BRAIN STATION 23

Project Summary Report

Credit Card Fraud Detection

Submitted By

Muhammad Junayed

Dept. of Electronics and Telecommunication Engineering

Chittagong University of Engineering and Technology

Credit Card Fraud Detection App

Introduction:

This report summarizes my complete work on building a robust credit card fraud detection system based on the Kaggle dataset containing 284,807 transactions. Only 0.172% of these are fraud, making this a highly imbalanced classification problem. I have explored data, trained models, and built a Streamlit app, gaining skills in data analysis, modeling, and deployment.

Day 1: Data Exploration & Cleaning

I began by loading and exploring the dataset using Pandas and Seaborn. Duplicate entries (1,081 rows) were identified and removed. The time distribution of fraud transactions showed concentrated activity in specific time windows (around hours 10–12, 26, and 42). Features like V17, V14, V10, and V12 showed the strongest correlation with the fraud label.

Day 2: Baseline Models and Metrics

I trained initial models using Logistic Regression and Random Forest. Performance was evaluated with metrics such as AUPRC, F1 Score, and Recall instead of Accuracy due to the data imbalance. Logistic Regression had limited performance due to lack of fraud detection sensitivity, while Random Forest provided better recall but required threshold tuning for balance.

Day 3: Handling Imbalance & Advanced Modeling

To address the imbalance, I implemented:

- SMOTE: Synthetic oversampling of the minority (fraud) class
- Undersampling: For benchmarking and fast prototyping

Then I developed models using:

- Random Forest (with class weights)
- XGBoost (with RandomizedSearchCV for hyperparameter tuning)
- LightGBM (for scalable performance)

Feature selection was refined using correlation analysis and model-driven feature importance. Top features included V4, V11, V17, V14, and V10.

Day 4: Streamlit Web Application

I have developed a Streamlit app for fraud prediction. Users can upload CSV file of transections (Time, Amount, V1–V28) to get predictions and fraud case count. Display is limited to 100 rows for large CSVs, with full results downloadable.

Model Performance Summary:

Model	AUPRC	Precision	Recall	F1 Score	ROC-AUC
Logistic Regression	0.75	0.65	0.60	0.62	0.93
Random Forest	0.98	0.96	0.94	0.95	0.99
XGBoost (tuned)	0.99	0.98	0.96	0.97	0.99
LightGBM	0.98	0.97	0.93	0.95	0.98

Challenges and Solutions:

- The main challenge was extreme class imbalance (0.172% fraud).
  → Solved using SMOTE and undersampling.
- Precision vs. Recall trade-off.
  → Solved with threshold tuning.
- Feature interpretability.
  → Solved using SHAP and feature importances.
- Deployment readiness.
  → Solved by saving models, enabling monitoring, and retraining pipeline.

Lesson Learned:

I gained strong hands-on experience with:

- Imbalanced classification and SMOTE
- Ensemble methods: Random Forest, XGBoost, LightGBM
- Model evaluation for rare class detection
- Threshold tuning and metric trade-offs
- Pipeline creation, job scheduling, and model versioning

Conclusion:

This project provided a deep understanding of fraud detection systems in FinTech. By combining data science best practices with powerful algorithms and deployment readiness, I built a high-performing fraud detection system that balances recall and precision, achieving AUPRC with XGBoost and Random Forest and deploying an interactive Streamlit app.

The project is now production-ready and capable of live monitoring, tuning, and retraining.