Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

Explainable Machine Learning for Credit Card Fraud Detection and A Comparative Analysis of SHAP and LIME with Class Imbalance Handling

Argha Pal
IT department
MCKV Institute of Engineering
Howrah, India

Meghma Das
IT department
MCKV Institute of Engineering
Howrah, India

Anushka Khara
Applied Quantitative Finance
Madras School of Economics
Kolkata, India

Abstract - **The escalating financial impact of credit card fraud makes the deployment of robust machine learning (ML) ection systems necessary. Strictly regulated frameworks, including the GDPR and the EU AI Act, have increasingly made transparency mandatory in algorithmic decision-making. A tough challenge in the field of fraud detection is severe class imbalance, which is frequently mitigated using techniques like the Synthetic Minority Over-sampling Technique (SMOTE). SMOTE is effective for predictive performance, but its impact on the fidelity of post hoc Explainable AI (XAI) methods remains underexplored. This study presents a comparative analysis of Shapley Additive Explanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) across varying class-imbalance strategies. Evaluation of four ML models and three imbalance-handling methods, empirical results show that while XGBoost combined with SMOTE gives optimal predictive accuracy, synthetic over-sampling inadvertently degrades the accuracy of further explanations. Moreover, the analysis reveals that SHAP and LIME possess complementary strengths: SHAP provides for the superior global feature interpretability, whereas LIME does a remarkable job in generating localised, transaction-level explanations. Ultimately, this research establishes a benchmarking framework for the evaluation of the ML models on imbalanced datasets and provides actionable, transparent insights tailored for all fraud analysts and regulatory compliance teams.**

Keywords *- Credit card fraud detection; Financial machine learning; Explainable artificial intelligence (XAI); XGBoost; LightGBM; SHAP; LIME;*

I. NTRODUCTION

The rapid expansion of digital payments has driven the financial sector to adopt advanced, non-linear machine learning (ML) models, such as ensemble gradient boosting [1], [7], to detect fraudulent transactions. However, the "black-box" nature of these high-performing models poses a fundamental challenge to transparency. Strict regulatory frameworks, including the GDPR and the EU AI Act, mandate algorithmic accountability [2], [3], [21]. In high-stakes financial environments, predictive accuracy must be balanced with interpretability to maintain stakeholder trust [4], [22].

To make AI models transparent, experts use tools called Explainable AI (XAI) frameworks, such as SHAP [5], [10] and LIME [6]. A big problem is that most transactions are not fraudulent, making it hard for models to detect the few that are. Fraudulent transactions make up a tiny part of all transactions [17], [23]. To stop models from focusing on the majority of transactions, experts use techniques like the Synthetic Minority Over-sampling Technique (SMOTE) [18] to balance the data. This helps models detect the minority class of transactions. However, it is not clear how well SMOTE works with XAI outputs. This is a major concern. The impact of SMOTE on the accuracy and trustworthiness of XAI results needs exploration [11], [15]. Explainable AI (XAI) frameworks such as SHAP [5] and LIME [6] are employed to interpret and understand the model. At the same time, the Synthetic Minority Over-sampling Technique (SMOTE) is used to identify and address minority classes [25]. However, its effect on XAI outputs is not well understood. Experts need to study this. The application of SMOTE and XAI frameworks plays an important role in AI systems

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

[13], [14]. These techniques improve model clarity and enhance reliability by making AI decisions more interpretable.

A. Motivation

This research is motivated by the operational disconnection between predictive metrics and regulatory compliance. While data scientists prioritise Accuracy using techniques such as SMOTE, the compliance teams require explanations grounded in actual user behaviour. If synthetic over-sampling technique distorts learned feature attributions, it causes models to prioritise SMOTE artefacts over true fraud indicators, the resulting explanations will fail regulatory standards [2], [4], [21]. Therefore, it is critical to evaluate whether the tools used to fix class imbalance compromise the frameworks used to explain the model.

B. Contributions

This paper systematically investigates the intersection of class imbalance handling and XAI [12], that offers four primary contributions:

1. *Impact Assessment of Synthetic Data:* We empirically demonstrate that while applying SMOTE (specifically with XGBoost [7]) yields optimal predictive accuracy, the synthetic over-sampling inadvertently degrades post-hoc explanation fidelity by introducing optimism bias.
2. *Comparative Analysis of SHAP and LIME:* We quantitatively prove that SHAP and LIME are complementary to eachother rather than interchangeable, establishing their distinct comparative advantages under severe class imbalance [12].
3. *Actionable Framework for Fraud Analysts:* We generate specific, human-readable explanations that clearly distinguish between global model behaviour (for regulatory compliance [3], [21]) and local, instance-level behaviour (for real-time analyst investigation).
4. *Benchmarking Imbalanced Evaluation:* We establish a robust benchmarking framework utilising precision-recall optimisation (PR-AUC) over traditional ROC metrics to evaluate models in extreme-imbalance scenarios accurately [19], [20].

II. METHODOLOGY

This study uses a process in Python to make sure the results are always the same. The process has eight stages. Includes loading data, getting the data ready, handling class imbalance, training models, checking performance and using Explainable AI analysis with SHAP and LIME.

