NORTHWESTERN UNIVERSITY

SCHOOL OF PROFESSIONAL STUDIES

FRAUD DETECTION IN FINANCIAL TRANSACTIONS: A COMPREHENSIVE APPROACH USING MACHINE LEARNING AND GRAPH NEURAL NETWORKS

A4: Final Report

MS DSP 458 – Artificial Intelligence & Deep Learning

Group 5

Abhigna Mallepally, Anishka Agarwal, Edwin Daniels, Nagasudhan N, Sachin Sharma

December 07, 2024

Abstract

This project addresses the challenge of detecting fraudulent credit card transactions using a large dataset of 24.4 million transactions from IBM's financial database. Due to the severe class imbalance, with only 0.1% of transactions labelled as fraudulent, we applied a variety of machine learning models to enhance detection accuracy. Data preprocessing steps included feature engineering, scaling, and under-sampling to balance the dataset. We evaluated models such as Logistic Regression, SVM, Random Forest, XGBoost, Neural Networks, and Graph Neural Networks (GNN). XGBoost delivered the best performance, achieving 97% accuracy, with high precision and recall. The GNN model showed potential but requires further optimization. This work underscores the importance of handling imbalanced data and exploring graph-based techniques, with future directions including the use of Graph Attention Networks (GAT) for improved results in fraud detection.

Introduction

Fraud detection in credit card transactions remains a critical challenge due to the increasing sophistication of fraudulent activities. Traditional machine learning methods, while effective to some extent, often struggle to capture complex, relational patterns in transactional data. This project explores the application of Graph Neural Networks (GNNs), which excel at modeling relationships in graph-structured data, to improve fraud detection performance.

Literature Review

Fraud detection in credit card transactions is a critical domain where machine learning (ML) has proven effective. Traditional methods like Support Vector Machines (SVM), Logistic Regression (LR), and Random Forests (RF) have been widely studied for their performance on imbalanced datasets, with ensemble methods like RF often outperforming standalone models due to their ability to handle noise and variance (Nadim et al., 2019). Varmedja et al. (2019) emphasized the importance of metrics such as precision and recall in evaluating models and highlighted preprocessing techniques like oversampling to mitigate class imbalance.

Wang et al. (2021) compared GNNs with traditional machine learning methods and consistently found that GNNs outperform other models in terms of precision and recall, particularly when the data includes strong relational structures, such as shared user-merchant interactions or temporal correlations between transactions.

Research Design and Modelling Methods

The goal of this project is to develop a robust fraud detection system using a combination of traditional machine learning models and advanced techniques such as Graph Neural Networks (GNN). The approach follows several stages: data pre-processing, model selection, hyperparameter optimization, and evaluation. Given the highly imbalanced nature of the dataset (with fraud transactions comprising only 0.1% of all transactions), our approach is designed to address the challenges posed by this class imbalance while achieving high precision, recall, and F1 score.

Several machine learning models were explored to achieve the best fraud detection performance. These include: *Logistic Regression, Support Vector Machine (SVM), Random Forest, XGBoost, Neural Networks and Graph Neural Networks (GNN).* Each model was trained using cross-validation techniques and evaluated based on key metrics: accuracy, precision, recall, and F1 score. Given the business context (fraud detection), recall and F1 score were prioritized to ensure that fraudulent transactions were identified while minimizing false positives.

Data Preparation, Exploration, and Visualization: The IBM credit card transaction dataset consists of 24.4 million transactions with 15 features. Data preparation involved handling missing values, scaling features, and ensuring the dataset was suitable for model training. We also addressed the highly imbalanced class distribution by employing undersampling, where we randomly selected a subset of non-fraudulent transactions to balance the number of fraudulent and non-fraudulent samples (70,243 non-fraud and 29,757 fraud transactions). Comprehensive exploratory data analysis (EDA) was conducted to understand the distribution of fraudulent versus non-fraudulent transactions. This analysis involved examining transaction amounts, merchant types, and temporal patterns using visualizations to highlight patterns and anomalies. For GNN, a graph was created where nodes

represented card IDs and merchant names, and edges included transaction details as attributes, enriching the model with relational data for fraud detection.

Implementation and Programming: The implementation was done in Python using libraries like Scikit-learn (for traditional models), XGBoost, TensorFlow/Keras (for neural networks), PyTorch, and NetworkX (for GNN). We used GridSearchCV for hyperparameter optimization and evaluated models using key metrics such as accuracy, precision, recall, confusion matrix and F1 score, with an emphasis on balanced performance due to the imbalanced nature of the dataset. Several machine learning models were explored to achieve the best fraud detection performance.

Results

Table 1 lists the results for traditional machine learning models and the graph neural network architecture, for the fraud detection task.

Table 1: Summary of Results

Model	Train Time (sec)	Accuracy	Precision	Recall	F1-Score
Logistic Regression	05.28	0.691900	0.483723	0.695017	0.570433
SVM	4203.29	0.859167	0.733875	0.818233	0.773762
Random Forest	282.75	0.952133	0.962356	0.871461	0.914656
XGBoost	162.54	0.974400	0.970581	0.941563	0.955852
MLP	109.45	0.925167	0.886761	0.854926	0.870553
GNN	250.0	0.702500	0.702500	1.0	0.825257
Traditional ML *Neural Networks*					

Logistic Regression and SVM: Logistic Regression achieved modest accuracy and F1 scores, serving as a straightforward baseline. The SVM, while improving performance with non-linear kernels, suffered from high computational costs and scalability issues, limiting its practicality for real-time detection.

Tree-Based Models: Random Forest and XGBoost: Random Forest delivered high accuracy and F1 scores due to its ability to capture complex interactions. However, XGBoost outperformed it with superior precision and recall, achieving a near-perfect F1 score of 0.96. Its gradient-boosting mechanism proved effective in handling both the undersampled data and subtle patterns in the dataset.

Neural Networks: The fully connected neural network demonstrated strong performance, with an F1 score of 0.87, showcasing its ability to generalize patterns in the data. Despite requiring more computational resources, it provided a balance between precision and recall, making it suitable for fraud detection tasks.

Graph Neural Networks (GNNs): GNNs leveraged the inherent graph structure of the dataset by modeling relationships between transactions, cards, and merchants. While achieving an impressive recall of 1.0, indicating all fraudulent cases were identified, the precision and accuracy showed room for improvement. This result underscores the potential of GNNs to extract relational insights but also highlights the need for further optimization, such as exploring attention-based architectures like Graph Attention Networks (GATs).

Key Insights

- **Trade-Offs Between Precision and Recall:** Models like Logistic Regression and Random Forest prioritized general detection accuracy, while XGBoost and GNNs excelled in capturing fraudulent transactions specifically.
- **Impact of Data Representation**: The success of tree-based models and GNNs illustrates the importance of representing transaction relationships effectively.
- **Scalability:** Computational efficiency emerged as a critical factor, especially for models like SVM and neural networks. Future approaches should emphasize both detection capability and scalability.

Conclusion

This study explored the application of various machine learning and deep learning models to address the critical challenge of credit card fraud detection. The extensive dataset of 24.4 million transactions highlighted the complexities associated with class imbalance, as fraudulent cases represented only 0.1% of the data.

Through strategic undersampling, feature engineering, and model optimization, several insights were gained about the strengths and limitations of different approaches:

- XGBoost emerged as the top-performing model, achieving a near-perfect F1 score of 0.96 due to its ability to handle imbalanced data and capture intricate patterns.
- Graph Neural Networks (GNNs) demonstrated promise, particularly in leveraging the relational structure of transactions, but require further refinement to enhance precision without sacrificing recall.
- Tree-based models and neural networks provided a strong foundation for balancing computational efficiency and detection accuracy, while simpler models like Logistic Regression offered baseline performance.

