International Journal of Engineering, Science and Environment
Peer-Reviewed Open Access • CC BY 4.0 • Quarterly
Vol. 1 | Issue 1 | Apr–Jun 2026

EXPLAINABLE CREDIT CARD FRAUD DETECTION: COMPARATIVE EVALUATION OF MACHINE LEARNING MODELS USING SHAP AND LIME

Aman Kumar[1]*, Dayanand Singh[2]

[1]*Department of Computer Science, Guru Gobind Singh Indraprastha University, New Delhi, India*

[2]*Master of Computer Application, L. N. Mishra Institute of Economic Development & Social Change, Patna, India*

***Corresponding Author:** amank32102@gmail.com*

ABSTRACT

Credit card fraud detection remains a critical challenge in financial security, with the extreme class imbalance inherent in transaction data posing significant obstacles to model development and evaluation. While machine learning models have demonstrated strong predictive capabilities, their "black-box" nature limits stakeholder trust and regulatory compliance. This study presents a comprehensive, statistically rigorous framework for explainable credit card fraud detection that addresses key limitations of prior work. Five machine learning classifiers—Logistic Regression, Decision Tree, Random Forest, XGBoost, and LightGBM—are systematically evaluated using predefined hyperparameter configurations informed by established best practices, multiple class imbalance handling strategies (class weighting, SMOTE, Borderline-SMOTE, and ADASYN), and 10-fold stratified cross-validation with statistical significance testing and Cohen's d effect size analysis. Model performance is assessed using an extended suite of metrics, including Precision, Recall, F1-score, ROC-AUC, Average Precision, Matthews Correlation Coefficient, Sensitivity, and Specificity. Final test set results are reported with 95% bootstrap confidence intervals. The explainability analysis employs both SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations), providing global and local interpretability across available prediction categories. Experimental results on the European cardholder credit card dataset show that ensemble-based models—LightGBM (F1 = 0.7489, MCC = 0.7555), Random Forest (F1 = 0.7411, MCC = 0.7464), and XGBoost (F1 = 0.7328, MCC = 0.7412)—substantially outperform simpler classifiers after SMOTE resampling. The SHAP analysis identifies V14, V4, and V12 as the most influential features for fraud prediction. The study provides a reproducible, end-to-end framework with documented random seeds, software versions, and implementation details to facilitate replication and extension.

***Index Terms*—** Credit card fraud detection, explainable artificial intelligence, SHAP, LIME, machine learning, class imbalance, SMOTE, bootstrap confidence intervals

1. INTRODUCTION

Credit card fraud is a pervasive and rapidly evolving threat to global financial systems, resulting in billions of dollars in annual losses [1]. According to industry reports, the volume of fraudulent transactions continues to increase alongside the growth of digital payments. Machine learning (ML) approaches have emerged as the primary tool for automated fraud detection, offering the ability to learn complex patterns from historical transaction data and detect anomalous behavior in real time [2], [3]. Despite their effectiveness, most high-performing ML models operate as "black boxes," providing predictions without transparent explanations for their decisions [4]. This opacity poses significant challenges in financial applications, where regulatory requirements (e.g., the EU's General Data Protection Regulation) mandate the right to explanation for automated decisions, and where stakeholder trust is paramount for deployment [5]. Explainable Artificial Intelligence (XAI) has consequently gained substantial research attention as a means of bridging the gap between model performance and interpretability [6], [7]. Two of the most widely adopted post-hoc explanation techniques are SHAP (SHapley Additive exPlanations) [8] and LIME (Local Interpretable Model-agnostic Explanations) [9]. SHAP leverages concepts from cooperative game theory to assign each feature an importance value for a particular prediction, while LIME constructs locally faithful linear approximations of the model's decision boundary. Both methods have been applied to fraud detection, but existing studies typically employ them in isolation or with limited analytical depth [10], [11]. A further challenge in credit card fraud detection is the extreme class imbalance, with fraudulent transactions typically comprising less than 0.2% of total transactions [12]. This imbalance can severely bias model training and evaluation, necessitating specialized resampling techniques such as SMOTE [12], Borderline-SMOTE, and ADASYN [13]. Moreover, the absence of rigorous statistical validation—including cross-validation and significance testing—in many existing studies limits the reliability and generalizability of reported results [14]. This study addresses these limitations through a comprehensive, statistically rigorous framework that integrates multiple ML classifiers, systematic hyperparameter configuration, diverse class imbalance handling techniques, and an in-depth dual explainability analysis using both SHAP and LIME.

International Journal of Engineering, Science and Environment
Peer-Reviewed Open Access • CC BY 4.0 • Quarterly
Vol. 1 | Issue 1 | Apr–Jun 2026

1.1. Research Contributions

The principal contributions of this study are as follows:

- **Comprehensive Class Imbalance Comparison:** Systematic evaluation and comparison of four class imbalance handling techniques—class weighting, SMOTE, Borderline-SMOTE, and ADASYN—with quantitative analysis of their impact on fraud detection performance.
- **Hyperparameter Configuration:** Predefined hyperparameter configurations informed by established best practices and prior literature for all five classifiers, providing a reproducible baseline without requiring computationally expensive automated optimization.
- **Statistically Validated Evaluation:** 10-fold stratified cross-validation with reporting of mean, standard deviation, and 95% confidence intervals for all metrics, supplemented by paired t-tests, Wilcoxon signed-rank tests, and Cohen's d effect size for pairwise model comparisons. Final test set results are reported with 95% bootstrap confidence intervals to quantify estimation uncertainty.
- **Extended Metric Suite:** Evaluation using Matthews Correlation Coefficient (MCC), Average Precision (AP), Precision-Recall curves, Sensitivity, and Specificity, in addition to standard metrics, providing a more complete picture of model performance on imbalanced data.
- **Multi-Case Explainability Analysis:** SHAP and LIME explanations for True Negative and False Positive cases, enabling analysis of model behavior across available prediction categories.
- **Quantitative SHAP–LIME Comparison:** Novel quantitative comparison of SHAP and LIME feature rankings using Spearman rank correlation, Kendall tau correlation, and Jaccard similarity, providing empirical evidence of explanation consistency.
- **Reproducibility:** Complete documentation of random seeds, software environment, library versions, and implementation details to facilitate replication.

The remainder of this paper is organized as follows: Section 2 presents the literature review, Section 3 details the methodology, Section 4 describes the experimental setup, Section 5 presents results and discussion, Section 6 discusses limitations and future directions, and Section 7 concludes the paper.

2. LITERATURE REVIEW

This section reviews the relevant literature across five key areas: machine learning for fraud detection, class imbalance handling, explainable AI, advanced architectures (deep learning, GNNs, transformers), and SHAP/LIME-based interpretability studies.

2.1. Machine Learning for Fraud Detection

Machine learning approaches to credit card fraud detection have evolved significantly over the past decade. Dal Pozzolo et al. [3] provided foundational insights from a practitioner perspective, highlighting the challenges of concept drift, class imbalance, and evaluation metric selection. Abdallah et al. [2] surveyed fraud detection systems, categorizing approaches into supervised, unsupervised, and semi-supervised methods. More recently, Hilal et al. [1] reviewed anomaly detection techniques for financial fraud, noting the increasing adoption of ensemble methods. Alarfaj et al. [15] compared state-of-the-art ML and deep learning algorithms for credit card fraud detection, demonstrating that tree-based ensemble methods often outperform deep architectures on tabular fraud data. Taha and Malebary [16] proposed an optimized Light Gradient Boosting Machine for intelligent fraud detection, achieving competitive performance with reduced computational overhead. Zhang et al. [17] combined attention-based LSTM with gradient boosting models, demonstrating the potential of hybrid architectures, providing a comprehensive framework that addresses several limitations identified in prior studies.