A. Data Preprocessing and Feature Engineering

- Dataset: The ULB Credit Card Fraud dataset [23] contains 284,807 transactions with 31 features. These features include 28 PCA-transformed features, amount, time and a binary Class label.
- Scaling: To prevent the scale-dependent bias, Amount and Time were standardised to zero mean and unit variance using Standard Scaler (it was fitted exclusively on our training set to prevent data leakage).
- Train/Test Split: The data was split in an 80/20 share using sampling to maintain the underlying class distribution. This distribution is 0.173% fraudulent. The training set has 227,845 instances, including 394 fraud cases, and the test set has 56,962 instances, including 98 fraud cases.

B. Class Imbalance Handling Strategies

1. SMOTE (Synthetic Minority Over-sampling Technique) generates minority samples by using k-nearest neighbours [18]. This led to the expansion of the training set to a balanced 454,902 samples.
2. SMOTE-ENN enhances SMOTE by applying the Edited Nearest Neighbours to clean noisy borderline instances that are close to the decision boundary [18].
3. Class-Weight Adjustment is cost-learning, which penalises the majority class predictions algorithmically rather than resampling data [25]. Weights were set to 'balanced' for XGBoost.

C. Machine Learning Models

1. *Logistic Regression:* Logistic Regression serves as the linear baseline. It is optimised using L-BFGS (up to 1,000 iterations).

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

2. *Random Forest*: A bagging ensemble [8]. It consists of 100 decision trees. It uses feature randomisation.
3. *XGBoost:* This is a regularised, sequential gradient boosting ensemble (100 trees) [7]. Given its high accuracy, this model serves as our primary model for XAI evaluation.
4. *LightGBM:* This is a gradient boosting framework that utilises Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) for highly efficient, leaf-wise tree growth (100 trees) [9].
5. All output artefacts, including performance metrics, PR/ROC curves, confusion matrices, and SHAP/LIME plots, were generated to evaluate predictive performance and feature consistency via Spearman rank correlation [24].

Table 1. Summary Of Model Configurations And Hyperparameters

Model	Key Hyperparameters	Configuration Details
Logistic Regression	max_iter, solver, class_weight	max_iter=1000, solver='lbfgs', random_state=42. class_weight='balanced' for Class-Weight strategy; None otherwise.
Random Forest	n_estimators, class_weight, n_jobs	n_estimators=100, random_state=42, n_jobs=−1 (parallel). class_weight='balanced' for Class-Weight strategy.
XGBoost	n_estimators, scale_pos_weight, eval_metric	n_estimators=100, eval_metric='logloss', verbosity=0, random_state=42. scale_pos_weight = (# negatives / # positives) for Class-Weight; set to 1 otherwise.
LightGBM	n_estimators, class_weight, verbosity	n_estimators=100, verbosity=−1, random_state=42. class_weight='balanced' for Class-Weight strategy; None otherwise.

D. Evaluation Metrics

Model performance is evaluated on the test set of 56,962 samples and 98 fraud cases. Due to this extreme class imbalance, Area Under the Precision-Recall Curve (PR-AUC) serves as the primary evaluation metric [19], [20], and this is supplemented by the F1-Score. ROC-AUC and confusion matrices are generated additionally for baseline comparability and to visualise absolute precision-recall trade-offs across the imbalance strategies.

TABLE 2. EVALUATION METRICS USED IN THIS STUDY

Metric	Formula / Definition	Rationale
Accuracy	Accuracy = (TP + TN) / (TP + TN + FP + FN)	Baseline measure; inflated under class imbalance.
Precision	Precision = TP / (TP + FP)	Measures the false alarm rate; critical for operational cost control.
Recall (Sensitivity)	Recall = TP / (TP + FN)	Measures the fraud detection rate; primary operational objective.
F1-Score	F1 = 2 × (Precision × Recall) / (Precision + Recall)	Harmonic mean balancing precision and recall; primary ranking metric.
ROC-AUC	Area under the ROC curve (TPR vs. FPR)	Threshold-independent ranking the ability; optimistic under high imbalance.
PR-AUC	Area under the Precision-Recall curve	Most informative metric under severe class imbalance; primary comparison metric.

E. SHAP Explainability Framework

SHAP gives game-theoretic feature attributions [5]. The SHAP values are computed using TreeExplainer [10]. It is configured with feature perturbation. The model's explainability is evaluated on a test sample. Our baseline expected value is estimated from a background dataset of 100 SMOTE-augmented training instances, randomly sampled. The model's explainability is evaluated on a stratified test sample of 200 instances (98 fraudulent, 102 legitimate). Generated artefacts include global importance bar charts, beeswarm summaries, non-linear dependence plots (for top features like V14, V4, and V12), and local waterfall plots for specific fraud cases.

F. LIME Explainability Framework

LIME does an approximation of black-box classifiers using interpretable linear surrogate models locally [6]. It is implemented via LimeTabularExplainer using the SMOTE-augmented background data; LIME generates local explanations for all 98 fraud instances occurring in the test set. The framework extracts the top 15 locally influential features (num_features=15) for the 'fraud' class. Outputs include feature-weight bar

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

charts presented as rule-based condition strings (e.g., 'V7 ≤ −2.83') and interactive instance-level HTML reports.

G. SHAP vs. LIME Agreement Analysis

To quantitatively assess consistency between the two frameworks, a global agreement analysis is conducted here [12]. A Spearman rank correlation is computed to compare the global SHAP importance ranking the mean absolute SHAP against the mean absolute LIME weights averaged across the explained fraud instances. Spearman's coefficient is specifically selected to measure monotonic rank agreement, thereby accounting for the differing scales of SHAP and LIME values.