Future Directions: Future research can refine fraud detection systems to be not only more accurate but also more robust and scalable for real-world application by using:

- **Advanced Graph-Based Models**: Exploring architectures like Graph Attention Networks (GATs) could significantly enhance the ability of GNNs to focus on critical relationships between nodes, improving precision and interpretability.
- **Hybrid Models**: Combining GNNs with traditional models like XGBoost may yield synergistic benefits, leveraging both graph-based insights and feature-based learning.
- **Real-Time Scalability**: Addressing the computational demands of large-scale datasets is essential. Techniques like incremental learning or distributed computing could help deploy these models in real-time fraud detection systems.
- **Dynamic Feature Engineering**: Incorporating temporal trends, user behavioural changes, and adaptive thresholds can further enhance detection capabilities.
- **Explainability**: Ensuring model interpretability is vital for operational deployment in financial systems. Techniques such as SHAP values or graph visualization can provide actionable insights to stakeholders.

References

IBM. 2021. “IBM Credit Card Dataset.” 2021. https://ibm.ent.box.com/v/tabformer-data/folder/130747715605

IBM. 2021. “Credit Card Fraud Detection - EDA & Predictive Model.” 2021. https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/0d75884b-99ad-4449-adb8-cd5c8b486446/view?access_token=0f9c4af6f8003247491ac83cf0d2bfbc2f0da909ce017cfd0373d1717308d8bf.

YICHENZHANG1226. 2023. “IBM Credit Card Fraud Detection: EDA+Random Forest.” https://www.kaggle.com/code/yichenzhang1226/ibm-credit-card-fraud-detection-eda-random-forest.

Nadim, A.H., Sayem, I.M., Mutsuddy, A. and Chowdhury, M.S., 2019, December. Analysis of machine learning techniques for credit card fraud detection. In 2019 International Conference on Machine Learning and Data Engineering (iCMLDE) (pp. 42-47). IEEE. https://ieeexplore.ieee.org/document/8995753

Varmedja, Danilo, Milica Karanovic, Stefan Sladojevic, Milica Arsenovic, and Andras Anderla. 2019. "Credit Card Fraud Detection: Machine Learning Approach." *Symmetry* 11 (5): 1–15. https://doi.org/10.3390/sym11050633.

Wu, Lingfei, Fei Long, and Xiangliang Zhang. 2019. "Graph Neural Networks: A Review of Methods and Applications." *arXiv preprint*. https://arxiv.org/abs/1901.00596.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2019). *A Comprehensive Survey on Graph Neural Networks*. https://doi.org/10.1109/TNNLS.2020.2978386

Zhang, G. (2022). Dual-discriminative Graph Neural Network for Imbalanced Graph-level Anomaly Detection. *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*. https://proceedings.neurips.cc/paper_files/paper/2022/hash/98a625423070cfc6ae3d82d4b59408a0-Abstract-Conference.html

Appendix A: Key EDA Charts

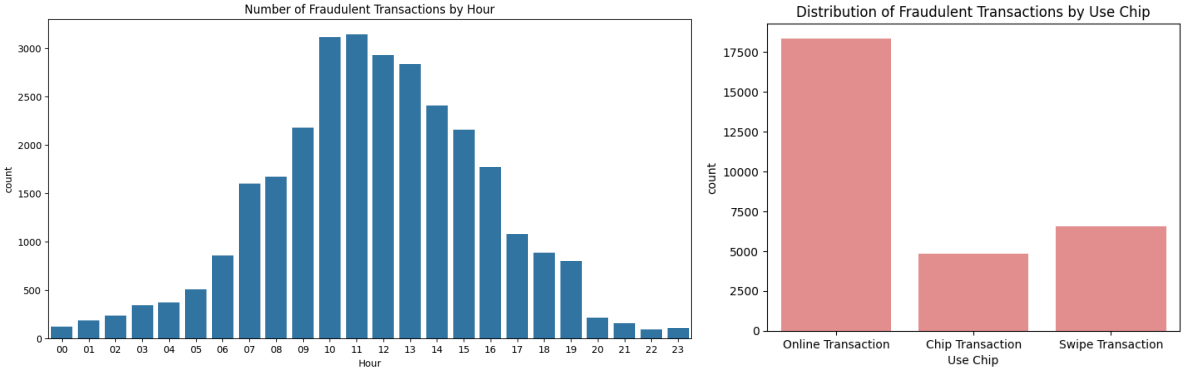

Figure 1: Distribution of fraudulent transactions(Time of day and Chipped Cards)