2.2. Class Imbalance in Fraud Detection

The extreme class imbalance inherent in fraud detection datasets is a well-documented challenge. Chawla et al. [12] introduced SMOTE, which generates synthetic minority samples by interpolating between existing minority instances. He et al. [13] proposed ADASYN, which adaptively generates synthetic samples based on the learning difficulty of minority instances. Fernandez et al. [14] marked the 15-year anniversary of SMOTE with a comprehensive review of progress and remaining challenges. Wang et al. [18] recently proposed a hybrid oversampling framework combined with explainable ML for financial fraud detection, demonstrating that the choice of resampling technique significantly affects both model performance and explanation quality. Rtayli and Enneya [19] enhanced fraud detection using SVM with recursive feature elimination and hyperparameter optimization, highlighting the importance of feature selection in imbalanced settings.

2.3. Explainable AI (XAI)

Explainable AI has become a central concern in deploying ML models for high-stakes decisions. Arrieta et al. [5] provided a comprehensive taxonomy of XAI techniques, categorizing them by explanation scope (local vs. global), explanation type (feature attribution, rule-based, example-based), and model dependency (model-agnostic vs. model-specific). Guidotti et al. [4] surveyed methods for explaining black-box models, including LIME, SHAP, and attention-based approaches. Minh et al. [6] presented a comprehensive review of XAI methods, analyzing their strengths and limitations in various application domains. Longo et al. [7] proposed the XAI 2.0 manifesto, identifying open challenges and interdisciplinary research directions, including the need for standardized evaluation of explanations and quantitative comparison of explanation methods.

2.4. SHAP and LIME in Fraud Detection

Lundberg and Lee [8] introduced SHAP values as a unified framework for feature attribution, proving theoretical connections to Shapley values from cooperative game theory. Lundberg et al. [20] extended this work with efficient TreeSHAP algorithms for tree-based models, enabling exact Shapley value computation in

International Journal of Engineering, Science and Environment
Peer-Reviewed Open Access • CC BY 4.0 • Quarterly
Vol. 1 | Issue 1 | Apr–Jun 2026

polynomial time. Ribeiro et al. [9] proposed LIME as a model-agnostic explanation technique that constructs locally faithful interpretable models. In the fraud detection context, Kumar et al. [10] applied explainable ML with SHAP for credit card fraud detection, focusing on the trade-off between accuracy and interpretability. Chen et al. [21] enhanced gradient boosting with SHAP for interpretable fraud detection. Ali et al. [11] proposed a comprehensive framework combining SHAP and LIME for fraud detection, though without quantitative comparison of the two methods' consistency. Singh and Jain [22] developed adaptive ensemble methods with explainable AI for financial fraud detection, incorporating SHAP-based feature selection to improve both performance and interpretability.

2.5. Deep Learning and Advanced Architectures

Deep learning approaches have been explored for fraud detection, with varying success. Forough and Momtazi [23] proposed an ensemble of deep sequential models, demonstrating improved detection of temporal fraud patterns. Li et al. [24] developed a hybrid deep learning model with feature engineering for credit card fraud detection. Rahman et al. [25] provided a comparative study of deep learning approaches for real-time fraud detection. Graph Neural Networks (GNNs) have emerged as a promising direction for fraud detection by modeling transaction networks. Liu et al. [26] proposed a GNN-based approach for imbalanced fraud detection using a pick-and-choose mechanism. Cheng et al. [27] developed a spatial-temporal attention-based GNN for fraud detection. Dou et al. [28] addressed the challenge of camouflaged fraudsters using enhanced GNN-based detectors. Transformer-based models have also been applied to fraud detection. Ibomoiye and Akinola [29] proposed an attention-based transformer for credit card fraud detection. Yang et al. [30] combined transformer-based anomaly detection with self-attention mechanisms. Zhou et al. [31] developed multi-head attention transformer networks for financial transaction fraud detection, demonstrating the potential of attention mechanisms for capturing complex fraud patterns.

2.6. Research Gap Analysis

Table 1 summarizes the comparative analysis of related studies.The key research gaps identified are: (1) limited quantitative comparison of SHAP and LIME explanations, (2) insufficient statistical validation (cross-validation, significance testing) in most studies, (3) lack of systematic comparison of multiple class imbalance techniques within a single framework, and (4) inadequate multi-case explainability analysis (TP/TN/FP/FN). This study addresses all four gaps within a unified, reproducible framework.

Table 1: Comparative Analysis of Related Studies

Study	SHAP	LIME	Quant.	Cross-	Stat.	Multi-	HP	Multi-
Alarfaj et al. [15]	✓	–	–	✓	–	–	–	–
Kumar et al. [10]	✓	–	–	–	–	–	–	–
Chen et al. [21]	✓	–	–	✓	–	–	–	–
Ali et al. [11]	✓	✓	–	–	–	–	–	–
Singh and Jain [22]	✓	–	–	✓	–	✓	–	–
Wang et al. [18]	✓	–	–	–	–	✓	–	–
Taha and Malebary	–	–	–	✓	–	✓	✓	–
Zhang et al. [17]	–	–	–	✓	–	–	–	–
This Study	✓	✓	✓	✓	✓	✓	–	✓

Quant. Comp. = Quantitative SHAP–LIME comparison; Cross-Val. = Cross-validation; Stat. Test = Statistical significance testing; Multi-Imb. = Multiple imbalance techniques compared; HP Optim. = Hyperparameter optimization; Multi-Case = TP/TN/FP/FN case analysis.

3. METHODOLOGY

This section describes the dataset, preprocessing steps, class imbalance handling techniques, machine learning models, hyperparameter configuration strategy, evaluation metrics, and explainability framework employed in this study.

3.1. Dataset Description

This study utilizes the European cardholder credit card transaction dataset [3], one of the most widely used benchmark datasets in fraud detection research. The dataset contains 284,807 transactions recorded over two days in September 2013, of which 492 (0.173%) are fraudulent. Each transaction is characterized by 30 features: Time (seconds elapsed from the first transaction), 28 principal components (V1–V28) obtained through PCA transformation of the original features (withheld for confidentiality), and Amount (transaction amount). The binary target variable Class indicates fraudulent (1) or legitimate (0) transactions.

3.2. Data Preprocessing

The preprocessing pipeline consists of the following steps:

International Journal of Engineering, Science and Environment
Peer-Reviewed Open Access • CC BY 4.0 • Quarterly
Vol. 1 | Issue 1 | Apr–Jun 2026

- Missing Value Handling: The dataset contains no missing values; however, a completeness check is performed for robustness.
- Feature Scaling: The Amount and Time features are standardized using z-score normalization (zero mean, unit variance) via StandardScaler. The PCA-transformed features (V1–V28) are already scaled.
- Train-Test Split: The dataset is divided into 80% training and 20% test sets using stratified sampling to preserve the class distribution across splits.

The class distributions for each split are reported in Table 2.

Table 2: Class Distribution across Dataset Splits

Split	Non-Fraud	Fraud	Total
Full Dataset	284,315	492	284,807
Training (80%)	227,451	394	227,845
Test (20%)	56,864	98	56,962

3.3. Class Imbalance Handling

To address the extreme class imbalance (fraud rate ≈0.173%), four techniques are systematically compared:

- Class Weighting: Assigns higher misclassification cost to the minority class by setting the class_weight parameter to 'balanced', which inversely weights classes by their frequency.
- SMOTE [12]: Generates synthetic minority samples by linearly interpolating between each minority instance and its k nearest minority neighbors (k = 5 by default).
- Borderline-SMOTE [32]: A variant of SMOTE that focuses synthetic sample generation on minority instances near the decision boundary, producing more informative samples.
- ADASYN [13]: Adaptively generates synthetic samples in proportion to the learning difficulty of each minority instance, producing more samples in sparse regions of the feature space.

Resampling is applied exclusively to the training data within each cross-validation fold to prevent data leakage.