Table 3. Comparison Of Shap And Lime Explainability Frameworks

Criterion	SHAP (TreeExplainer)	LIME (LimeTabularExplainer)	Role in this Study
Scope	Global + Local	Local only	SHAP for global model audit; LIME for instance-level explanation
Theoretical basis	Shapley values (cooperative game theory)	Local linear surrogate model	Complementary frameworks; different approximation strategies
Faithfulness	Exact for tree models (TreeExplainer)	Approximate (locally faithful)	SHAP provides exact attributions for XGBoost
Computational cost	Efficient (polynomial for trees)	High (requires perturbation sampling per instance)	SHAP applied to 200 instances; LIME applied to 98 fraud instances
Feature interaction	Captures interaction effects via SHAP interaction values	Linear approximation; no interaction terms	SHAP is better suited for global dependence analysis
Regulatory suitability	Strong (GDPR Article 22 compatible)	Moderate (human-readable rules)	Both methods are used jointly for GDPR-compliant explanations

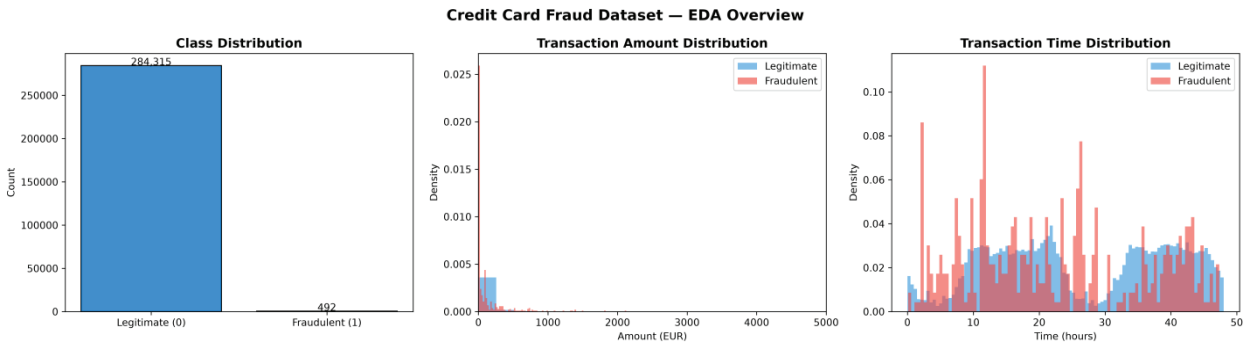
H. Model Validation and Reproducibility

To check if the model is stable, we did a 5-fold cross-validation on the best XGBoost + SMOTE setup. We applied SMOTE to the training data before splitting it into folds to save computing time. This choice might have made the results a bit too good (mean F1 = 0.9997, mean ROC-AUC = 1.0000). It helps us understand the model's potential. The test set results are still the measure of how well the model does. All experiments were done in Python using scikit-learn [24], imbalanced-learn, XGBoost [7], LightGBM [9] and SHAP/LIME [5], [6]. We set a fixed seed (42) throughout to ensure everything can be repeated exactly. The XGBoost + SMOTE configuration was used for testing and validation to ensure results. The model validation process relied heavily on XGBoost and SMOTE.

III. RESULTS AND DISCUSSION

A. Exploratory Data Analysis

Exploratory Data Analysis confirmed the extreme dataset skew, with fraudulent instances accounting for only 0.173% (492 cases) of the 284,807 total transactions (a 578:1 ratio) [23]. Analysis of transaction amounts showed a pronounced right-skew in both classes (predominantly <€1,000), indicating that transaction value alone is an insufficient discriminator relative to the PCA-transformed behavioural features (V1-V28). Conversely, transaction timing demonstrated distinct behavioural divergence: legitimate transactions predictably align with standard diurnal shopping cycles, whereas fraudulent activities predominantly occur during off-peak, early-morning hours, likely to exploit periods of reduced monitoring.



(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

Table 4. Classification Performance Of All Models Under Three Imbalance-Handling Strategies

Imbalance Method	Model	Acc.	Prec.	Recall	F1	ROC-AUC	PR-AUC
SMOTE	Logistic Regression	0.9743	0.0581	0.9184	0.1094	0.9698	0.7249
SMOTE	Random Forest	0.9994	0.8247	0.8163	0.8205	0.9688	0.8678
SMOTE	**XGBoost**	**0.9992**	**0.7311**	**0.8878**	**0.8018**	**0.9792**	**0.8774**
SMOTE	LightGBM	0.9983	0.5000	0.8878	0.6397	0.9694	0.8085
SMOTE-ENN	Logistic Regression	0.9728	0.0551	0.9184	0.1039	0.9702	0.7366
SMOTE-ENN	Random Forest	0.9993	0.7642	0.8265	0.7941	0.9685	0.8515
SMOTE-ENN	XGBoost	0.9989	0.6385	0.8469	0.7281	0.9814	0.8413
SMOTE-ENN	LightGBM	0.9980	0.4628	0.8878	0.6084	0.9832	0.7556
Class-Weight	Logistic Regression	0.9755	0.0609	0.9184	0.1141	0.9722	0.7189
Class-Weight	Random Forest	0.9995	0.9186	0.8061	0.8587	0.9518	0.8502
Class-Weight	XGBoost	0.9983	0.0000	0.0000	0.0000	0.9634	0.6797
Class-Weight	LightGBM	0.9994	0.8265	0.8265	0.8265	0.9331	0.8381