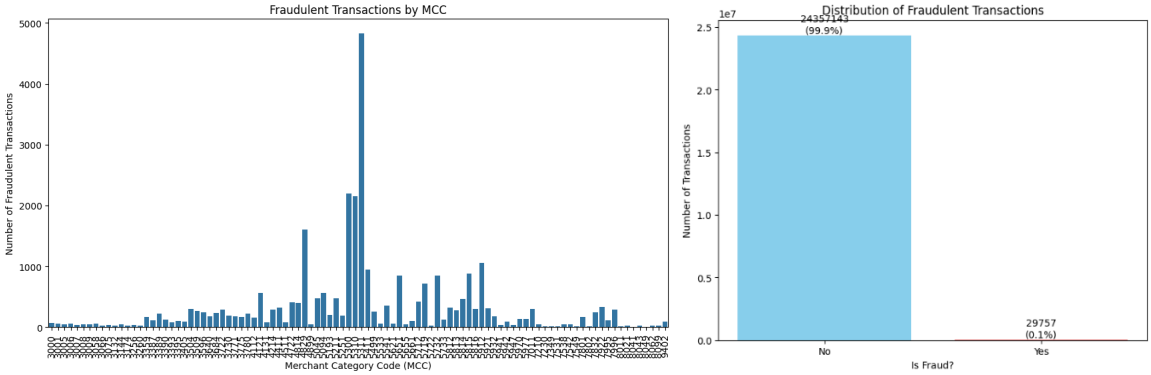

Figure 2: Distribution of fraudulent transactions(MCC and Dataset Proportion)

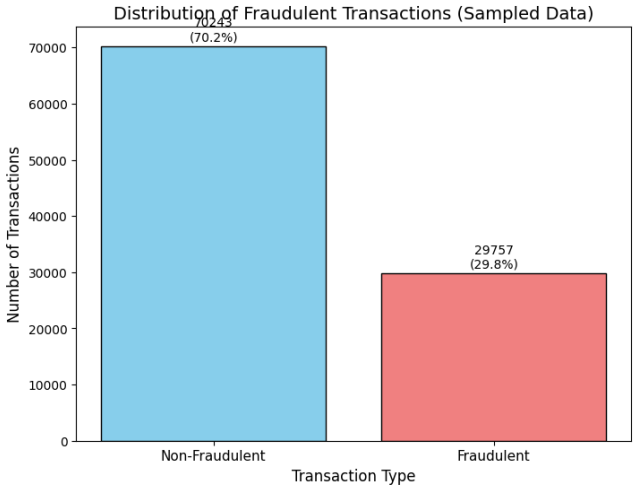

Figure 3: Balanced Dataset (~30% fraudulent transactions)

Appendix B: Model Results Table

	Model	Train Time	Accuracy	Precision	Recall	F1-Score	Best Parameters
Experiment1	Logistic Regression	5.28 seconds	0.6919	0.483723	0.695017	0.570433	{'C': 0.1, 'max_iter': 100, 'penalty': 'l1', 'solver': 'liblinear'}
Experiment2	Support Vector Machine	4203.29 seconds	0.859167	0.733875	0.818233	0.773762	{'C': 1, 'gamma': 0.1, 'kernel': 'rbf'}
Experiment3	Random Forest	282.75 seconds	0.952133	0.962356	0.871461	0.914656	{'max_depth': None, 'max_features': 'sqrt', 'min_samples_split': 2, 'n_estimators': 200}
Experiment4	XGBClassifier	162.54 seconds	0.9744	0.970581	0.941563	0.955852	{'colsample_bytree': 0.8, 'gamma': 0.1, 'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 200, 'subsample': 1.0}
Experiment5	Neural Network	109.45 seconds	0.925167	0.886761	0.854926	0.870553	{'epochs': 50, 'optimizer': 'adam', 'lr': 0.001, 'batch_size': 64}
Experiment6	Graph Neural Network	250.0 seconds	0.7025	0.7025	1.0	0.825257	{'epochs': 300, 'optimizer': 'adam', 'lr': 0.001, 'hidden_dimension': 128}