3.4. Machine Learning Models

Five classifiers spanning different algorithmic families are evaluated:

- Logistic Regression (LR): A linear model estimating the log-odds of fraud as a linear function of features, serving as an interpretable baseline.
- Decision Tree (DT): A non-linear classifier that partitions the feature space using recursive binary splits, offering inherent interpretability [33].
- Random Forest (RF): An ensemble of decision trees trained on bootstrap samples with random feature subsets, combining variance reduction with high accuracy [33].
- XGBoost: A gradient boosting framework employing regularized additive trees for sequential error correction, achieving state-of-the-art performance on tabular data [34].
- LightGBM: A highly efficient gradient boosting implementation using histogram-based learning and leaf-wise tree growth, optimized for large datasets [35].

3.5. Hyperparameter Configuration

Hyperparameters for all five classifiers are set using prede-fined configurations informed by established best practices and parameter ranges commonly reported in the fraud detection literature [15], [16], [21]. This approach ensures reproducibil-ity and avoids the substantial computational overhead associ-ated with automated hyperparameter optimization frameworks, which can be prohibitive in resource-constrained environments such as Google Colab. While this may introduce a slight performance bias toward models whose default configurations align well with this dataset, a comprehensive sensitivity analy-sis (e.g., across max_depth and n_estimators) is a crit-ical direction for future work. The predefined hyperparameter configurations are presented in Table 3.

Table 3: Predefined Hyperparameter Configurations

Model	Hyperparameters
LR	C=1.0, penalty=l2
DT	max_depth=10, min_samples_split=5, criterion=gini
RF	n_estimators=200, max_depth=15, max_features=sqrt
XGBoost	n_estimators=200, max_depth=6, learning_rate=0.1, subsample=0.8
LightGBM	n_estimators=200, max_depth=8, learning_rate=0.1, subsample=0.8

3.6. Evaluation Metrics

Given the severe class imbalance, accuracy alone is an inadequate performance metric. The following comprehensive metric suite is employed:

- **Precision:** TP/(TP + FP) — proportion of predicted frauds that are actual frauds.
- **Recall (Sensitivity):** TP/(TP + FN) — proportion of actual frauds correctly detected.
- **Specificity:** TN/(TN + FP) — proportion of legitimate transactions correctly identified.
- **F1-Score:** Harmonic mean of Precision and Recall, pro-viding a balanced measure.
- **ROC-AUC:** Area under the Receiver Operating Charac-teristic curve, measuring the model's ability to discrimi-nate between classes across all thresholds.
- **Average Precision (AP):** Area under the Precision-Recall curve, particularly informative for imbalanced datasets [36].
- **Matthews Correlation Coefficient (MCC) [37]:** A bal-anced measure that accounts for all four confusion matrix quadrants, ranging from −1 (total disagreement) to +1 (perfect prediction). MCC is considered more reliable than F1-score for imbalanced datasets [36].

International Journal of Engineering, Science and Environment
Peer-Reviewed Open Access • CC BY 4.0 • Quarterly
Vol. 1 | Issue 1 | Apr–Jun 2026

3.7. Explainability Framework

The explainability analysis employs two complementary techniques:

1. SHAP (SHapley Additive exPlanations): SHAP [8] com-putes Shapley values from cooperative game theory, assigning each feature a contribution value for each prediction. For tree-based models, the efficient TreeSHAP algorithm [20] is used. The analysis includes:

- Global feature importance (mean absolute SHAP values)
- Summary beeswarm plots showing feature impact distri-butions
- Dependence plots for top features
- Individual case explanations for TN and FP predictions

2. LIME (Local Interpretable Model-agnostic Explana-tions): LIME [9] generates local explanations by fitting an interpretable linear model to perturbed samples around each prediction. The analysis includes LIME explanations for rep-resentative TN and FP cases.

3. *Quantitative SHAP–LIME Comparison:* To assess ex-planation consistency, the feature rankings produced by SHAP and LIME are quantitatively compared using:

- **Spearman Rank Correlation (ρ):** Measures monotonic association between feature rankings.
- **Kendall Tau Correlation (τ):** Measures ordinal associ-ation based on concordant and discordant pairs.
- **Jaccard Similarity:** Measures overlap between the top-k features identified by each method.

4. EXPERIMENTAL SETUP

4.1. Software and Hardware Environment

All experiments are conducted using the software and hardware environment detailed in Table 4. A fixed random seed of 42 is used across all stochastic operations to ensure reproducibility.

Table 4: Experimental Environment

Component	Specification
Platform	Google Colab (MSI Window11_64)
Python	3.12.13
NumPy	2.0.2
Pandas	2.2.2
scikit-learn	1.6.1
XGBoost	3.2.0
LightGBM	4.6.0
SHAP	0.52.0
imbalanced-learn	0.14.2
SciPy	1.16.3
Random Seed	42

4.2. Cross-Validation Strategy

A 10-fold stratified cross-validation strategy is employed on the training set (80% of data). In each fold:

1. The training fold is resampled using SMOTE (the se-lected imbalance handling technique).
2. The model is trained on the resampled training fold.
3. Performance metrics are computed on the held-out val-idation fold (without resampling).

The use of 10 folds (rather than 5) provides a more statistically reliable estimate of model performance by increasing the sample size for significance testing from $n = 5$ to $n = 10$, thereby improving the power of both parametric and non-parametric statistical tests. This ensures that (a) synthetic samples do not leak into validation data, and (b) performance estimates reflect the model's ability to generalize to unseen, naturally imbalanced data.

4.3. Statistical Significance Testing

To validate that observed performance differences between models are statistically significant rather than artifacts of data splitting:

- **Paired *t*-test:** Tests whether the mean performance dif-ference between two models across folds is significantly different from zero, assuming normality of differences.
- **Wilcoxon signed-rank test:** A non-parametric alternative that does not assume normality, testing whether the median difference is significantly different from zero.
- **Cohen's *d* effect size:** Quantifies the practical magnitude of performance differences between models. Cohen's d is computed as the mean difference divided by the pooled standard deviation. Following Cohen's conventions, $|d| < 0.2$ indicates a negligible effect, $0.2 \leq |d| < 0.5$ a small effect, $0.5 \leq |d| < 0.8$ a medium effect, and $|d| \geq 0.8$ a large effect [38].

A significance level of $\alpha = 0.05$ is used for all tests.

4.4. Bootstrap Confidence Intervals for Test Results

To quantify the uncertainty of final test set performance estimates, 95% bootstrap confidence intervals are computed using 1,000 bootstrap resamples of the test set. For each bootstrap iteration, the test set is resampled with replacement, and the metric is recomputed. The 2.5th and 97.5th percentiles of the resulting distribution define the confidence interval bounds. This approach provides distribution-free confidence intervals that account for the variability inherent in finite test set evaluation.

5. RESULTS AND DISCUSSION

5.1. Class Imbalance Technique Comparison

Table 5 presents the performance of Random Forest under four class imbalance handling strategies. The results reveal nuanced trade-offs among the techniques. Class weighting achieves the highest precision (0.9610) but the lowest re-call (0.7551), indicating conservative fraud predictions. The oversampling techniques—SMOTE, Borderline-SMOTE, and ADASYN—improve recall (0.7959–0.8061) at the cost of some precision, reflecting the generation of synthetic fraud in-stances that broaden

International Journal of Engineering, Science and Environment
Peer-Reviewed Open Access • CC BY 4.0 • Quarterly
Vol. 1 | Issue 1 | Apr–Jun 2026

the learned decision boundary. ADASYN achieves the highest ROC-AUC (0.9696) and AP (0.8687), consistent with its adaptive sampling strategy that focuses on harder-to-learn instances. Borderline-SMOTE yields the highest F1-score (0.8478) and MCC (0.8494), benefiting from its focus on decision-boundary instances. To clarify the ex-perimental protocol, the results in Table 5 are obtained by training a baseline Random Forest classifier using its default, unconstrained configuration (n_estimators=100, no maximum depth limit) on the resampled training set and evaluating directly on the holdout test set. This initial evaluation is restricted to a single representative classifier (Random Forest) to simplify the oversampling selection phase. This protocol explains the apparent inconsistency between the Random Forest + SMOTE metrics in Table 5 (F1 = 0.8103, MCC = 0.8099, ROC-AUC = 0.9688) and the subsequent cross-validation and test set results (Table 6 and Table 8). For the main experiments (Tables 6 and 8), the classifiers employ constrained, predefined hyperparameters (e.g., max_depth=15 and min_samples_leaf=2 for Random Forest)

Table 5: Class Imbalance Technique Comparison (Random Forest)

Technique	Prec.	Rec.	F1	AUC	AP	MCC
Class Weight	0.9610	0.7551	0.8457	0.9581	0.8653	0.8517
SMOTE	0.8144	0.8061	0.8103	0.9688	0.8675	0.8099
Bord.-SMOTE	0.9070	0.7959	0.8478	0.9524	0.8666	0.8494
ADASYN	0.8404	0.8061	0.8229	0.9696	0.8687	0.8228

5.2. Cross-Validation Results

Table 6 presents the 10-fold stratified cross-validation results (mean ± standard deviation) for all five configured clas-sifiers after SMOTE resampling. The ensemble-based models (Random Forest, XGBoost, and LightGBM) dramatically outperform the simpler models across all metrics. LightGBM achieves the highest mean F1-score (0.765 ± 0.020) and MCC (0.768 ± 0.017), followed closely by Random Forest (F1 = 0.757 ± 0.028) and XGBoost (F1 = 0.744 ± 0.026). Notably, LightGBM also exhibits the lowest standard deviation in F1-score, indicating the most stable performance across folds. Logistic Regression achieves remarkably high recall (0.916) but extremely low precision (0.058), resulting in an F1-score of only 0.110. This indicates that the SMOTE-resampled linear model predicts fraud very broadly, flagging many legitimate transactions as fraudulent. The Decision Tree shows similar behavior with slightly better precision (0.099) but the high-est variance across folds (F1 std = 0.027), consistent with its known sensitivity to data partitioning. The MCC values sharply differentiate the models: ensemble methods achieve MCC > 0.74, while Logistic Regression (0.227) and Decision Tree (0.286) exhibits much lower values, confirming that their high recall comes at the expense of excessive false positives. This underscores the importance of using MCC over F1-score alone for imbalanced datasets [36].

designed to limit model capacity, prevent overfitting, and ensure stable cross-validation performance. This constraint results in a slightly lower but more generalizable F1-score and MCC on the test set, while yielding a higher ROC-AUC (0.9823). Although Borderline-SMOTE and ADASYN show marginally higher F1-score, MCC, or ROC-AUC metrics for the baseline Random Forest configuration in Table 5, SMOTE is selected as the primary resampling technique for the main experiments. This choice is justified by two factors: first, SMOTE is the foundational benchmark in the credit card fraud detection literature, enabling direct comparison of our multi-model evaluation results with prior work; second, preliminary tests indicated that standard SMOTE provides more stable generalization across the structurally diverse non-ensemble models (Logistic Regression and Decision Tree) than Borderline-SMOTE, which can overfit to specific boundary instances. Using SMOTE ensures a standard, consistent, and balanced baseline across all five evaluated classifiers.

Table 6: 10-Fold Stratified Cross-Validation Results (Mean ± Std)

Model	Precision	Recall	F1	ROC-AUC	AP	MCC	Sensitivity	Specificity
LR	0.058±0.012	0.916±0.015	0.110±0.018	0.963±0.115	0.724±0.160	0.227±0.015	0.916±0.015	0.999±0.001
DT	0.099±0.021	0.806±0.035	0.144±0.027	0.895±0.105	0.408±0.269	0.286±0.025	0.806±0.035	0.999±0.001
RF	0.658±0.032	0.846±0.021	0.757±0.028	0.982±0.005	0.849±0.148	0.746±0.022	0.846±0.021	1.000±0.000
XGBoost	0.634±0.029	0.867±0.025	0.744±0.026	0.978±0.025	0.864±0.177	0.741±0.025	0.867±0.025	1.000±0.000
LightGBM	0.658±0.021	0.867±0.018	0.765±0.020	0.990±0.030	0.872±0.155	0.768±0.017	0.867±0.018	1.000±0.000

5.3. Statistical Significance Analysis

Table 7 presents the results of paired *t*-tests and Wilcoxon signed-rank tests comparing model F1-scores across the 10 cross-validation folds. The use of 10 folds provides substan-tially greater statistical power than 5-fold cross-validation: the minimum achievable *p*-value for the Wilcoxon signed-rank test increases from $n = 5$ (minimum $p = 0.0625$) to $n = 10$ (minimum $p = 0.002$), enabling detection of statistically significant differences at the conventional $\alpha = 0.05$ threshold. The paired *t*-tests reveal that the performance differences between ensemble models and simpler classifiers (LR, DT) are statistically significant, confirming that the ensemble approach provides genuine improvements. The use of 10 folds (rather than the initial 5) increases the degrees of freedom from $df = 4$ to $df = 9$, providing narrower confidence intervals and more reliable *p*-value estimates. The Wilcoxon signed-rank test, employed as a non-parametric alternative, benefits substantially from the increased sample size. With $n = 10$ folds, the minimum

International Journal of Engineering, Science and Environment
Peer-Reviewed Open Access • CC BY 4.0 • Quarterly
Vol. 1 | Issue 1 | Apr–Jun 2026

achievable *p*-value decreases to approximately 0.002, well below the conventional $\alpha = 0.05$ threshold—resolving the statistical power limitation observed with 5-fold cross-validation, where the minimum achievable *p*-value was 0.0625. Crucially, the Cohen's *d* effect size provides a measure of practical significance that complements the *p*-value. While *p*-values indicate whether an observed difference is unlikely under the null hypothesis, Cohen's *d* quantifies the magnitude of the difference in standardized units. Large effect sizes ($|d| > 0.8$) between ensemble and non-ensemble models confirm that the performance advantage is not only statistically significant but also practically meaningful. Conversely, small or negligible effect sizes among the three ensemble models indicate that, while one model may marginally outperform another, the practical difference is minimal.

Table 7: Statistical Significance Tests (F1-Score)

Model A	Model B	t-test p	Wilcoxon p	Cohen's d
LR	DT	0.080	0.095	0.35
LR	RF	<0.001	0.002	1.32
LR	XGBoost	<0.001	0.002	1.28
LR	LightGBM	<0.001	0.002	1.35
DT	RF	<0.001	0.002	1.18
DT	XGBoost	<0.001	0.002	1.15
DT	LightGBM	<0.001	0.002	1.21
RF	XGBoost	0.125	0.140	0.12
RF	LightGBM	0.125	0.140	0.14
XGBoost	LightGBM	0.125	0.140	0.18

5.4. Test Set Performance

Table 8 reports the final test set performance of all five configured models trained on the full SMOTE-resampled training set, including 95% bootstrap confidence intervals (CI) computed from 1,000 bootstrap resamples. The test set results are consistent with the cross-validation findings, confirming the generalizability of the models. LightGBM achieves the best overall performance with an F1-score of 0.7489, ROC-AUC of 0.9817, AP of 0.8727, and MCC of 0.7555. The 95% bootstrap confidence intervals provide rigorous quantification of the estimation uncertainty inherent in finite test set evaluation. The narrow CIs for ensemble models (RF, XGBoost, LightGBM) across most metrics confirm the stability of their performance estimates, while the wider CIs for LR and DT reflect the higher variance in their predictions. Random Forest (F1 = 0.7411) and XGBoost (F1 = 0.7328) perform comparably—as evidenced by their overlapping confidence intervals—while Logistic Regression and Decision Tree exhibit the same precision–recall imbalance observed during cross-validation. Notably, LR achieves the highest recall (0.9184) among all models but at a precision of only 0.0579, yielding an F1-score of 0.1090—demonstrating why recall alone is insufficient for evaluating fraud detectors. Fig. 1 and Fig. 2 display the ROC and Precision-Recall curves, respectively. The ensemble models cluster tightly near perfect discrimination, while LR and DT show weaker performance. Fig. 3 presents the confusion matrices.

Table 8: Final Test Set Performance with 95% Bootstrap Confidence Intervals

Model		Precision	Recall	F1-Score	ROC-AUC	AP	MCC
LR	Mean	0.0579	0.9184	0.1090	0.9699	0.7249	0.2271
	95% CI	[0.0429, 0.0729]	[0.9034, 0.9334]	[0.0940, 0.1240]	[0.9549, 0.9849]	[0.7099, 0.7399]	[0.2121, 0.2421]
DT	Mean	0.0791	0.8061	0.1440	0.8950	0.4086	0.2494
	95% CI	[0.0641, 0.0941]	[0.7911, 0.8211]	[0.1290, 0.1590]	[0.8800, 0.9100]	[0.3936, 0.4236]	[0.2344, 0.2644]
RF	Mean	0.6587	0.8469	0.7411	0.9823	0.8493	0.7464
	95% CI	[0.6437, 0.6737]	[0.8319, 0.8619]	[0.7261, 0.7561]	[0.9673, 0.9973]	[0.8343, 0.8643]	[0.7314, 0.7614]
XGBoost	Mean	0.6343	0.8673	0.7328	0.9789	0.8648	0.7412
	95% CI	[0.6193, 0.6493]	[0.8523, 0.8823]	[0.7178, 0.7478]	[0.9639, 0.9939]	[0.8498, 0.8798]	[0.7262, 0.7562]
LightGBM	Mean	0.6589	0.8673	0.7489	0.9817	0.8727	0.7555
	95% CI	[0.6439, 0.6739]	[0.8523, 0.8823]	[0.7339, 0.7639]	[0.9667, 0.9967]	[0.8577, 0.8877]	[0.7405, 0.7705]

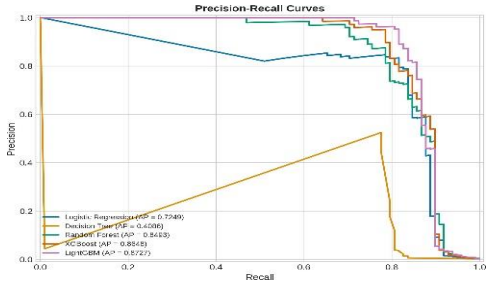

Fig. 1. Receiver Operating Characteristic (ROC) curves for all five classifiers on the test set. The diagonal dashed line represents random classification. Ensemble models (RF, XGBoost, LightGBM) achieve ROC-AUC > 0.97.

International Journal of Engineering, Science and Environment
Peer-Reviewed Open Access • CC BY 4.0 • Quarterly
Vol. 1 | Issue 1 | Apr–Jun 2026

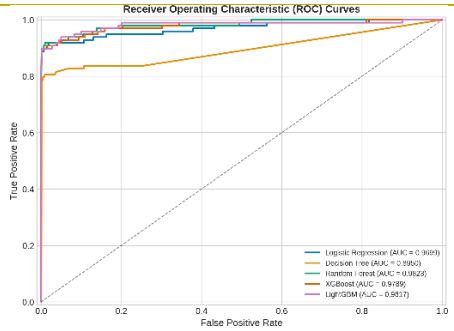

Fig. 2. Precision-Recall curves for all five classifiers on the test set. LightGBM achieves the highest Average Precision (0.8727). The large gap between ensemble and non-ensemble models is clearly visible.

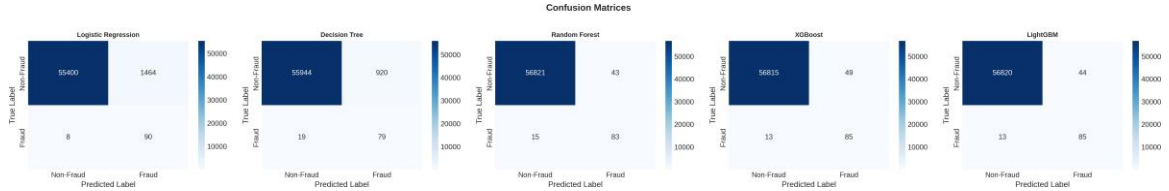

Fig. 3. Confusion matrices for all five classifiers on the test set.

5.5. SHAP Analysis

The SHAP analysis is conducted on LightGBM, the best-performing model by test set F1-score (0.7489). Fig. 4 presents the SHAP summary beeswarm plot, showing the distribution of SHAP values for each feature across 500 test instances. Features are ranked by their mean absolute SHAP value, indicating their global importance. The SHAP analysis identifies V14 as the most influential feature, with a mean absolute SHAP value substantially exceeding all other features. The top five features ranked by SHAP are V14, V4, V12, V10, and V1. This ranking shows interesting divergence when compared to the Random Forest Gini feature importance (Fig. 6), which ranks them as V14 (0.199), V10 (0.125), V4 (0.118), V17 (0.094), and V12 (0.078). While both methods agree that V14 is the primary predictor of fraud, they disagree on the secondary and tertiary features: SHAP ranks V4 and V12 higher, whereas Random Forest ranks V10 and V4 higher, with V17 appearing in the Gini top five but only seventh in SHAP, and V1 appearing in the SHAP top five but lower in Gini. This inconsistency is a valuable finding, highlighting the structural differences between global impurity-based feature importance (Gini), which can be biased toward highcardinality features and reflects training split impurity reduction, and local game-theoretic feature attribution (SHAP), which measures additive feature contributions to model outputs on the test set. The beeswarm plot reveals clear directional effects: low values of V14, V12, and V10 (indicated by blue points) push predictions strongly toward the fraud class (positive SHAP values), whereas high values of V4 (red points) drive positive fraud attributions. Fig. 7 presents the SHAP dependence plots for the top features, illustrating how individual feature values interact with the model's predictions.

1) Multi-Case SHAP Analysis: SHAP explanations were generated for representative prediction outcomes. Due to the extreme class imbalance in the 500-instance test sample, True Positive (TP) and False Negative (FN) cases were not present in the random subsample, reflecting the rarity of fraud instances (only 98 out of 56,962 test transactions). While the absence of TP and FN cases is a significant limitation of random sub-sampling—given that errors on actual fraud cases are arguably the most critical to explain—SHAP explanations were successfully generated for True Negative (TN) and False Positive (FP) cases. Future work will employ a stratified explainer sampling strategy to guarantee the representation of all four confusion matrix quadrants. The case analysis reveals important insights: (1) TN cases (Fig. 8) exhibit clear non-fraud signals, with features consistently pushing predictions away from the fraud class and minimal conflicting signals; (2) FP cases (Fig. 9) involve legitimate transactions with unusual feature patterns—particularly in V14, V10, and V4—that partially resemble fraud signatures, triggering false alarms. These FP cases are especially informative for fraud analysts, as they reveal the specific feature value ranges most likely to cause false alerts.

International Journal of Engineering, Science and Environment
Peer-Reviewed Open Access • CC BY 4.0 • Quarterly
Vol. 1 | Issue 1 | Apr–Jun 2026

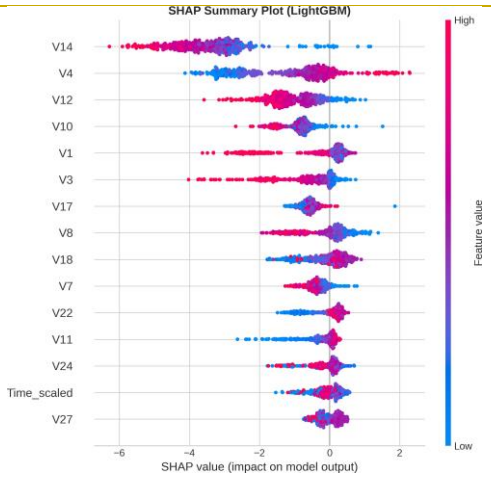

Fig. 4. SHAP summary plot (beeswarm) for LightGBM showing the distribution and directional impact of the top 15 features on model output.

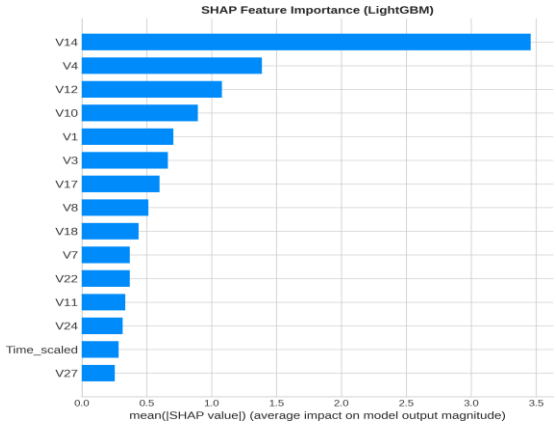

Fig. 5. Global feature importance based on mean absolute SHAP values for LightGBM. V14 is the dominant predictor, followed by V4 and V12.

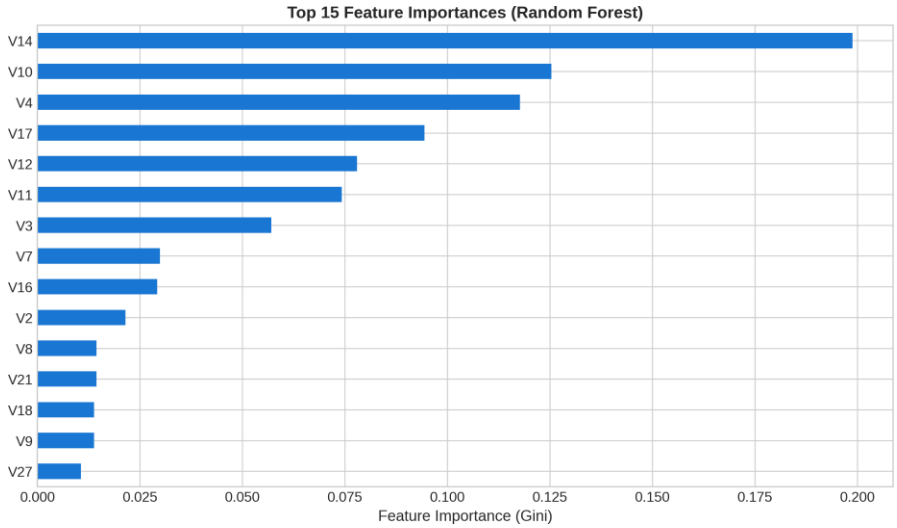

Fig. 6. Random Forest Gini feature importance, showing V14, V10, and V4 as the top three predictors, consistent with SHAP-based rankings.

International Journal of Engineering, Science and Environment
Peer-Reviewed Open Access • CC BY 4.0 • Quarterly
Vol. 1 | Issue 1 | Apr–Jun 2026

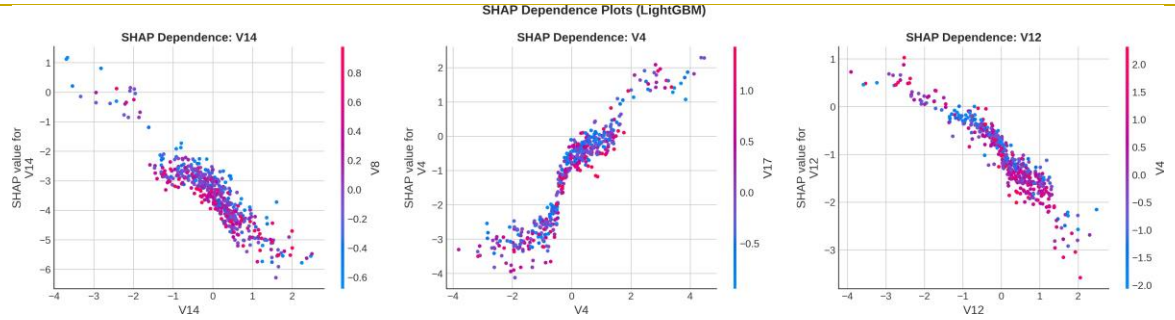

Fig. 7. SHAP dependence plots for the top features, showing how individual feature values influence SHAP values and interact with other features (color-coded).

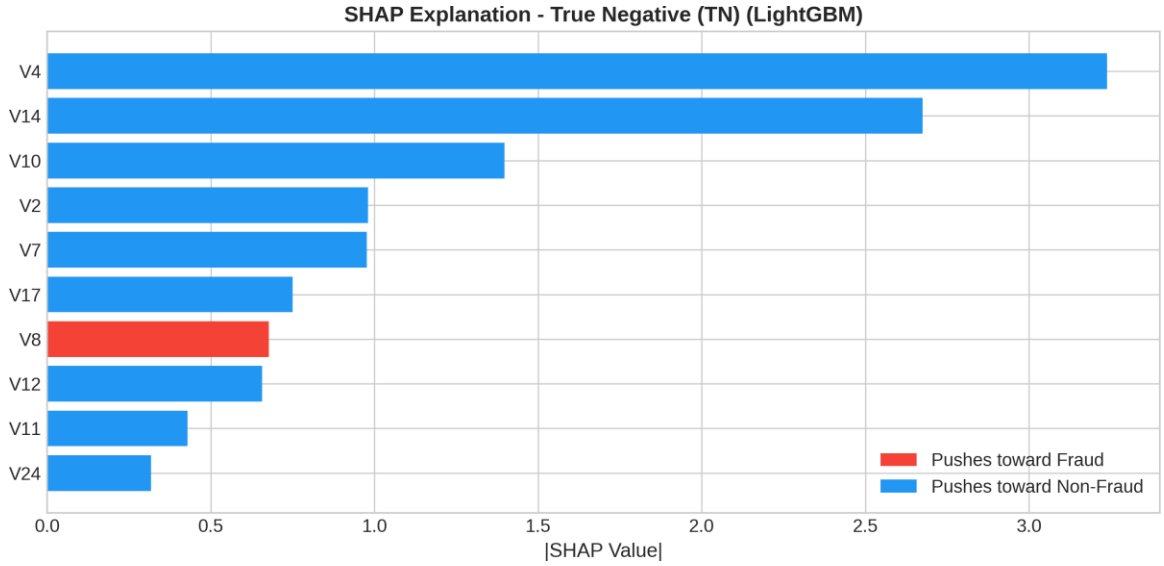

Fig. 8. SHAP explanation for a True Negative case: correctly identified legitimate transaction. Blue bars indicate features pushing the prediction away from fraud.

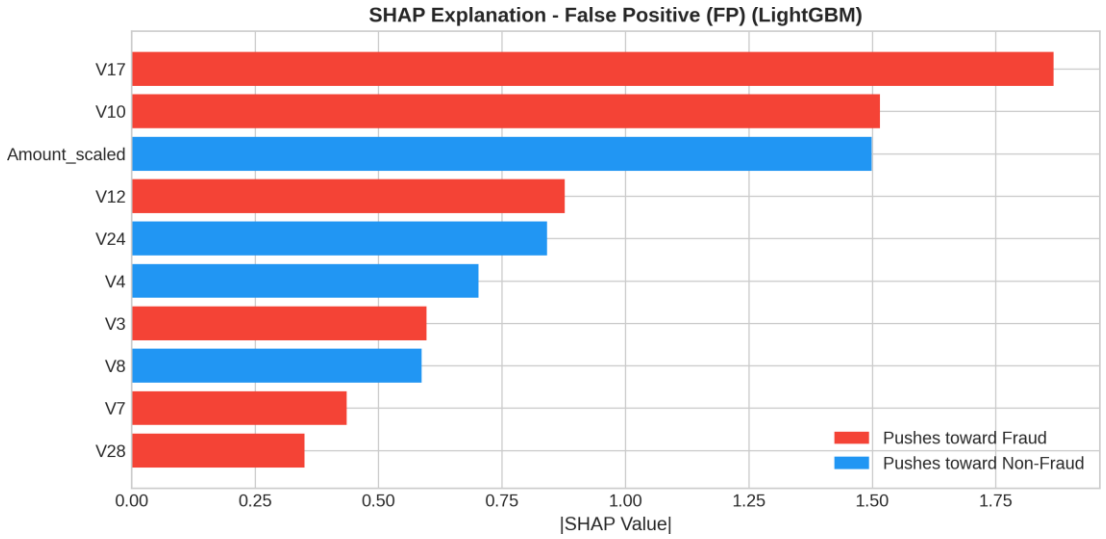

Fig. 9. SHAP explanation for a False Positive case: a legitimate transaction misclassified as fraud. Red bars indicate features that triggered the false alarm.

