
 - - Random Forest (RF)
 - - XGBoost (XGB)
 - - Graph Neural Network (GNN)

Model Performance Comparison

- Comparison of models based on Precision, Recall, and F1-score:

- 1. Logistic Regression:

- - Accuracy: 89%
- - Recall: 89 (fraud)
- - F1-score: 77

- 2. Random Forest:

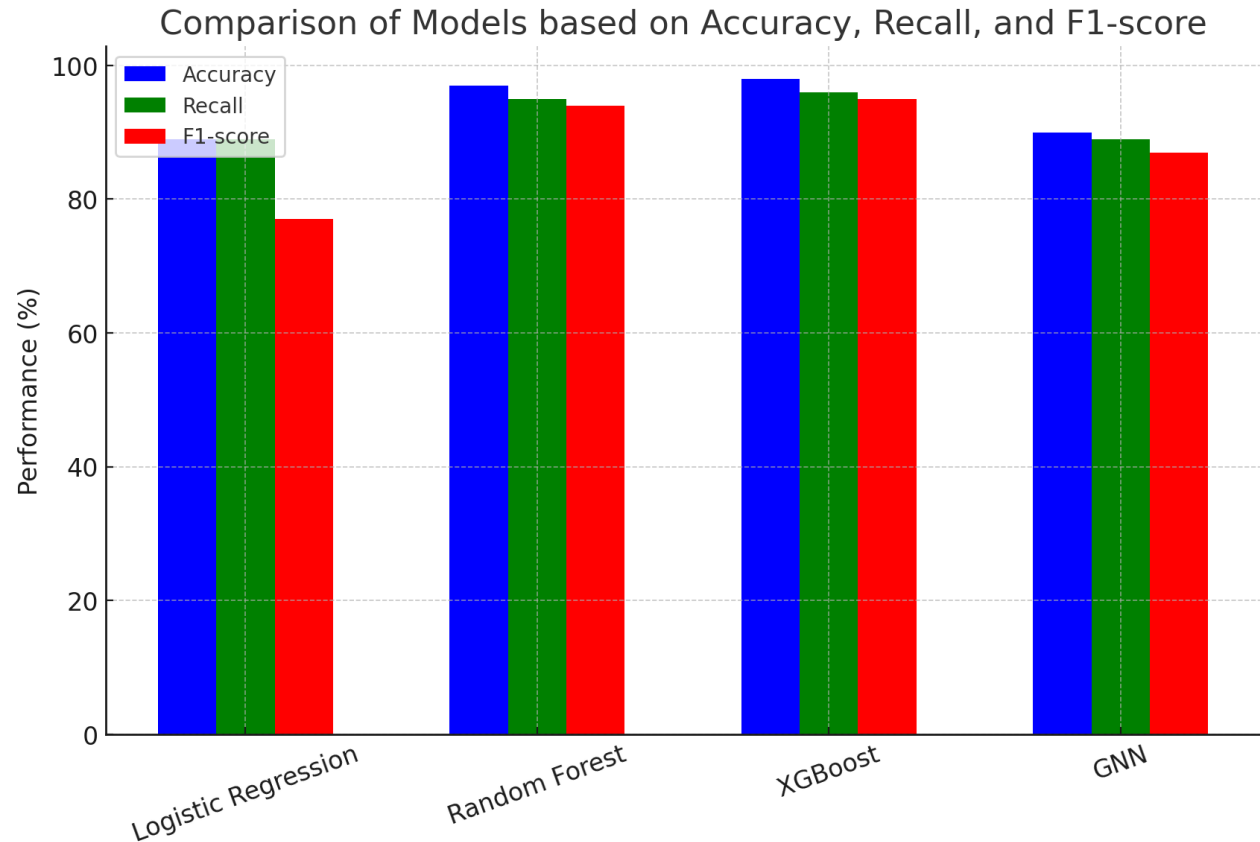
- - Accuracy: 97%
- - Recall: 95
- - F1-score: 94

- 3. XGBoost:

- - Accuracy: 98%
- - Recall: 96
- - F1-score: 95

- 4. GNN :

- - Accuracy: 90%
- - Recall: 89
- - F1-score: 87

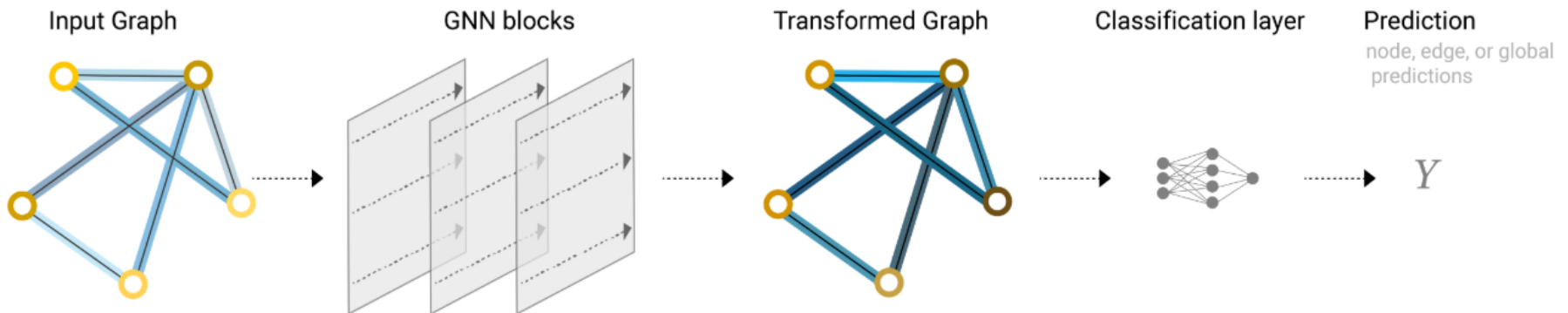


Understanding Model Performance (1)

- - Logistic Regression struggles with non-linear relationships
- - Random Forest improves performance but still misclassifies some fraud cases
- - XGBoost optimizes results by reducing both false positives and false negatives
- - GNN captures complex transaction patterns more effectively

Why GNN is Better?

- - Captures relationships between transactions using graph structures
- - Detects subtle fraud patterns that traditional models miss
- - Higher recall ensures fewer fraud cases go undetected
- - Excels in handling dynamic and evolving fraud schemes



An end-to-end prediction task with a GNN model.

Project Details

- Colab Link:
<https://colab.research.google.com/drive/1sDabxu78CUvKluZd8erylovGzlh06qpm?usp=sharing>
- Technologies Used:
 - - Python, Scikit-Learn, XGBoost, NetworkX, PyTorch Geometric
 - - Data Processing: Pandas, NumPy
- Code Structure:
 - - Data Loading and Preprocessing
 - - Model Training and Evaluation
 - - Proposed GNN Model Implementation

Conclusion and Future Work

- - XGBoost and GNN outperform traditional models in fraud detection
- - GNN is the best choice due to its ability to model transaction networks
- Future Scope:
 - - Improve GNN architecture for even better recall
 - - Use real-time fraud detection methods
 - - Extend to other cryptocurrencies and fraud types