Figure 4: Result Table

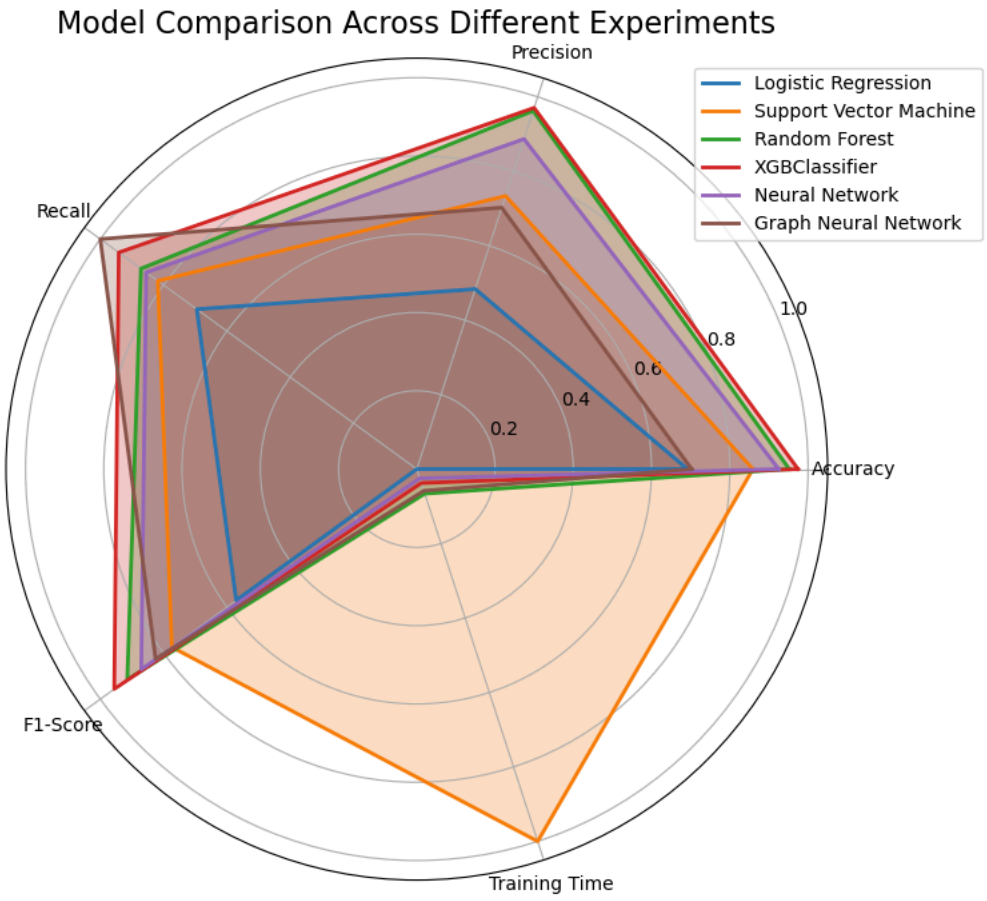

Figure 5: Result Table Visualization using Radar Graph

Appendix C: Confusion Matrix and ROC Curve

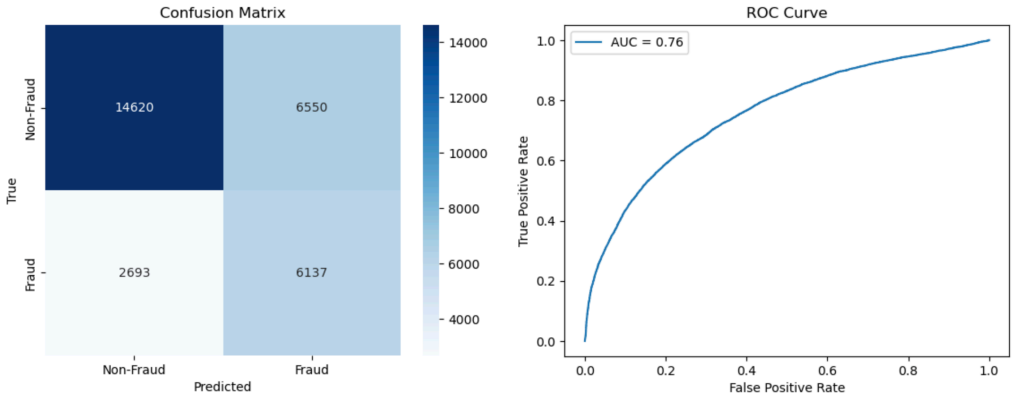

