# Credit Card Fraud Detection using Machine Learning and Deep Learning

**Tarigopula Abhilash**
Master's (M2) Data Science

**YARRAGANGIREDDY Pradeep Kumar Reddy**
Master's (M2) Data Science

January 2026

**Abstract**

This project presents an end-to-end study of **credit card fraud detection** using classical Machine Learning models, Explainable AI techniques, and a Deep Learning baseline. The dataset is extremely imbalanced, with fraudulent transactions representing only **0.17%** of the data.

To address this challenge, the project emphasizes proper preprocessing, leakage-free experimental design, suitable evaluation metrics (PR-AUC, F1-score), systematic model comparison, hyperparameter tuning using **RandomizedSearchCV**, and interpretability through **SHAP**.

The results demonstrate that classical machine learning models, particularly tree-based and instance-based approaches, outperform deep learning on this tabular, highly imbalanced dataset. The study highlights the importance of methodological rigor and interpretability in real-world fraud detection systems.

# Contents

# 1 Introduction

Credit card fraud detection is a critical problem for financial institutions, with global losses exceeding billions of dollars annually. From a machine learning perspective, fraud detection presents several challenges:

- Extreme class imbalance (fraud rate $\approx 0.17\%$)
- High-dimensional feature space
- Complex and non-linear decision boundaries
- High cost of false negatives (missed fraud)

This project aims to build a robust and interpretable fraud detection pipeline that reflects both academic rigor and practical deployment considerations.

# 2 Dataset Description

## 2.1 Dataset Source

- Dataset: Credit Card Fraud Detection
- Source: Kaggle (ULB credit card dataset)
- File: `creditcard.csv`

The dataset contains anonymized transactions made by European cardholders. Features `V1{V28` are PCA-transformed for confidentiality, while `Time` and `Amount` are raw numerical features.

## 2.2 Dataset Characteristics

Table 1: Dataset Summary

| | |
|---|---|
| Total transactions | 284,807 |
| Fraudulent transactions | 492 (0.17%) |
| Features | 30 |
| Missing values | 0 |

# 3 Preprocessing and Experimental Design

## 3.1 Train–Validation–Test Split

To prevent data leakage, the dataset was split using stratified sampling:

- Training set: 60%
- Validation set: 10%
- Test set: 30%

Each split preserves the original fraud ratio.

### 3.2 Feature Engineering

- PCA features (`V1{V28`) kept unchanged
- `Amount`: log-transformation + RobustScaler
- `Time`: RobustScaler

All transformations were fit on the training set only and applied consistently to validation and test sets.

## 4 Evaluation Metrics

Accuracy is misleading for fraud detection due to class imbalance. Therefore, the following metrics were used:

- Recall
- Precision
- F1-score
- ROC-AUC
- PR-AUC

PR-AUC is particularly informative, as it focuses on the minority (fraud) class.

## 5 Machine Learning Models

The following classical models were trained:

- Logistic Regression
- K-Nearest Neighbors (k=5)
- Decision Tree
- Random Forest
- Support Vector Machine (RBF)

All models use `class_weight='balanced'` where applicable.

### 5.1 Hyperparameter Tuning

Hyperparameter tuning was performed only for the Random Forest model using **RandomizedSearchCV**. GridSearchCV was intentionally avoided due to computational cost.

- Search method: RandomizedSearchCV
- Metric: F1-score
- Cross-validation: Stratified 3-fold

The tuned model was saved and used for final evaluation and explainability.

## 6 Explainability using SHAP

Model interpretability was achieved using SHAP (SHapley Additive Explanations) applied to the tuned Random Forest model.

- Global explainability: feature importance
- Local explainability: individual fraud vs non-fraud predictions

Limitations include PCA feature opacity and feature independence assumptions.

## 7 Deep Learning Baseline

A minimal Artificial Neural Network (ANN) was implemented as a baseline:

- Two hidden layers (64, 32 neurons)
- Batch Normalization and Dropout
- Class weights to address imbalance

The ANN achieved reasonable performance but did not outperform classical machine learning models.

## 8 Discussion

Key findings:

- Classical ML outperforms deep learning on tabular imbalanced data
- KNN achieved strong F1-score but is impractical for deployment
- Random Forest provides the best balance between performance, interpretability, and scalability
- Proper metric selection is crucial

## 9 Conclusion

This project demonstrates a complete and rigorous fraud detection pipeline. By combining careful preprocessing, appropriate metrics, hyperparameter tuning, explainability, and critical analysis, the study satisfies both academic and practical requirements of a Master's-level Data Science project.

**Final Recommendation:** While KNN achieved strong predictive performance, the tuned Random Forest model is recommended for real-world deployment due to its stability, interpretability, and efficiency.

# References


1. Kaggle Credit Card Fraud Dataset: `https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud`
2. Breiman, L. (2001). Random Forests. *Machine Learning*
3. Lundberg, S. M., & Lee, S.-I. (2017). SHAP. *NeurIPS*