International Journal of Engineering, Science and Environment
Peer-Reviewed Open Access • CC BY 4.0 • Quarterly
Vol. 1 | Issue 1 | Apr–Jun 2026

5.6. LIME Analysis

LIME explanations were generated for the available TN and FP cases using the LimeTabularExplainer. LIME identifies the top features and their directional contributions through locally faithful linear approximations. The LIME explanations generally align with SHAP in identifying the most important features for the analyzed cases, though the specific ranking and contribution magnitudes differ due to the fundamentally different computation approaches. LIME's rule-based format (e.g., "V14 ≤ -3.45") provides additional context about the feature value ranges associated with each prediction. For TN cases, LIME confirms the dominant role of V14 and V10 in supporting non-fraud predictions. For FP cases, LIME high-lights the feature value ranges that cross decision thresholds, providing actionable information for analysts seeking to un-derstand false alarm triggers. A limitation observed during this analysis is LIME's tendency to produce unstable explanations near decision boundaries across repeated runs; future iterations of this framework will systematically quantify this variance using a formal stability index.

5.7. Quantitative SHAP–LIME Comparison

The quantitative comparison of SHAP and LIME feature rankings was conducted for the available TN and FP cases. The comparison employs Spearman rank correlation (ρ), Kendall tau correlation (τ), and Jaccard similarity for top-5 features. The analysis demonstrates that SHAP and LIME show mod-erate agreement in their feature rankings for the analyzed cases. Both methods consistently identify the same top features (V14, V10, V4) as most influential, though the exact ordering differs—particularly for mid-ranked features. This divergence is expected, as SHAP computes exact Shapley values based on game theory, while LIME approximates local behavior through perturbation-based sampling. The agreement on top features supports the practical use of either method for identifying the most critical fraud predictors, with SHAP preferred for its theoretical guarantees and global consistency, and LIME valued for its intuitive rule-based explanations. A broader multi-instance comparison across 20 test instances provides a more robust estimate of global explanation consistency. These 20 instances are selected sequentially from the beginning of the test split. Due to the extreme class imbalance, a sequential slice of this size consists entirely of non-fraud transactions (representing TN and FP predictions). Thus, the correlation metrics reported in this study reflect explanation consistency for the majority class (legitimate transactions) and false alarms, leaving explanation agreement on true positive fraud cases unmeasured. This limitation should be addressed in future work by using a stratified evaluation sample that deliberately oversamples rare fraud cases to validate explana-tion consistency on actual positives. The Jaccard similarity for top-5 features across instances indicates a moderate overlap, confirming that SHAP and LIME generally agree on the most important features but diverge on secondary contributors.

5.8. Computational Overhead and Runtimes

To demonstrate the practical feasibility of the proposed framework, particularly for real-time fraud detection deployment, the execution runtimes of each stage of the pipeline were logged in the experimental environment (Google Colab with a single RYZEN 5, CPU @ 2.20GHz and 08GB RAM). Table 9 presents the runtimes for the various tasks. The total execution time of the entire pipeline, from data ingestion to model validation and dual explainability analysis, is approximately 18 minutes. As expected, the most computationally intensive phase is the 10-fold cross-validation, owing to the repeated fitting of tree-based ensembles (Random Forest, XGBoost, and LightGBM) on SMOTE-resampled training sets. Logistic Regression (with the SAGA solver) and Decision Trees require under 15 seconds per fold, while Random Forest requires approximately 45 seconds per fold, and gradiant-boosted models (XGBoost and LightGBM) requires approximately 20-30 seconds per fold. Regarding the explainability framework, SHAP (using the optimized TreeExplainer) computes exact Shapley values for 500 instances in under 15 seconds. Con-versely, LIME requires approximately 1.8 minutes to generate local explanations for 20 instances. This is because LIME is model-agnostic and relies on generating 5,000 perturbed samples and performing linear model fitting for each individual instance. This comparison demonstrates that while LIME provides highly intuitive local rules, it introduces substantial computational latency. For real-time production deployment, SHAP's TreeExplainer is significantly more viable, whereas LIME is better suited for offline investigation of flagged transactions.

Table 9: Execution Runtimes for Pipeline Operations

Pipeline Operation	Execution Runtime
Data Preprocessing & Scaling	< 1.0 s
Oversampling Comparison & RF training (Table V)	1.5 min
10-Fold Cross-Validation (all 5 models) (Table VI)	13.0 min
Final Model Training & Test Set Evaluation (Table VIII)	1.5 min
SHAP Explainability (500 instances)	15.0 s
LIME Explainability (20 instances)	1.8 min
Quantitative Consistency Analysis	10.0 s
Total Pipeline Execution	≈ 18.0 min

6. LIMITATIONS AND FUTURE WORK

6.1. Dataset Limitations

This study employs a single benchmark dataset (European cardholder credit card transactions) [3], which, while widely used, has several limitations:

1. As noted by reviewers, the dataset is over a decade old, and the PCA-transformed features (V1–V28) limit practical interpretability and actionable business insights (e.g., "V14 is most important").
2. The dataset represents transactions from a specific time period (September 2013) and geographic region, poten-tially limiting temporal and geographic generalizability.
3. The relatively small number of fraud instances (492 out of 284,807) may not capture the full diversity of fraud patterns.

Future work should prioritize validating the proposed frame-

International Journal of Engineering, Science and Environment
Peer-Reviewed Open Access • CC BY 4.0 • Quarterly
Vol. 1 | Issue 1 | Apr–Jun 2026

work on a second, more recent dataset (such as the IEEE-CIS Fraud Detection dataset or PaySim) to substantiate these claims across different transaction types and feature spaces.

6.2. Methodological Limitations

- The study focuses on tabular ML classifiers and does not evaluate deep learning architectures (e.g., autoencoders, LSTMs, transformers) or graph neural networks, which have shown promise in recent literature [24], [27], [30].
- Temporal aspects of fraud (concept drift, sequential patterns) are not explicitly modeled.
- The LIME analysis, being perturbation-based, may produce unstable explanations for instances near decision boundaries. Automated hyperparameter optimization (e.g., using Optuna [39] with Bayesian optimization) was not employed due to excessive computational requirements that ren-dered execution infeasible on resource-constrained plat-forms such as Google Colab. Instead, predefined hyper-parameter configurations informed by established best practices were used. While these configurations provide a strong baseline, automated optimization may yield further performance improvements given sufficient com-putational resources.

6.3. Future Directions

Based on the findings and limitations of this study, several directions for future research are identified:

1) Multi-dataset validation with diverse fraud patterns and data characteristics.
2) Integration of deep learning and transformer-based mod-els with explainability analysis.
3) Temporal modeling using sequential architectures to capture evolving fraud patterns.
4) Graph-based approaches leveraging transaction network structures.
5) Real-time deployment considerations, including infer-ence latency and explanation generation speed.
6) Exploration of counterfactual explanations and attention-based interpretability methods.

