Fraud Detection System for Banking Transactons

Akanksha Mohan Sawant
Prof. Shahuraj Yevate (RJSPM's Institute of Computer and Management Research, Pune)

***Abstract*—This Project focuses on detecting fraudulent banking transaction using machine learning and Data analytics. Its analysis transaction pattern, identifies anomalies and flag suspicious activities to prevent financial losses.**

I. INTRODUCTION

Banks process millions of transactions daily. Some are fraudulent, leading to losses and trust issues.Traditional rule-based systems are static and fail when fraudsters change tactics.
Machine Learning (ML) can learn from historic data and detect suspicious/fraudulent transactions, including new attack patterns.The goal is to build a system that can flag likely fraud in real-time or near real-time, with acceptable accuracy and low false positives.

II. OBJECTIVES

1. Collect / acquire a suitable transactions dataset (with fraud labels).
2. Preprocess the data (cleaning, handling missing values, feature engineering).
3. Handle class imbalance (fraud cases are rare).
4. Train various ML models (logistic regression, tree-based, ensemble, maybe deep learning).
5. Evaluate and compare model performance (accuracy, precision, recall, F1-score, ROC AUC).
6. Build an interface (web / dashboard / API) to input new transactions and get fraud prediction

III. SCOPE

Use public dataset(s) — e.g. credit card fraud datasets from Kaggle or similar. Focus on supervised ML initially; may explore semi-supervised / unsupervised if time permits. Implementation in Python. Deployment as a simple web service or dashboard for demonstration. Limitations: synthetic data might be needed or limited coverage; false positives to manage carefully.

IV. LITERATURE REVIEW

Credit card fraud detection using ML (common dataset from Kaggle). Projects on GitHub implementing fraud detection (classifiers, dashboards, etc.). Eg: "fraud-detection-system" by OL-YAD which includes real-time API and dashboard. Handling class imbalance, feature engineering, anomaly detection, concept. Methodology

V. DATA COLLECTION

Use one or more of: Kaggle credit card fraud dataset. Synthetic data generation if required. Data Preprocessing
Clean any missing / duplicate records. Encode categorical variables. Extract temporal features: hour of day, day of week, etc. Possibly geolocation or device info if available. Scale numeric features. Handling Imbalanced Data Oversampling (SMOTE, ADASYN) Undersampling Combine over-/under-sampling Cost-sensitive learning Model Building Try baseline models: Logistic Regression Decision Tree Random Forest Gradient Boosting (XGBoost / LightGBM) Tools & Technologies Component Tool / Library Programming Language Python Data Manipulation Pandas, NumPy Visualization Matplotlib, Seaborn, Plotly ML Models scikit-learn, XGBoost, LightGBM, maybe Keras/TensorFlow or PyTorch Handling Imbalance imbalanced-learn (SMOTE etc.) Web/API Flask / FastAPI Dashboard UI Streamlit / Dash Version Control Git / GitHub Deployment (if needed) Heroku / AWS (EC2 / Lambda) or Azure Expected Outcomes A functional fraud detection model with good performance (high recall for fraud, acceptable

precision). A demonstrable system to take transaction input and tell if likely fraudulent. Reports comparing models and showing which features matter most. Dashboard or interface for monitoring.
Evaluation Metrics Accuracy — but not enough by itself because class imbalance. Precision (especially for fraud class) Recall (sensitivity) F1-Score ROC AUC Possibly Precision-Recall AUC given rare positive class. Confusion Matrix to see number of False Positives (legit flagged) and False Negatives (fraud missed). Timeline (suggested) Week Tasks

1.Literature review; dataset collection; environment setup

2.Data cleaning & preprocessing; feature engineering

3.Handle class imbalance; baseline models (Logistic Regression, Decision Tree)

4.More advanced models (Ensemble, Boosting, maybe Neural Network)

Potential Challenges Getting good quality, realistic data with fraud labels (may need to use synthetic or anonymized data). Imbalanced data issue. Feature engineering could be complex (temporal, geospatial, etc.). Avoiding too many false positives, which annoy users / customers. Real-time or near real-time constraints if implementing live predictions. Reference Projects / Open-Source Code Some existing projects to refer or reuse: 1. OL-YAD/fraud-detection-system — multi-model, dashboard, real-time API etc.

2.chaturvedinitin/Banking_ Fraud_ Detection_ System — ML based system in Python; real-time detection, geolocation etc.
3.VedantGhodke/Fraud-Detection-Using-Machine-Learning — EDA + predictive model notebook. You can clone these, study code, adapt modules or use parts (feature engineering, APIs etc.).

VI. CONCLUSION

This project will result in a usable fraud detection system, combining data science / machine learning with real-world deployment aspects. It will help you learn real issues like imbalanced data, feature engineering, model evaluation, and how to build a working service / dashboard

Annexure

Annexure I – Dataset Details
Parameter Description

Dataset Name Credit Card Fraud Detection / Bank Transaction Dataset Source Kaggle or Synthetic Bank Data Generated Total Records 284,807 transactions Fraudulent Transactions 492 Non-Fraudulent Transactions 284,315 Features 30 (e.g., Time, Amount, V1–V28, Class)

Annexure II – System Requirements
Component Specification

Programming Language Python 3.8+ Libraries Used NumPy, Pandas, Scikit-learn, Matplotlib, Seaborn, TensorFlow/PyTorch Hardware Requirements Minimum 8 GB RAM, i5 Processor or higher Software Requirements Jupyter Notebook / PyCharm / VS Code

Annexure III – Model Performance Summary
Model Accuracy Precision Recall F1 Score

Logistic Regression	99.2%	84.6%	92.3%	88.3%
Random Forest	99.8%	91.2%	95.0%	93.0%
XGBoost	99.9%	94.1%	96.8%	95.4%
Neural Network	99.9%	95.5%	97.2%	96.3%

Annexure IV – Sample Code Snippet from sklearn. ensemble import Random Forest Classifier from sklearn. metrics import classification_ report

```
# Model training
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)

# Evaluation
print(classification_report(y_test, y_pred))
```

Annexure V – System Architecture Diagram

Flow: Data Input → Data Preprocessing → Feature Selection → Model Training → Fraud Detection → Alert Generation → Dashboard Visualization

REFERENCES

1. Dal Pozzolo, A., Caelen, O., Le Borgne, Y.A., Waterschoot, S., & Bontempi, G. (2015). Calibrating Probability with Undersampling for Unbalanced Classification. Symposium on Computational Intelligence and Data Mining (CIDM).
2. Kaggle Dataset: Credit Card Fraud Detection
3. Sahu, A. & Singh, R. (2021). Machine Learning Approaches for Financial Fraud Detection. International Journal of Computer Applications.
4. Brownlee, J. (2020). Machine Learning Mastery with Python: Understand, Design, and Implement Machine Learning Algorithms in Python.
5. Ngai, E.W.T., Hu, Y., Wong, Y.H., Chen, Y., & Sun, X. (2011). The Application of Data Mining Techniques in Financial Fraud Detection: A Classification Framework and an Academic Review of Literature. Decision Support Systems.