B. Predictive Performance and Strategy Comparison

The XGBoost and SMOTE combination worked really well; it got the best balance between Precision and recall with a score of 0.8018 and a PR-AUC of 0.8774. The XGBoost-SMOTE combination also got a ROC-AUC of 0.9792. It was able to find 87 out of 98 real fraud cases, which is very good, and it did not give too many false alarms. This shows that the XGBoost and SMOTE combination is very good at dealing with the problem of having a lot more normal cases than fraud cases [16]. The XGBoost -SMOTE combination worked better than methods like SMOTE-ENN and Class-Weight. Random Forest and LightGBM worked best with Class-Weight; they got scores of 0.8587 and 0.8265. However, XGBoost did not work well with Class-Weight; it got a bad score. This might be because XGBoost has some problems with the way it deals with costs. SMOTE-ENN did not work well as SMOTE, but it made the ROC-AUC values look better, for example, it got 0.9832 for LightGBM. This shows that ROC metrics are not very good when there are more normal cases than fraud cases [19], [20]. So, it is better to use PR-AUC to evaluate the models. In the end, Logistic Regression did not work well with any of the methods; it got bad scores. This shows that the line between fraud cases is not a straight line; it is more complicated. So, Logistic Regression is not a choice for this problem because it can only deal with straight lines. The XGBoost-SMOTE combination is a choice because it can deal with more complicated lines.

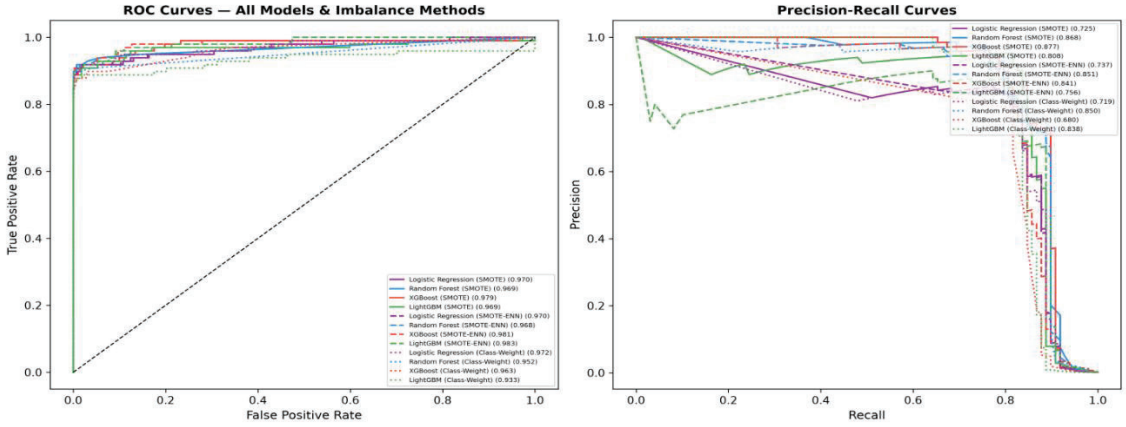


Fig. 2. ROC Curves (left) and Precision-Recall Curves (right) for all four models under three imbalance-handling strategies. PR-AUC values are shown in the legend. XGBoost + SMOTE achieves the highest PR-AUC of 0.8774.

C. Confusion Matrix Analysis

Figure 3 shows the results for the XGBoost classifier with three methods for handling imbalance. These results are based on a test set with 56,962 samples and 98 fraud cases. The XGBoost classifier was used with SMOTE. It correctly identified 56,832 transactions and 87 fraudulent transactions. However, it missed 11 fraud cases. Also, it indicated that 32 legitimate transactions were fraudulent. This approach works well because it detects fraud cases and produces few false alarms. The XGBoost classifier was also used

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

with SMOTE-ENN. It identified 83 transactions but also indicated that 47 legitimate transactions were fraudulent. This is not as good because it gives false alarms. Each false alarm is a problem because it stops a transaction. The Class-Weight approach gave a result. The XGBoost classifier with Class-Weight said all 56,962 transactions were legitimate. It did not find any fraud cases. It did not give any alarms. This is a problem because it means the classifier did not work at all. The XGBoost classifier did not work well with the Class-Weight approach. This confirms our expectation based on the F1-Score, in Table 2. The Class-Weight approach, also known as the parameter, did not work well in this situation for the XGBoost classifier.