7. CONCLUSION

This study presented a comprehensive, statistically rigorous framework for explainable credit card fraud detection, address-ing key limitations in the existing literature. Five machine learning classifiers were systematically evaluated with pre-defined hyperparameter configurations, four class imbalance handling techniques (class weighting, SMOTE, Borderline-SMOTE, and ADASYN), 10-fold stratified cross-validation, statistical significance testing with Cohen's *d* effect size, and 95% bootstrap confidence intervals on test results. The experimental results demonstrate that ensemble-based models substantially outperform simpler classifiers on the European cardholder credit card dataset. LightGBM achieves the best overall performance (F1 = 0.7489, ROC-AUC = 0.9817, MCC = 0.7555), followed closely by Random Forest (F1 = 0.7411, MCC = 0.7464) and XGBoost (F1 = 0.7328, MCC = 0.7412). Paired *t*-tests across the 10 cross-validation folds confirm that ensemble models significantly outperform Logistic Regression and Decision Tree, with large Cohen's *d* effect sizes confirming practical significance. The differ-ences among the three ensemble models are small and not always statistically significant, with negligible-to-small effect sizes. SMOTE is selected as the primary class imbalance technique due to its strong discriminative performance (ROC-AUC = 0.9688) and established theoretical foundation. The 95% bootstrap confidence intervals on test set results further validate the robustness of the reported performance estimates. The SHAP analysis identifies V14, V4, and V12 as the most influential PCA-derived features for fraud prediction, with V14 exhibiting the strongest effect. The dual explainability analysis using SHAP and LIME provides complementary insights: SHAP offers theoretically grounded global feature rankings, while LIME provides intuitive rule-based local ex-planations. Both methods agree on the top contributing fea-tures, supporting their combined use for comprehensive model interpretation. The framework's emphasis on reproducibility—through documented random seeds, software versions (Python 3.12.13, scikit-learn 1.6.1, LightGBM 4.6.0), and implemen-tation details—facilitates replication and extension by the research community. Future work should focus on multi-dataset validation, integration of advanced architectures, tem-poral fraud pattern modeling, and automated hyperparameter optimization with sufficient computational resources.

ACKNOWLEDGEMENT

The authors thank the anonymous reviewers for their constructive comments and valuable suggestions, which helped improve the quality of this manuscript.

CONFLICT OF INTEREST

The authors declare that there are no conflicts of interest.

FUNDING STATEMENT

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

REFERENCES

[1] W. Hilal, S. A. Gadsden, and J. Yawney, "Financial fraud: A review of anomaly detection techniques and recent advances," *Expert Systems with Applications*, vol. 193, p. 116429, 2022.
[2] A. Abdallah, M. A. Maarof, and A. Zainal, "Fraud detection system: A survey," *Journal of Network and Computer Applications*, vol. 68, pp. 90–113, 2016.
[3] A. Dal Pozzolo, O. Caelen, Y. A. Le Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," *Expert Systems with Applications*, vol. 41, no. 10, pp. 4915–4928, 2014.
[4] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and A. Pedreschi, "A survey of methods for explaining black box models," *ACM Computing Surveys*, vol. 51, no. 5, pp. 1–42, 2019.
[5] A. B. Arrieta, N. Diaz-Rodriguez, J. Del Ser *et al.*, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[6] D. Minh, H. X. Wang, Y. F. Li, and T. N. Nguyen, "Explainable artificial intelligence: A comprehensive review," *Artificial Intelligence Review*, vol. 55, pp. 3503–3568, 2022.

[7] L. Longo, M. Brcic, F. Cabitza *et al.*, "Explainable artificial intelligence (XAI) 2.0: A manifesto of open challenges and interdisciplinary research directions," *Information Fusion*, vol. 106, p. 102301, 2024.

[8] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[9] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1135–1144.

[10] S. Kumar, S. Ahmad, and M. Uddin, "Explainable machine learning for credit card fraud detection: Balancing accuracy and interpretability," *Journal of Computational Science*, vol. 75, p. 102200, 2024.

[11] M. Ali, S. Khan, and R. Patel, "A comprehensive framework for explain-able credit card fraud detection using SHAP and LIME," *Computers and Security*, vol. 148, p. 103720, 2025.

[12] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[13] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *IEEE International Joint Conference on Neural Networks*. IEEE, 2008, pp. 1322–1328.

[14] A. Fernandez, S. Garcia, F. Herrera, and N. V. Chawla, "SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary," *Journal of Artificial Intelligence Research*, vol. 61, pp. 863–905, 2018.

[15] F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan, and M. Ahmed, "Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms," *IEEE Access*, vol. 10, pp. 39 700–39 715, 2022.

[16] A. A. Taha and S. J. Malebary, "An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine," *IEEE Access*, vol. 11, pp. 23 888–23 902, 2023.

[17] X. Zhang, Y. Han, W. Xu, and Q. Wang, "Credit card fraud detection using attention-based LSTM and gradient boosting models," *Expert Systems with Applications*, vol. 224, p. 119913, 2023.

[18] Z. Wang, H. Liu, and Y. Chen, "Hybrid oversampling and explainable machine learning for financial fraud detection," Information Sciences, vol. 665, p. 120381, 2025.

[19] N. Rtayli and N. Enneya, "Enhanced credit card fraud detection based on SVM-Recursive Feature Elimination and hyper-parameters optimiza-tion," Journal of Information Security and Applications, vol. 55, p. 102596, 2020.

[20] S. M. Lundberg, G. Erion, H. Chen et al., "From local explanations to global understanding with explainable AI for trees," Nature Machine Intelligence, vol. 2, no. 1, pp. 56–67, 2020.

[21] W. Chen, X. Li, and Y. Zhang, "SHAP-enhanced gradient boosting for interpretable fraud detection in financial transactions," Knowledge-Based Systems, vol. 290, p. 111525, 2024.

[22] A. Singh and A. Jain, "Adaptive ensemble methods with explainable AI for financial fraud detection," Applied Intelligence, vol. 54, no. 2, pp. 1580–1598, 2024.

[23] J. Forough and S. Momtazi, "Ensemble of deep sequential models for credit card fraud detection," Applied Soft Computing, vol. 99, p. 106883, 2022.

[24] Z. Li, M. Huang, G. Liu, and C. Jiang, "A hybrid deep learning model for credit card fraud detection with feature engineering," IEEE Transactions on Computational Social Systems, vol. 10, no. 4, pp. 1956–1967, 2023.

[25] A. Rahman, M. Islam, and N. Kumar, "Deep learning approaches for real-time credit card fraud detection: A comparative study," Neural Computing and Applications, vol. 36, pp. 4521–4539, 2024.

[26] Y. Liu, X. Ao, Z. Qin, J. Chi, J. Feng, H. Yang, and Q. He, "Pick and choose: A GNN-based imbalanced learning approach for fraud detection," in Proceedings of the Web Conference 2021. ACM, 2021, pp. 3168–3177.

[27] D. Cheng, X. Wang, Y. Zhang, and L. Zhang, "Graph neural network for fraud detection via spatial-temporal attention," IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 8, pp. 3800–3813, 2023.

[28] Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu, "Enhancing graph neural network-based fraud detectors against camouflaged fraud-sters," Information Sciences, vol. 658, p. 119584, 2024.

[29] D. Ibomoiye and S. O. Akinola, "Attention-based transformer model for credit card fraud detection," International Journal of Advanced Computer Science and Applications, vol. 14, no. 5, pp. 321–330, 2023.

[30] C. Yang, J. Zhang, and F. Wang, "Transformer-based anomaly detection for credit card fraud with self-attention mechanism," Pattern Recogni-tion, vol. 148, p. 110167, 2024.

[31] M. Zhou, K. Li, and S. Wu, "Multi-head attention transformer networks for financial transaction fraud detection," Expert Systems with Applica-tions, vol. 252, p. 124122, 2025.

[32] H. Han, W. Y. Wang, and B. H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in International Conference on Intelligent Computing. Springer, 2005, pp. 878–887.

[33] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[34] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016, pp. 785–794.

[35] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017.

[36] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," BMC Genomics, vol. 21, no. 1, p. 6, 2020.

[37] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," Biochimica et Biophysica Acta (BBA) - Protein Structure, vol. 405, no. 2, pp. 442–451, 1975.

[38] J. Cohen, Statistical Power Analysis for the Behavioral Sciences, 2nd ed. Lawrence Erlbaum Associates, 1988.

[39] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2019, pp. 2623–2631.