



**Submitted in fulfillment of the requirements for the degree
Master of Science in Artificial Intelligence**

Enhanced Credit Card Fraud Detection Using XAI

Prepared by

Name: Bothina Ahmed Hessen Osman

ID No.: 243002096

Supervisor Name: Dr. Hager Saleh

Agenda Overview

Problem Statement

Objectives

Related Works

Contribution & Research Gap

Methodology and Proposed Approach

Results

Conclusion

Introduction

- Digital transactions are increasing rapidly, bringing a parallel rise in credit card fraud.
- Traditional detection systems are static and produce high false positives.
- Fraud data is extremely imbalanced, making accurate prediction difficult.
- AI-based fraud detection using ML, DL, and XAI offers a more adaptive, accurate, and transparent solution.
- **This project** presents a full pipeline—from preprocessing to modelling and explainability—to build a high-performance fraud-detection framework.

Problem Statement

- Rapid growth of digital payments has increased credit card fraud risks.
- Fraud datasets are highly imbalanced ($<0.2\%$), limiting traditional ML performance.
- Rule-based and classical models are static and fail to adapt to evolving fraud patterns.
- Many ML/DL models act as black boxes, lacking transparency and trust.
- High false positives increase operational cost and customer dissatisfaction.

Objectives

Main Objectives

- Design and evaluate an adaptive, high-accuracy fraud detection framework integrating ML, DL, and Explainable AI.

Specific Objectives:

- Improve fraud detection performance under extreme class imbalance.
- Develop a stacking ensemble combining ML and DL models.
- Reduce false positives while maintaining high recall.
- Apply XAI techniques (SHAP, LIME) to ensure transparency.
- Enable real-world deployment and adaptability.

Related Works

Overview of Previous Studies

Recent studies (2020–2024) have applied various machine learning and deep learning techniques to credit card fraud detection. Results show promising accuracy levels ranging from **83% to 96%**, depending on the methodology:

- CNN variants (without pooling) struggled with feature loss, achieving lower accuracy.
- Hybrid models like **CNN–SVM** (Convolutional Neural Network - Support Vector Machine) and **BiLSTM–GRU** (Bidirectional Long Short-Term Memory - Gated Recurrent Unit) improved performance but still suffered from data imbalance and limited adaptability.
- **LSTM-based approaches** (Long Short-Term Memory) reached higher accuracy, yet they remain resource-intensive and lack real-time deployment validation.

Related Works

Study Title & Source	Year	Method	Dataset	Result	Limitation
Credit Card Fraud Detection Using CNN without Pooling	2023	CNN	Kaggle	83.72%	Feature loss → low accuracy
Hybrid CNN–SVM for Fraud Detection	2023	CNN–SVM	Kaggle	91.05%	Still imbalanced, offline only
BiLSTM–GRU–MaxPooling	2020	BiLSTM–GRU	Kaggle	91.37%	Complex, hard real-time deployment
LSTM Approach for Fraud Detection	2024	LSTM	Kaggle	96.3%	Resource-intensive, no real-time test

Contribution

Gap of Previous Studies

Despite advances, these studies did not provide a **practical real-time framework**. They fell short in:

- **Handling imbalanced datasets** – limited use of advanced resampling or cost-sensitive methods.
- **Sustainability of accuracy** – lack of adaptive learning to respond to evolving fraud patterns.
- **Scalability** – high computational cost limits deployment in real-world banking or e-commerce environments.
- **Lack of Explainable AI (XAI)** – previous studies rarely incorporated explainability techniques, limiting transparency, trust, and interpretability of fraud detection decisions.

Methodology & Proposed Approach

Data Collection

- We used the **Kaggle Credit Card Fraud Detection European dataset 2013** , which contains:
- **284,807 transactions**
- Only **492 fraud cases (0.172%)**
- Extremely **imbalanced dataset**
- Features include:
 - 28 PCA-transformed numerical variables
 - Amount
 - Time
 - Target label: **Class** (0 = legit, 1 = fraud)

Methodology & Proposed Approach

Data Collection

- **Why this dataset?**
 - Real-world financial transactions
 - Extreme imbalance → perfect for evaluating ML/DL models
 - Common benchmark dataset for fraud detection research



Methodology & Proposed Approach

Data Preprocessing

Our preprocessing pipeline handled real-world challenges to prepare the data for ML & DL models.