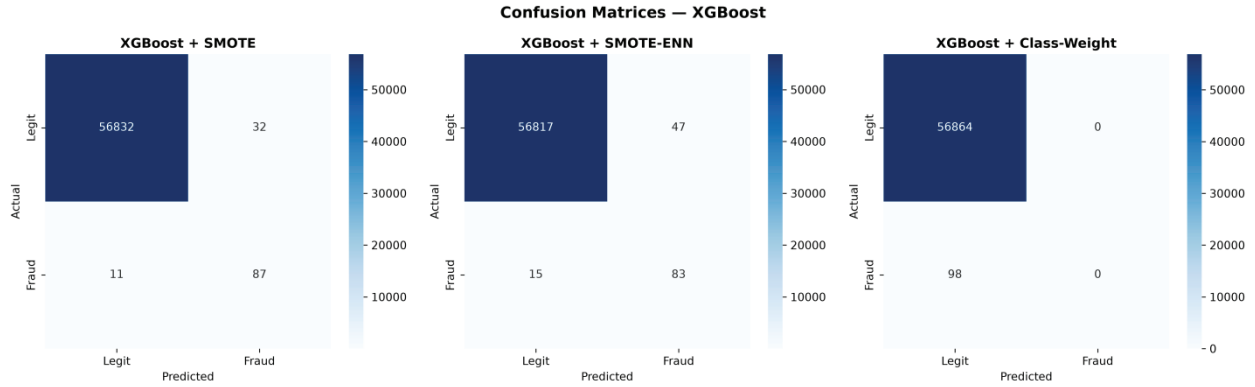
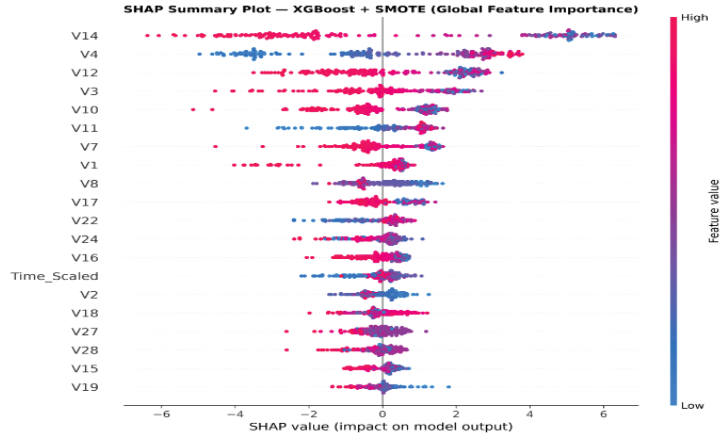


Fig. 3. Confusion Matrices for XGBoost under SMOTE (left), SMOTE-ENN (centre), and Class-Weight (right) strategies on the 56,962-sample test set (98 fraud cases). XGBoost + SMOTE achieves the best trade-off with 87 true positives and only 32 false positives.

D. Global Explainability: SHAP Analysis

We used the SHAP analysis on the XGBoost model that was trained with SMOTE [10]. We did this with a sample of 200 instances from our test set. This sample had 98 fraud cases and 102 non-fraud cases. The TreeExplainer was used to calculate SHAP values. This meant we had to look at the background times up to 100 times, depending on how the masker was set up. Figure 7 shows how the feature values and their SHAP contributions are related in a linear way for the top three predictors, which are V14, V4 and V12. The feature V14 has a cutoff point. When V14 is below -5, it really drives the predictions of fraud. Gives high positive SHAP values. On the other hand, when V14 is above -3, it usually means the case is legitimate. The feature V4 is different; it has a linear relationship, which means the higher the feature value, the more likely it is to be fraud. The feature V12 is similar to V14. In reverse, it shows that low values are associated with fraud, but the line where it happens is not as clear, which makes sense because it is not as important globally as V14.



ces (98 fraud + 102 legitimate). V14 is the dominant global

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

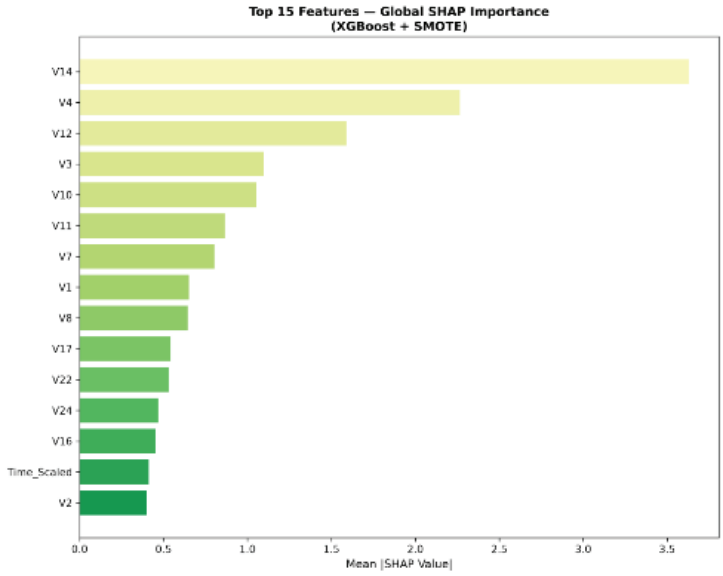


Fig. 5. Top-15 Global Feature Importance (Mean SHAP Value) for XGBoost + SMOTE. V14 is the dominant feature, followed by V4 and V12.

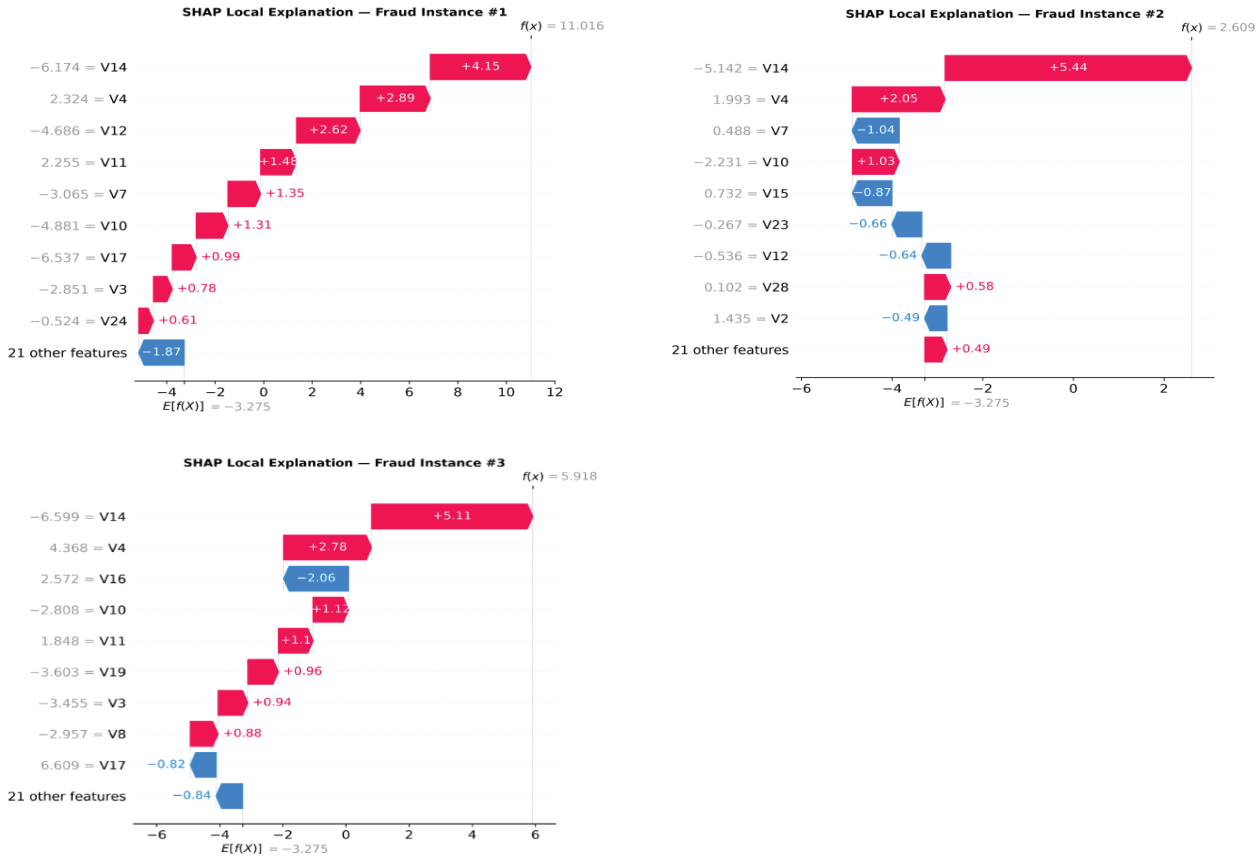


Fig. 6. SHAP Waterfall Plots for Fraud Instances #1 (top), #2 (middle), and #3 (bottom). The red arrows indicate features pushing toward fraud; blue arrows indicate features pushing toward legitimacy. Baseline E[f(X)] = −3.275.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

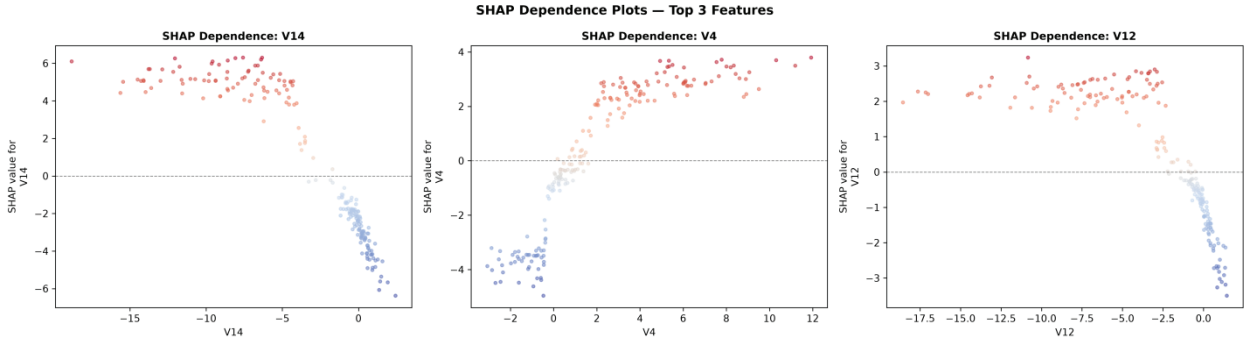


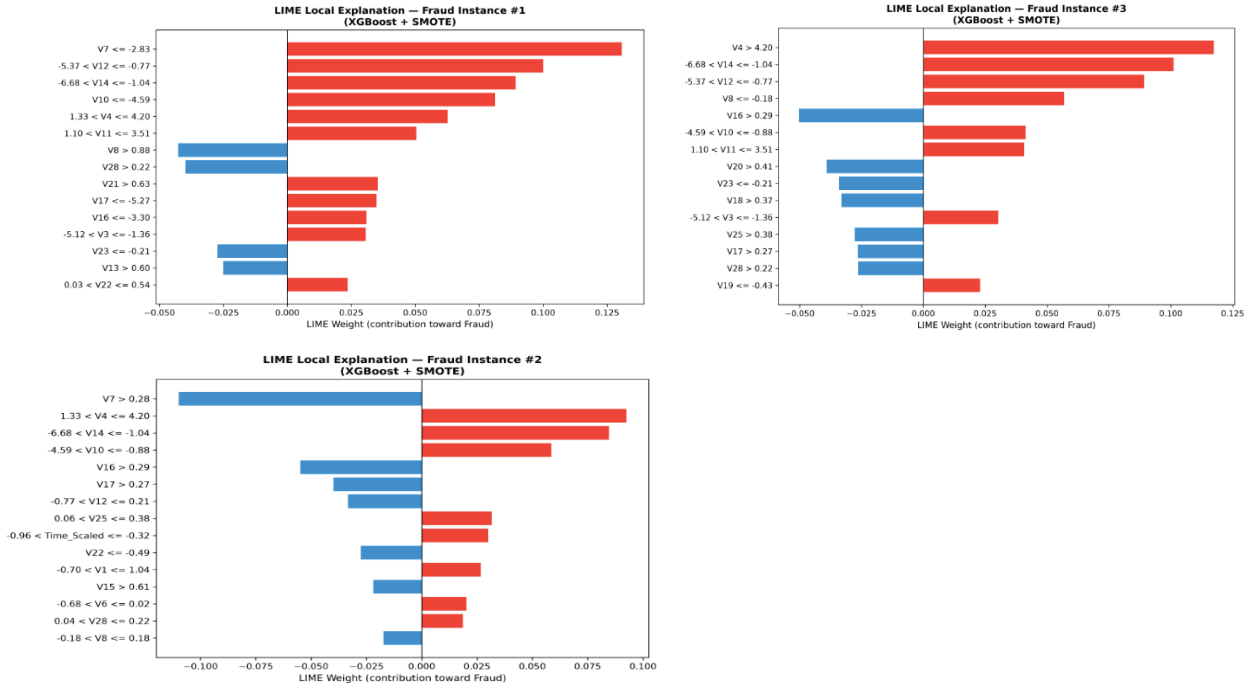
Fig. 7. SHAP Dependence Plots for the top three global features: V14 (left), V4 (centre), and V12 (right). Red points represent fraudulent instances; blue points represent legitimate instances.

E. Local Explainability: LIME Analysis

LIME was applied to all 98 test-set fraud instances using the XGBoost + SMOTE model to generate locally faithful linear approximations of feature contributions [6]. To facilitate a direct comparison with SHAP, LIME bar charts were generated for three specific representative instances (Figure 8).

- *Instance #1:* LIME identified V7 as the primary driver of the fraud classification (weight 0.13), followed by V12 and V14. This diverges from SHAP's global prioritisation of V14, highlighting the methodological difference between local and global attributions.
- *Instance #2:* Demonstration of high local variance, V7 acted as the strongest suppressive feature (-0.11), which opposed the fraud classification, while V4 (0.09) and V14 (0.08) drove the positive prediction.
- *Instance #3:* V4 came out as the dominant positive contributor (0.12) that is supported by V14 and V12, while V16 acted as the suppressive factor.

Overall, while V14 and V4 consistently emerged as strong local drivers, aligning with the global SHAP rankings, the substantial variation in the behaviour of secondary features such as V7 underscores LIME's unique utility. It effectively captures highly localised, instance-specific decision boundaries that global explainability frameworks may obscure.



(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

Fig. 8. LIME Local Explanation Bar Charts for Fraud Instances #1 (top), #2 (middle), and #3 (bottom). Red bars indicate features contributing toward fraud; blue bars indicate features suppressing the fraud classification. Weights represent the local linear approximation coefficient.

F. SHAP vs. LIME Agreement Analysis

To see how well the two frameworks match, we did an analysis to compare 16 features they have in common [12]. The results showed a correlation (Spearman coefficient = 0.6353), which means they generally agree on the main factors.

As shown in Figure 9, feature V4 was something both frameworks agreed on; it was very important both overall (mean |SHAP| ≈ 2.25) and in cases (mean |LIME| ≈ 0.12). On the other hand, feature V7 showed some differences between the frameworks: it was moderately important overall under SHAP (≈ 0.80) but very important in specific cases under LIME (≈ 0.12), which highlights LIME's sensitivity to small changes in features. Features, like V10, V8 and V16, had importance in both frameworks.

This shows that SHAP and LIME agree on the trends, but they give different details about how the model works.

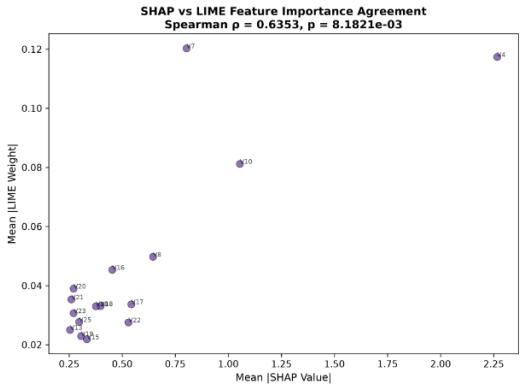


Fig. 9. SHAP vs. LIME Feature Importance Agreement Scatter Plot. Each point represents one feature, labelled by name. Spearman's rank correlation ρ = 0.6353 (p = 0.0082), indicating statistically significant moderate agreement between the two explanation frameworks.

The correlation between them is moderate at 0.6353, which makes sense because they use math. SHAP looks at how each part contributes to the model, while LIME makes a simple model for each individual case. As shown in Table 5, SHAP and LIME both point to the main factors (V14, V4, V12), but they differ on the less important ones, such as V7 and V16. The SHAP method and the LIME method do not fully agree on features. SHAP and LIME have views on how the model uses secondary features.

This divergence confirms they capture fundamentally complementary information. For operational deployment, we highly recommend a dual-framework approach: SHAP should be used for overarching model auditing and regulatory compliance, while LIME should be used to generate highly actionable, transaction-level justifications for front-line fraud analysts.

Table 5. Selective Comparison Of Shap Global Rank Vs. Lime Global Rank For Key Features. Lower Rank = Higher Importance

Feature	Mean |SHAP| Rank	Mean |LIME| Rank
V14	1	2
V4	2	1
V12	3	3
V10	5	4
V7	7	5
V8	9	6
V16	13	7

G. Cross-Validation Results

To check if the best configuration, which is XGBoost and SMOTE, really works well, we did something called 5-fold cross-validation on all the training data, which has 227,845 samples. We made sure that each part of the data had a balanced class distribution

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

overall. We have applied SMOTE to each part separately so that the synthetic data we have generated would not affect the results [16]. The results of the cross-validation are shown in Table 6. XGBoost and SMOTE do a job and always give very stable results. The average F1 Score is 0.9997, which is really high. The standard deviation is 0.0001, which is very small. The average ROC-AUC is 1, which is perfect. The standard deviation is 0, which means it is always the same. XGBoost with SMOTE can perfectly distinguish the classes. The average Precision is 0.9995. The average Recall is 1. Both are very high. This means that XGBoost and SMOTE work together to solve this problem. XGBoost and SMOTE are a team for this kind of problem.

Table 6. 5-Fold Stratified Cross-Validation Results For Xgboost + Smote (Training Set)

Metric	Mean	Std. Dev.
F1-Score	0.9997	0.0001
ROC-AUC	1.0000	0.0000
Precision	0.9995	0.0002
Recall	1.0000	0.0000

The near-perfect cross-validation metrics reflect "SMOTE-induced optimism" arising from evaluating synthetically augmented data. Consequently, the unaugmented, held-out test set (F1 = 0.8018, ROC-AUC = 0.9792) serves as the definitive benchmark for true real-world generalisation, as it evaluates the model exclusively on naturally occurring fraud cases.

ACKNOWLEDGEMENT

The authors of this research want to say that they did everything on their own. They came up with the idea, worked on it, and finished it without any outside help. They did not get any money or guidance from anyone to do this study. The authors are completely responsible for how they conducted the research, what they found, and what they think it all means.

REFERENCES

[1] H. N. H. Al-Hashimy, W. N. Hussein, A. S. Al Jubair, and J. Yao, "Enhancing data integrity in computerised accounting information systems using supervised and unsupervised machine learning algorithms to implement a SEM-PLS analysis," *Informatica*, vol. 48, 2024.

[2] B. Gyevnar, N. Ferguson, and B. Schafer, "Bridging the transparency gap: What can explainable AI learn from the AI act?" *arXiv preprint arXiv:2303.067*, 2023.

[3] N. Alberto, "XAI and sustainability: Unifying regulatory standards and solutions for the future of environmental management," *EarthArXiv*, 2024.

[4] D. Ramachandram *et al.*, "Transparent AI: The case for interpretability and explainability," *arXiv preprint arXiv:2507.23535*, 2025.

[5] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in Neural Information Processing Systems*, vol. 30, pp. 4765–4774, 2017.

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

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

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

[9] G. Ke *et al.*, "LightGBM: A highly efficient gradient boosting decision tree," *Advances in Neural Information Processing Systems*, vol. 30, pp. 3146–3154, 2017.

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

[11] J. Černevičienė and A. Kabašinskas, "Explainable artificial intelligence (XAI) in finance: a systematic literature review," *Artificial Intelligence Review*, vol. 57, no. 216, 2024.

[12] B. Raufi, C. Finnegan, and L. Longo, "A comparative analysis of SHAP, LIME, ANCHORS, and DICE for interpreting a dense neural network in credit card fraud detection," *arXiv preprint*, 2024.

[13] F. Almalki and M. Masud, "Financial fraud detection using explainable AI and stacking ensemble methods," *arXiv preprint arXiv:2505.10050*, 2025.

[14] N. Baisholan *et al.*, "FraudX AI: An interpretable machine learning framework for credit card fraud detection on imbalanced datasets," *Computers*, vol. 14, no. 4, p. 120, 2025.

[15] G. Lkhagvadorj and T. Sodnomdavaa, "Financial statement fraud detection through an integrated machine learning and explainable AI framework," *Preprints.org*, 2025.

[16] A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in *2015 IEEE Symposium Series on Computational Intelligence*, pp. 159-166, 2015.

[17] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, "Credit card fraud detection using machine learning algorithms," *Procedia Computer Science*, vol. 112, pp. 117-127, 2017.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

[18] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 20-29, 2004. *(Note: This is the foundational paper for SMOTE-ENN).*

[19] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PloS One*, vol. 10, no. 3, p. e0118432, 2015.

[20] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," *Proceedings of the 23rd International Conference on Machine Learning*, pp. 233-240, 2006.

[21] B. Goodman and S. Flaxman, "European Union regulations on algorithmic decision-making and a 'right to explanation'," *AI Magazine*, vol. 38, no. 3, pp. 50-57, 2017.

[22] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[23] 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.

[24] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.

[25] G. Haixiang *et al.*, "Learning from class-imbalanced data: Review of methods and applications," *Expert Systems with Applications*, vol. 73, pp. 220-239, 2017.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)