Figure 6: Logistic Regression Results

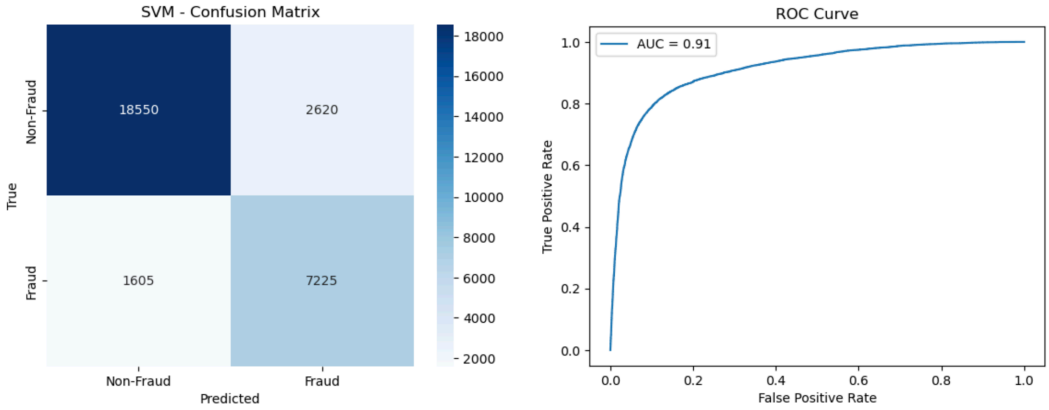

Figure 7: SVM Results

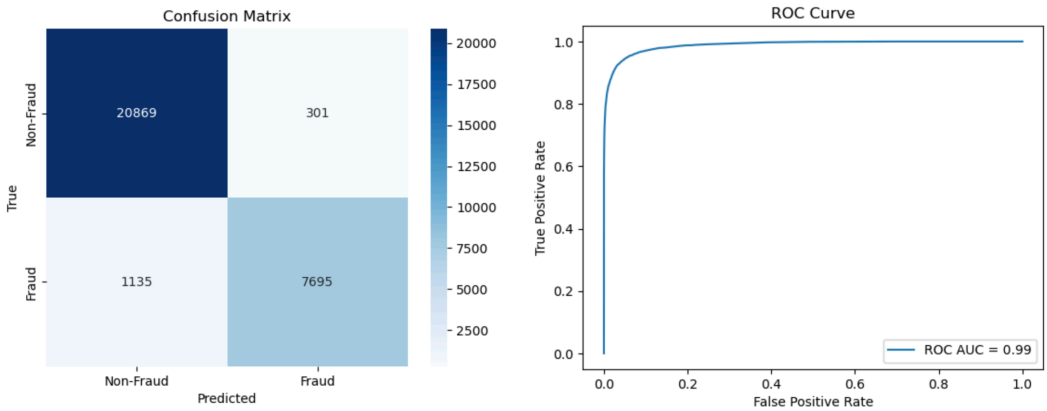

Figure 8: Random Forest Results

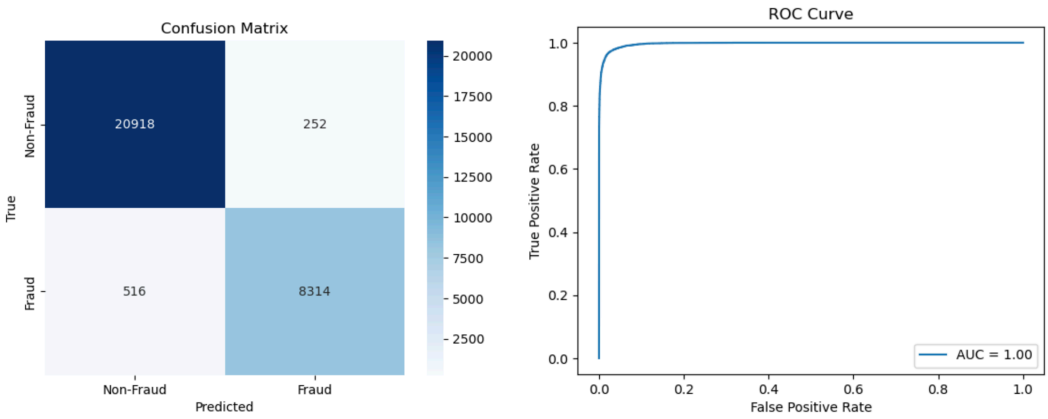

