Project Report

Online Payments Fraud Detection using Machine Learning

IBM SkillsBuild | AI/ML Final Year Project

Academic Year 2024–25

1. Executive Summary

This project presents a machine learning-based system for detecting fraudulent online payment transactions. A Random Forest classifier was trained on the PaySim synthetic financial dataset (6.3 million transactions) and deployed as a Flask web application. The system achieves 99.96% overall accuracy with 100% precision on fraud detection and a 75% recall rate, resulting in an F1-score of 0.85 for the fraud class.

2. Problem Statement

Online payment fraud is a growing global threat, with annual losses exceeding $32 billion. Traditional rule-based detection systems are static, easy to evade, and produce high false-positive rates. This project addresses the need for an intelligent, adaptive, and accessible fraud detection solution.

3. Objectives

1. Build a high-accuracy fraud detection model using real-world transaction features
2. Handle the severe class imbalance inherent in fraud datasets
3. Deploy the model as an accessible web application for non-technical users
4. Achieve at least 99% accuracy and 0.80 F1-score on the fraud class

4. Methodology

4.1 Dataset

The PaySim dataset (Kaggle) contains 6.36 million synthetic mobile money transactions generated using agent-based simulation of real transaction logs. It includes ground-truth fraud labels for 8,213 transactions (0.13% of total).

4.2 Preprocessing

- Label encoding of categorical 'type' column (PAYMENT=0, TRANSFER=1, CASH_OUT=2, DEBIT=3, CASH_IN=4)
- Selection of 7 most predictive features (dropped nameOrig, nameDest, isFlaggedFraud)

- 80/20 stratified train-test split

4.3 Model Training

- Algorithm: RandomForestClassifier (100 estimators, scikit-learn defaults)
- Training time: approximately 15 minutes on standard hardware
- No feature scaling required (tree-based models are scale-invariant)
- Decision threshold tuned from 0.50 to 0.20 to improve fraud recall

5. Results

Metric	Target	Achieved
Overall Accuracy	>= 99%	99.96% ✓
Fraud Precision	>= 90%	100.00% ✓
Fraud Recall	>= 70%	75.00% ✓
Fraud F1-Score	>= 0.80	0.85 ✓
Web App Response	< 2s	~0.18s ✓

6. Web Application

The trained model was serialized using Python's pickle module and integrated into a Flask web application. The application provides three pages: a home page introducing the project, a prediction form for entering transaction details, and a result page displaying the fraud verdict and probability score.

7. Conclusion

All project objectives were successfully met. The Random Forest classifier demonstrated excellent performance on the PaySim dataset, particularly its perfect precision (zero false positives) which is critical in a real-world deployment where blocking legitimate customers is highly undesirable. The Flask web application provides an accessible, user-friendly interface that requires no ML knowledge to operate.

Future Work

- Add SHAP-based explainability to show which features drove each prediction
- Experiment with XGBoost and LightGBM to improve recall beyond 75%
- Deploy to cloud platform (Render/Railway) for public access
- Add batch CSV upload mode for screening multiple transactions at once
- Explore real-world datasets (IEEE-CIS Fraud Detection) for validation

8. References

- Lopez-Rojas, E.A. et al. (2016). PaySim: A Financial Mobile Money Simulator for Fraud Detection. EMSS 2016.
- Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.
- scikit-learn: Machine Learning in Python. Pedregosa et al., JMLR 12, 2011.
- Pallets Projects. Flask Web Framework. https://flask.palletsprojects.com
- IBM SkillsBuild Platform. AI/ML Project Guidelines. 2024.