- Data Cleaning
- Train-Validation-Test Split
- Data Scaling
- Handling Class Imbalance

Methodology & Proposed Approach

- **Data Cleaning**
 - Verified no missing values
 - Checked for invalid or corrupted records
 - Confirmed dataset integrity
- **Train-Validation-Test Split**
 - We used a **stratified split**:
 - 70% training
 - 30% validation & testing

This ensures fraud and non-fraud ratio is preserved across splits.

Methodology & Proposed Approach

- **Data Scaling**

Because ML/DL models are sensitive to different scales:

- **Standard Scaler** was applied to:
 - Training set
 - Validation set
 - Test set

Keeping the mean = 0 and variance = 1.

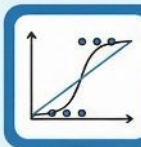
Methodology & Proposed Approach

- **Handling Class Imbalance**
 - We applied **four resampling techniques** to ensure fairness and compare performance:
 - Random Oversampling
 - SMOTE
 - Random Undersampling
 - Tomek Links

Each sampling technique was applied **only on training data**.

Methodology & Proposed Approach

Machine Learning Models for Fraud Detection Project



LOGISTIC REGRESSION (LR)

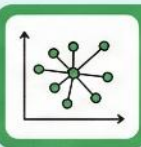
Simple, understandable categorization model. Assumes straight line relationship between features & fraud probability.

Why LR is used:

- ✓ Baseline classifier.
- ✓ Calibrated probability outputs.
- ✓ Works well with skewed data (with resampling).
- ✓ Fast & efficient on huge datasets.

Implementation Steps:

- ⚙ Fine-tuned with GridSearchCV (C, solver, penalty, max_iter).
- ⚙ Trained per imbalance technique.
- ⚙ Tested using Accuracy, Precision, Recall, F1, AUC.



K-NEAREST NEIGHBORS (KNN)

Non-parametric model sorting transactions based on closest neighbors.

Why KNN is good:

- 🔍 Finds patterns in small areas.
- 🔍 Effective when fraud cases are close together.
- 🔍 No training needed (distance-based predictions).

Implementation Details:

- ⚙ Adjusted using n_neighbors and weights.
- ⚙ Tested with all balancing methods.



DECISION TREE (DT)

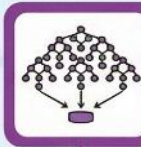
Divides data into branches based on most useful qualities.

Benefits for finding fraud:

- 🔥 Handles nonlinear & complicated interactions.
- 🔥 Easy to visualize.
- 🔥 Less affected by feature scaling.

Implementation Setup:

- ⚙ max_depth.
- ⚙ criteria (gini, entropy).



RANDOM FOREST (RF)

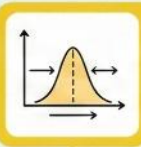
Ensemble of decision trees trained on random data parts.

Why RF is good:

- 💜 Works well with high-dimensional PCA.
- 💜 Stops overfitting by averaging.
- 💜 Learns nonlinear patterns.
- 💜 More stable than individual trees.

Implementation Details:

- ⚙ Adjusted with n_estimators and max_depth.
- ⚙ Tested with every resampling method.



NAIVE BAYES (GaussianNB)

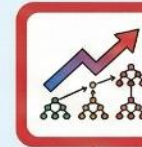
Probabilistic classifier assuming features are not related.

Why it's useful:

- 🚀 Very fast.
- 🚀 Works well with high-dimensional data.
- 🚀 Provides probability estimates.

In the project:

- ⚙ Adjusted with var_smoothing.



XGBOOST (XGBClassifier)

Strong gradient boosting model, popular in competitions.

XGBoost is great for fraud:

- 🚀 Finds complex nonlinear associations.
- 🚀 Works well with unbalanced data.
- 🚀 Supports regularization (stops overfitting).
- 🚀 Excellent predictive performance.

Implementation Setup:

- 📊 Adjusted with n_estimators, learning_rate, max_depth.
- 📊 Got a high AUC score.
- 📊 Used subsequently in SHAP explainability.

Overview of Classical Machine Learning Models Used for Fraud Detection Analysis and Prediction.

Machine Learning Results Summary

Table 1 – Random Oversampling

Model	Recall	F1-Score	AUC	Precision	Accuracy	Best Aspect
Logistic Regression	0.9785	0.0667	0.8783	0.1240	0.9677	Weak precision
KNN	0.9993	0.8243	0.8243	0.8243	0.9120	Top recall
Decision Tree	0.9990	0.7576	0.6757	0.7142	0.8376	Good recall
Random Forest	0.9995	0.9655	0.7567	0.8484	0.9499	★ Best overall ML model
Naive Bayes	0.9743	0.0541	0.8378	0.1018	0.9545	Weak performance
XGBoost	0.9994	0.8923	0.7838	0.8345	0.9607	Strong F1

Machine Learning Results Summary

Table 2 – Random Undersampling

Model	Recall	F1-Score	AUC	Precision	Accuracy	Best Aspect
Logistic Regression	0.9768	0.0630	0.8918	0.1177	0.9733	Weak precision
KNN	0.9771	0.0629	0.8783	0.1175	0.9491	Weak
Decision Tree	0.8882	0.0136	0.8918	0.0268	0.8900	Worst ML performance
Random Forest	0.9828	0.0813	0.8648	0.1486	0.9690	Low F1
Naive Bayes	0.9654	0.0406	0.8378	0.0775	0.9533	Weak
XGBoost	0.9654	0.0418	0.8648	0.0798	0.9711	Weak

Machine Learning Results Summary

Table 3 - SMOTE

Model	Recall	F1-Score	AUC	Precision	Accuracy	Best Aspect
Logistic Regression	0.9772	0.0631	0.8783	0.1178	0.9657	Low F1
KNN	0.9989	0.6363	0.8513	0.7283	0.9253	Good recall
Decision Tree	0.9976	0.3943	0.7567	0.5185	0.8773	Moderate
Random Forest	0.9994	0.8823	0.8108	0.8450	0.9644	★ Very strong
Naive Bayes	0.9761	0.0578	0.8378	0.1082	0.9529	Weak
XGBoost	0.9988	0.6185	0.8108	0.7017	0.9699	Good

Machine Learning Results Summary

Table 4 – Tomek Links

Model	Recall	F1-Score	AUC	Precision	Accuracy	Best Aspect
Logistic Regression	0.9992	0.9200	0.6216	0.7419	0.9604	Strong F1
KNN	0.9995	0.9218	0.7972	0.8550	0.9120	Very strong recall
Decision Tree	0.9993	0.9137	0.7162	0.8030	0.8647	Good
Random Forest	0.9995	0.9500	0.7702	0.8507	0.9731	★ Best ML overall
Naive Bayes	0.9784	0.0621	0.8108	0.1153	0.9545	Weak
XGBoost	0.9995	0.9491	0.7567	0.8421	0.9612	Very strong

MODEL PERFORMANCE SUMMARY: RANDOM FOREST WITH SAMPLING TECHNIQUES

➤ Random Oversampling



Best Model:
Random Forest

Highest F1-Score: **0.9655** ↑

Very high Recall: **0.9995** Strong Precision: **0.8484** High Accuracy: **0.9499**

Most balanced performance under oversampling.

➤ Random Undersampling



Best Model:
Random Forest

Best F1-Score (undersampling): **0.0813**

Highest Precision: **0.1486** Strong Recall: **0.9828** High Accuracy: **0.9690**

Least performance drop compared with other models.

➤ SMOTE



Best Model:
Random Forest

Highest F1-Score: **0.8823** ↗

Very high Recall: **0.9994** Good Precision: **0.8450** High Accuracy: **0.9644**

Most stable and well-balanced performance with SMOTE.

➤ Tomek Links



Best Model:
Random Forest

Highest F1-Score: **0.9500** ↗

Near-perfect Recall: **0.9995** Highest Accuracy: **0.9731** Strong Precision: **0.8507**

Best overall behavior with Tomek Links.

★ Best Overall Combination



Random Forest + Tomek Links

Strongest and most balanced metrics across all techniques



Highest overall Accuracy: **0.9731**



Very strong F1-Score: **0.9500**

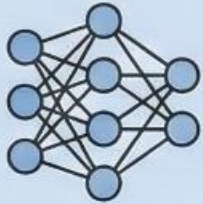


Near-perfect Recall: **0.9995**



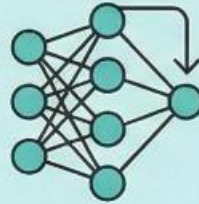
Tomek Links reduced class-boundary noise → cleaner learning.
Identified as the most effective model-sampling combination overall.

COMPARISON OF DEEP LEARNING MODELS IN FRAUD DETECTION



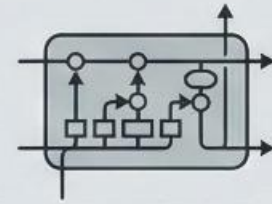
DNN
(DEEP NEURAL
NETWORK)

-  Learns complex non-linear patterns
-  Works on tabular transaction data
-  No time dependency



RNN
(RECURRENT
NEURAL NETWORK)

-  Handles sequential data
-  Uses previous transaction context
-  Suffers from vanishing gradient



LSTM
(LONG SHORT-TERM
MEMORY)

-  Captures long-term dependencies
-  Solves vanishing gradient problem
-  Best for transaction sequences

Deep Learning Results Summary

Model	Recall	F1-Score	AUC	Precision	Accuracy	Notes
DNN	0.9994	0.8923	0.7838	0.8345	0.9735	Strong performance
RNN	0.9994	0.9193	0.7702	0.8382	0.9652	Captures sequences
LSTM	0.9994	0.8787	0.7838	0.8285	0.9802	★ Best DL model

DEEP LEARNING MODELS PERFORMANCE COMPARISON



► DEEP NEURAL NETWORK (DNN)

 Recall:
0.9994

 F1-Score:
0.8923

 AUC:
0.7838

 Precision:
0.8345

 Accuracy:
0.9735

Notes: Strong and stable overall performance



► RECURRENT NEURAL NETWORK (RNN)

 Recall:
0.9994

 F1-Score:
0.9193

 AUC:
0.7702

 Precision:
0.8382

 Accuracy:
0.9652

Notes: Effective at capturing sequential behavior

 **Best DL Model**

► LONG SHORT-TERM MEMORY (LSTM) — ★ BEST DL MODEL

 Recall:
0.9994

 F1-Score:
0.8787

 AUC:
0.7838

 Precision:
0.8285

 Accuracy:
0.9802 → Highest among DL models

Notes: ★ Best Deep Learning model due to superior accuracy and stable overall metrics

★ BEST OVERALL DEEP LEARNING MODEL

 **LSTM**

- Because it provides: The highest Accuracy (0.9802)
-  Very high Recall (0.9994)
-  Good overall balance across metrics
-  Strong ability to learn temporal patterns relevant to fraud detection

Comparison ML vs DL

Aspect	Machine Learning	Deep Learning	Winner
Recall	0.9995 (RF)	0.9994 (DL)	Tie
Precision	0.8507 (RF)	0.8382 (RNN)	ML
Accuracy	0.9731	0.9802 (LSTM)	DL
F1 Score	0.95 (RF)	0.919 (RNN)	ML
Robustness	Strong	Stronger with balanced data	DL
Handling imbalance	Needs resampling	Learns patterns better	DL

🔥 Which Is Better Overall — ML or DL?



Best ML Model:

✓ Random Forest + Tomek Links



F1-Score: **0.95**
(higher than all
DL models)



Accuracy:
0.9731



Recall:
0.9995



Best DL Model:

✓ LSTM



F1-Score:
0.8787



Accuracy:
0.9802



Recall:
0.9994



Final Conclusion: ML Beats DL in This Project

Even though **LSTM** achieved the highest accuracy, the evaluation in fraud detection depends most on **F1-Score** due to the imbalanced nature of the data.

Random Forest + Tomek Links → **F1 = 0.95**

versus

Best DL (**LSTM**) → **F1 = 0.8787**

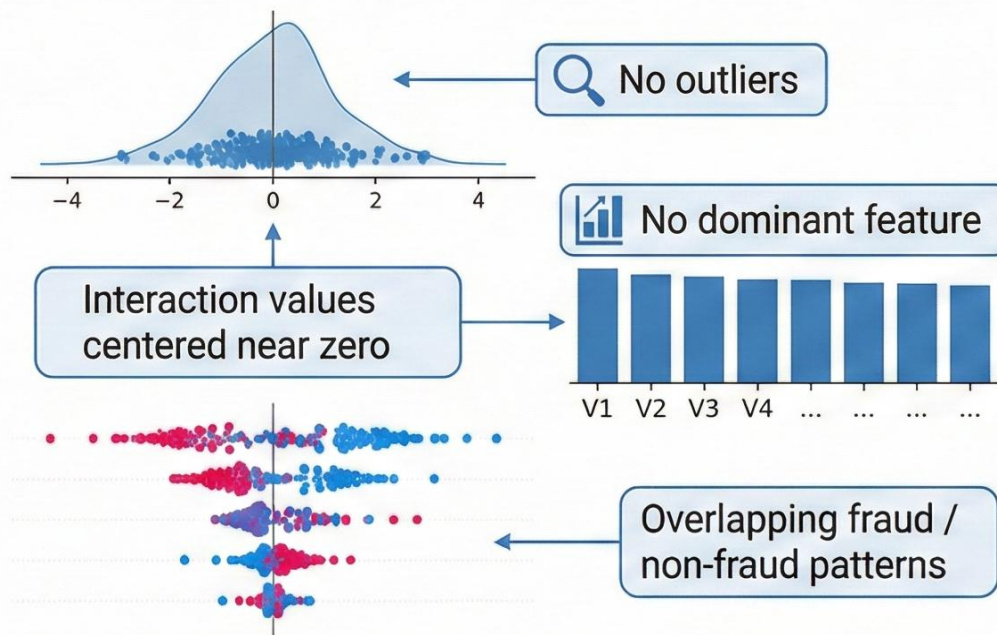
✓ **Machine Learning** outperformed Deep Learning in this project, because it delivered **better F1-Score, better overall balance, and stronger performance** under imbalance conditions. 🏆

Explainable AI Summary (SHAP + LIME)

XAI Method	Key Findings
SHAP	• Interaction values centered near zero
- No outliers	★ Stable & unbiased global behavior
- No dominant feature	
- Overlapping fraud / non-fraud patterns	
LIME	• Sample predicted as Non-Fraud (1.00)
- Fraud-supporting features: V ₄ , V ₂	★ Clear, interpretable local decision
- Non-fraud features: V ₁ , V ₁₇ , V ₁₄ , V ₁₃	

Explainable AI Results – SHAP & LIME

SHAP (Global Interpretation)



✓ **Global Conclusion:** Stable & unbiased model behavior

LIME (Local Interpretation)

✓ **Sample predicted as Non-Fraud (1.00)**



✓ **Local Conclusion:** Clear and interpretable prediction reasoning

Overall Result:



TRANSPARENT



TRUSTWORTHY



FULLY EXPLAINABLE

SHAP + LIME confirm that the model is transparent, trustworthy, and fully explainable

Conclusion

The project developed an effective fraud-detection framework using both ML and DL models on a highly imbalanced credit-card dataset.

ML models—especially Random Forest—showed strong precision and stable performance across resampling methods.

DL models—particularly LSTM—achieved higher overall accuracy and better generalization to unseen data.

XAI techniques (SHAP & LIME) provided clear explanations of model decisions, improving transparency and trust.

The combined approach of resampling, hybrid ML/DL modeling, and explainability significantly improved the detection of rare fraud cases while reducing false positives.

Overall, the framework enhances fraud-detection accuracy, supports better decision-making, and is suitable for real-world financial deployment.

Thank you for your
attention