Figure 9: XGBoost Results

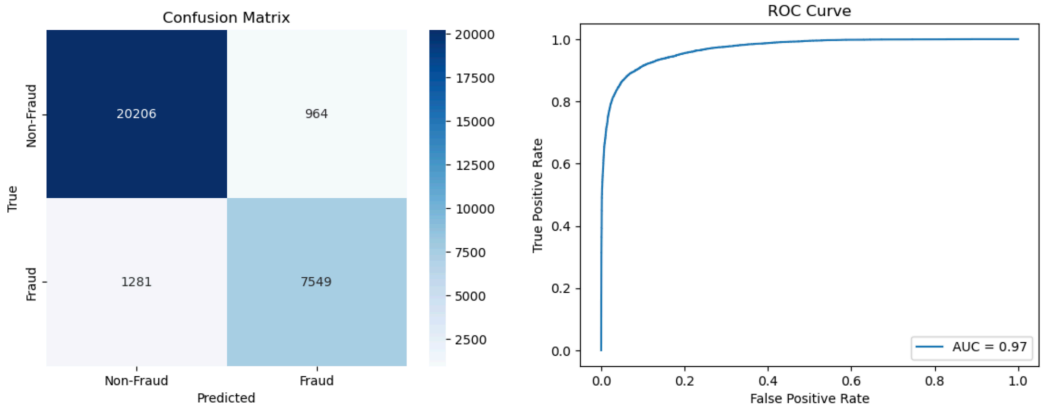

Figure 10: Neural Network Results

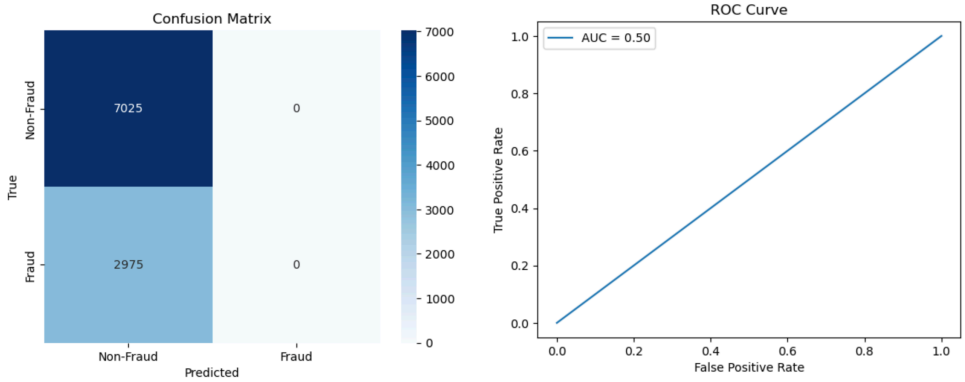

Figure 11: Graph Neural Network Results