Credit Card Fraud Detection Using Machine Learning

Industrial Attachment Report

Submitted by:

Yeasir Mahmud

ID 2008036

Department of Electronics And Telecommunication Engineering
Chittagong University of Engineering and Technology

Submitted to:

Brain Station 23 PLC
8th Floor, 2 Bir Uttam AK Khandakar Road
Mohakhali Commercial Area, Dhaka 1212, Bangladesh
Phone: +880 2 222290728
Email: sales@brainstation-23.com
`https://brainstation-23.com`

May 23, 2025

1. Introduction

Credit card fraud detection is essential in financial technology due to the increasing rate of online transactions. However, fraud cases are rare and often hidden among large volumes of legitimate transactions. This project aimed to develop a supervised machine learning model to detect fraud with high precision and deploy it via a Streamlit web app.

2. Dataset

We used a publicly available dataset from Kaggle, consisting of 284,807 credit card transactions from European cardholders in 2013. Among them, only 492 are labeled as fraudulent (0.172%), presenting a severe class imbalance challenge. Features include:

- **Time, Amount** — numeric features
- **V1–V28** — anonymized features via PCA
- **Class** — 0 for normal, 1 for fraud

3. Methodology

After validating and cleaning the data, we scaled the `Time` and `Amount` features using StandardScaler. We performed an 80/20 stratified split to preserve class distribution. Multiple models were trained and evaluated using metrics suitable for imbalanced classification.

Models Evaluated:

- Logistic Regression
- Decision Tree
- Naive Bayes
- K-Nearest Neighbors (KNN)
- Random Forest (best performer)

Performance Summary:

Table 1: Model Performance Metrics

Model	Accuracy	Precision	Recall	AUC
Logistic Regression	0.9986	0.86	0.62	0.951
Decision Tree	0.9990	0.89	0.70	0.974
Naive Bayes	0.9630	0.13	0.81	0.831
KNN (k=5)	0.9985	0.86	0.59	0.938
Random Forest	**0.9992**	**0.91**	**0.76**	**0.980**

4. Streamlit App

To enable real-time fraud detection, a web app was developed using Streamlit. Users can upload a CSV file, and the app will:

- Validate required features
- Preprocess data using the trained scaler
- Display fraud predictions with probabilities
- Visualize fraud vs. normal cases
- Offer downloadable results

Custom CSS was used for a clean UI with feedback and error handling.

5. Reflection

This project helped me bridge theory and practice in machine learning. I learned how to manage highly imbalanced datasets, evaluate models beyond accuracy, and deploy interactive ML apps. Handling CSV validation and UI responsiveness in Streamlit presented practical challenges that improved my development skills.

6. Conclusion

This project successfully delivered a robust credit card fraud detection system leveraging Random Forest classification and interactive deployment via Streamlit. The experience deepened my understanding of imbalanced data challenges, model evaluation, and real-world application deployment. Future work could focus on further improving recall and exploring advanced techniques like anomaly detection or deep learning models.

7. References

- Kaggle Dataset: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud
- Scikit-learn: https://scikit-learn.org/
- Streamlit Docs: https://docs.streamlit.io/
- Breiman, L. Random Forests (2001): https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf