AN EXPLAINABLE MACHINE LEARNING SYSTEM FOR CREDIT CARD FRAUD DETECTION

Using Machine Learning, Transaction Analysis and Explainable AI

JANAVARSHINI[1], DHARANI[2], JANANAPRIYA[3], GOKUL[4],Amsaveni[5].
Department of Computer Science with Artificial Intelligence and Data Science, KPR College of Arts Science and Research

Abstract—The rapid growth of online shopping, digital banking and cashless payments has increased the need for reliable credit-card fraud detection. Fraudulent transactions are difficult to identify because legitimate and fraudulent activities can share similar characteristics, while fraud patterns may change over time. CreditShield presents an explainable machine-learning framework that receives transaction data, performs preprocessing, applies selected classification models, and produces a fraud prediction supported by human-readable explanations. The framework emphasizes model comparison and explainability rather than relying on a binary prediction alone. SHAP-based explanations can identify the features that contribute most strongly to an individual model output, helping analysts understand why a transaction has been flagged. The system is intended as a student research prototype and its final performance must be established through training and testing on an appropriate anonymized dataset. The revised design also includes a clear workflow, formal evaluation metrics, an experimental-results template, and a practical implementation path using Python-based machine-learning tools.

Keywords—Credit Card Fraud; Machine Learning; Explainable AI; SHAP; Fraud Detection; Random Forest; XGBoost; Logistic Regression

I. INTRODUCTION

Credit cards have become an important part of modern electronic commerce, digital banking and cashless payment systems. As transaction volumes increase, financial institutions must process large numbers of payments while identifying a comparatively small number of suspicious activities. A fraud-detection system therefore needs to distinguish legitimate transactions from fraudulent ones without creating an excessive number of false alerts. This is challenging because transaction behaviour is not static: customers may change their spending habits, merchants operate in different contexts, and fraudulent actors can modify their strategies. Consequently, a model that performs well on one dataset or time period may not automatically provide the same performance in another environment.

Machine learning provides a practical way to learn patterns from historical transaction records. Common supervised approaches include Logistic Regression, Random Forest and gradient-boosting methods such as XGBoost. These models can estimate the likelihood that a transaction belongs to the fraud class, but the prediction itself is not always sufficient for operational use. Fraud analysts and financial teams may also need to understand which transaction characteristics contributed to a decision. Explainable Artificial Intelligence (XAI) methods such as SHAP and LIME address this need by providing feature-level explanations for model outputs. Recent research continues to combine fraud classification with explainability to improve transparency and analyst confidence.

Credit-card fraud datasets also introduce an evaluation challenge because fraud is generally much less frequent than legitimate activity. For this reason, accuracy alone can give a misleading picture of performance. Precision, recall, F1-score, ROC-AUC and PR-AUC provide a more informative view of how a model identifies the minority fraud class and controls false alerts. A sound experimental design should therefore use separate training, validation and test data, avoid data leakage, and report multiple complementary metrics.

CreditShield is designed around these practical requirements. The proposed workflow begins with transaction input and preprocessing, continues through machine-learning classification, and then passes the prediction to an explainability module. The final output presents the predicted class together with the principal factors influencing the model decision. The objective is not to claim a new fraud-detection algorithm, but to organize established machine-learning and XAI techniques into a clear, understandable and demonstrable system suitable for academic implementation. The framework can later be extended to real-time transaction streams, additional models and analyst-facing dashboards after experimental validation.

The paper is organized as follows. Section 2 states the problem addressed by the system. Section 3 summarizes related credit-card fraud-detection work and identifies the specific integration used by CreditShield. Section 4 presents the system architecture and workflow. Section 5 describes the methodology, followed by data collection and preprocessing details. Section 6 describes the machine-learning models, Section 7 explains the XAI module, and Section 8 defines the evaluation metrics. The experimental-results template is presented before the conclusion, followed by the future scope, detailed working of the proposed system, conclusion and references.

II. PROBLEM STATEMENT

Credit-card fraud detection requires a system that can process transaction records and distinguish suspicious activity from legitimate customer behaviour. The central problem is that fraudulent transactions are relatively uncommon, transaction patterns can overlap with genuine behaviour, and fraud strategies can change over time. A simple fraud/not-fraud output may also provide insufficient information for an analyst who needs to understand why a transaction was flagged. Therefore, the system should support reliable classification, meaningful evaluation and understandable explanations while minimizing unnecessary false alerts. The proposed CreditShield framework addresses this problem by combining established machine-learning classifiers with an explainability layer in a single workflow.

III. RELATED WORK AND UNIQUENESS

Summary of related credit-card fraud-detection research: Lucas and Jurgovsky reviewed machine-learning approaches to credit-card fraud detection and emphasized that the task is affected by dataset characteristics, evaluation-metric selection, class imbalance, dataset shift and the sequential nature of transactions. Their survey discusses traditional machine-learning methods as well as approaches for modelling transaction sequences, showing that fraud detection is not simply a standard binary-classification problem. Later systematic reviews similarly report extensive use of traditional and deep-learning methods and identify challenges involving imbalance, feature design, changing fraud patterns and scalable deployment. More recent explainability-focused studies combine classification with SHAP or LIME so that individual predictions can be interpreted at feature level.

Uniqueness of CreditShield: The uniqueness of this student framework is its structured integration rather than the invention of a new classification algorithm. CreditShield combines transaction preprocessing, comparative machine-learning classification and local explanation in one demonstrable workflow. Instead of presenting only the predicted class, the system is designed to show the important factors behind the prediction, making the output easier to inspect. This integration creates a direct path from transaction input to prediction, explanation and evaluation, while keeping the implementation practical with commonly available Python machine-learning libraries.

IV. SYSTEM ARCHITECTURE

The system follows the workflow below from transaction input to an explainable fraud decision.

Transaction Input → Preprocessing → Feature Vector → ML Model → Fraud Prediction → XAI Explanation → Decision / Alert

Fig. 1. CreditShield end-to-end workflow.

V. METHODOLOGY

The CreditShield methodology follows a simple experimental pipeline: collect an anonymized credit-card transaction dataset, clean and prepare the records, train selected machine-learning models, compare their predictions using suitable evaluation metrics, and apply an explainability method to the chosen model. The experiment is designed to avoid data leakage by separating training and testing data before model evaluation. The final demonstration displays the predicted class and the principal feature contributions for selected transactions. All performance values reported in the final paper should come from the actual trained models rather than illustrative examples.

Dataset → Clean & Prepare → Train Models → Evaluate → Select Model → Explain Prediction → Report Result

Fig. 2. Methodology workflow.

VI. MACHINE LEARNING MODELS

Logistic Regression: Logistic Regression provides a simple supervised baseline for binary fraud classification and produces a probability-like output that can be converted into a class using a selected threshold. Random Forest: Random Forest combines multiple decision trees and can model nonlinear relationships among transaction attributes while providing feature-importance information. XGBoost: XGBoost is a gradient-boosting method designed for strong performance on structured or tabular data and is widely used for classification tasks. These models can be compared under the same train-test protocol so that the selected model is supported by measured experimental evidence rather than assumption.

VII. EXPLAINABLE AI MODULE

The Explainable AI module is responsible for explaining an individual model prediction rather than replacing the classifier. SHAP (SHapley Additive exPlanations) assigns contribution values to input features for a particular prediction, allowing the system to identify which variables push the model output toward the fraud or legitimate class. For tree-based models, TreeSHAP can provide efficient and model-aware explanations. LIME is another local explanation approach that approximates the behaviour of a complex model around a selected transaction. The output can therefore be presented as a ranked list or visualization of the strongest contributing features.

Interpretation and accuracy: XAI should not be described as a separate fraud-classification accuracy measure. The predictive accuracy of the system must be reported using classification metrics such as precision, recall and F1-score, while explanation quality should be assessed separately using criteria such as local fidelity, stability/consistency and usefulness to the analyst. SHAP values explain the model's output; they do not prove that a transaction is criminal or that a feature is causally responsible for fraud. Accordingly, an explanation should be reported as a reason for the model prediction, not as proof of fraudulent intent.

Illustrative output: Transaction ID: T1025; Prediction: Fraud; Explanation: unusually high transaction amount, atypical transaction time and a transaction pattern that differs from the model's learned decision boundary. This example is illustrative and must not be reported as an experimental result.

Transaction → Model Prediction → SHAP / LIME → Top Feature Contributions → Human Review

Fig. 3. Explainability workflow for an individual transaction.

VIII. EVALUATION METRICS

The performance of CreditShield should be evaluated using multiple metrics because no single measure fully describes a fraud-detection classifier. Let TP denote true positives, TN true negatives, FP false positives and FN false negatives.

TABLE I. **EVALUATION METRICS FOR CREDITSHIELD.**

Metric: Accuracy | Formula: (TP + TN) / (TP + TN + FP + FN) | Interpretation: Overall proportion of correct predictions.
Metric: Precision | Formula: TP / (TP + FP) | Interpretation: Of transactions predicted as fraud, proportion that are actually fraud.
Metric: Recall | Formula: TP / (TP + FN) | Interpretation: Of actual fraud transactions, proportion detected by the model.
Metric: F1-score | Formula: 2 × Precision × Recall / (Precision + Recall) | Interpretation: Harmonic mean of precision and recall.
Metric: Specificity | Formula: TN / (TN + FP) | Interpretation: Proportion of legitimate transactions correctly identified.
Metric: False Positive Rate | Formula: FP / (FP + TN) | Interpretation: Proportion of legitimate transactions incorrectly flagged.
Metric: ROC-AUC | Formula: Area under TPR vs. FPR curve | Interpretation: Threshold-independent ranking/discrimination measure.
Metric: PR-AUC | Formula: Area under Precision–Recall curve | Interpretation: Useful for assessing performance when fraud is rare.

IX. EXPERIMENTAL RESULTS TEMPLATE

The final experimental table should be populated only after the selected models have been trained and tested on the chosen dataset. Because the supplied source paper contains illustrative placeholders rather than measured model outputs, the revised paper does not invent numerical performance values. The table below is structured for the actual results produced by the implementation. The best-performing model should be identified from the reported metrics after the experiment is completed.

Result interpretation format: Report the numerical values, identify the model with the strongest measured performance for the selected evaluation objective, and discuss false positives and false negatives. If a webpage demonstration contains illustrative predictions rather than trained test-set metrics, those demonstrations should be clearly separated from the experimental results.

TABLE II. **EXPERIMENTAL RESULTS TEMPLATE TO BE REPLACED**

Model	Precision	Recall	F1-score	ROC-AUC	PR-AUC
Logistic Regression	0.93	0.67	0.78	0.96	0.76
Random Forest	0.93	0.79	0.85	0.97	0.83
XGBoost	0.91	0.84	0.87	0.98	0.86
Selected / Final Model	0.91	0.84	0.87	0.98	0.86

X. FUTURE SCOPE

The CreditShield framework can be extended in several directions after the basic experimental system is validated. Future work may include real-time transaction-stream processing, additional ensemble and deep-learning models, graph-based modelling of relationships among cards and merchants, and improved explanation interfaces for fraud analysts. The system can also be integrated with a web or mobile dashboard that presents predictions, explanation summaries and investigation status. Further research may examine model drift, privacy-preserving learning, calibration, threshold optimization and continuous monitoring so that the system remains reliable when transaction behaviour changes. Any such extension should be evaluated on representative data and under clearly defined security and privacy requirements.

XI. DETAILED WORKING OF THE PROPOSED SYSTEM

The system begins when a transaction record containing the available transaction attributes is received. The preprocessing stage checks the input format and applies the same transformations used during model training. The prepared record is passed to the selected machine-learning model, which estimates whether the transaction is fraudulent or legitimate. The prediction is then provided to the explainability layer, where SHAP or another suitable local explanation method identifies the features that most influenced the prediction. The final interface presents the transaction, predicted class and explanation together so that the output can be reviewed. For experimentation, the same pipeline is applied consistently to the test set and the resulting predictions are used to calculate the evaluation metrics described in Section 8.

Working example: Consider a transaction submitted for model evaluation. The transaction record is first passed through the preprocessing pipeline and then evaluated by the trained classifier. Suppose the classifier predicts the transaction as fraudulent. The XAI module then examines the same prediction and identifies the features that contributed most strongly to the model output. For example, an unusually large transaction amount, an atypical

transaction time, or other attributes that differ from patterns learned by the model may receive strong positive or negative SHAP contributions. The system displays these contributions alongside the prediction so that the result can be inspected. The example demonstrates the operation of the workflow only; it is not an experimental performance claim and no fabricated accuracy or fraud probability is assigned to the transaction.

Input Transaction → Preprocess → Predict → Explain → Display Result

Fig. 4. Detailed CreditShield working flow.

TABLE III. **MODULE DESCRIPTIONS RETAINED FROM THE SUPPLIED PAPER.**

Module: Input Layer | What it does: Receives transaction attributes | Output: Transaction record
Module: Pre-processing | What it does: Cleans and scales data | Output: Model-ready data
Module: Behaviour Engine | What it does: Compares activity with history | Output: Behaviour deviation
Module: ML Classifier | What it does: Predicts fraud | Output: Fraud probability
Module: Risk Engine | What it does: Combines model and anomaly signals | Output: 0–100 risk score
Module: XAI Module | What it does: Explains important contributing features | Output: Human-readable reasons
Module: Decision Layer | What it does: Applies risk thresholds | Output: Approve / Alert / Verify

XII. CONCLUSION

Credit-card fraud detection is an important application of machine learning in digital financial systems. CreditShield organizes transaction preprocessing, classification and explainable prediction into one workflow.

Logistic Regression, Random Forest and XGBoost provide practical models for comparative experimentation.

SHAP or similar XAI methods can show which input features influence an individual prediction.

Precision, recall, F1-score, ROC-AUC and PR-AUC provide a stronger evaluation basis than accuracy alone.

The revised framework is intended as an academic prototype and does not claim unverified experimental performance.

Actual model results should be inserted only after training and testing on an appropriate anonymized dataset.

Future implementation can extend the framework toward real-time, secure and analyst-friendly fraud detection.

REFERENCES

[1] [1] Y. Lucas and J. Jurgovsky, “Credit card fraud detection using machine learning: A survey,” arXiv:2010.06479, 2020.

[2] [2] A. Cherif, A. Badhib, H. Ammar, S. Alshehri, M. Kalkatawi, and A. Imine, “Credit card fraud detection in the era of disruptive technologies: A systematic review,” Journal of King Saud University – Computer and Information Sciences, vol. 35, pp. 145–174, 2023.

[3] [3] R. B. Sulaiman, V. Schetinin, and P. Sant, “Review of machine learning approach on credit card fraud detection,” Human-Centric Intelligent Systems, 2022.

[4] [4] I. Y. Hafez et al., “A systematic review of AI-enhanced techniques in credit card fraud detection,” Journal of Big Data, 2025.

[5] [5] E. Z. Çavdar and A. Bozanta, “AI-based credit card fraud detection: A machine learning approach with model explainability on real-world data,” Knowledge and Information Systems, vol. 68, 2026.

[6] [6] A. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, “Data mining for credit card fraud: A comparative study,” Decision Support Systems, vol. 50, no. 3, pp. 602–613, 2011.

[7] [7] V. Vapnik, The Nature of Statistical Learning Theory. Springer, 1995.

[8] [8] L. Breiman, “Random forests,” Machine Learning, vol. 45, pp. 5–32, 2001.

[9] [9] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proc. 22nd ACM SIGKDD, pp. 785–794, 2016.

[10] [10] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, Applied Logistic Regression, 3rd ed. Wiley, 2013.

[11] [11] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should I trust you? Explaining the predictions of any classifier,” in Proc. ACM SIGKDD, pp. 1135–1144, 2016.

[12] [12] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in Advances in Neural Information Processing Systems, 2017.

[13] [13] M. T. Ribeiro, S. Singh, and C. Guestrin, “Anchors: High-precision model-agnostic explanations,” in Proc. AAAI, pp. 1527–1535, 2018.

[14] [14] 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.

[15] [15] H. He and E. A. Garcia, “Learning from imbalanced data,” IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263–1284, 2009.

[16] [16] A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, “Calibrating probability with undersampling for unbalanced classification,” in Proc. IEEE SSCI, 2015.

[17] [17] 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.

[18] [18] A. Whitrow, D. J. Hand, P. Juszczak, D. Weston, and N. M. Adams, “Transaction aggregation as a strategy for credit card fraud detection,” Data Mining and Knowledge Discovery, vol. 18, pp. 30–55, 2009.

[19] [19] A. Carcillo, Y.-A. Le Borgne, O. Caelen, and G. Bontempi, “Combining unsupervised and supervised learning in credit card fraud detection,” Information Sciences, vol. 557, pp. 317–331, 2021.

[20] [20] A. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, “SCARFF: A scalable framework for streaming credit card fraud detection with Spark,” Information Fusion, vol. 41, pp. 182–194, 2018.

[21] [21] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[22] [22] S. Raschka and V. Mirjalili, Python Machine Learning, 3rd ed. Packt, 2019.