

Credit Card Fraud Detection System Using Machine Learning and Imbalanced Data Handling

Abhishek Rajan K¹, Nirupam Sen², Dhanush Shiju³, Rishikesh VV⁴, Mr. Vidwath⁵

^{1,2,3,4} Student, Department of Artificial Intelligence and Data Science, Shree Devi Institute of Technology, Mangaluru, Karnataka, India

⁵ Assistant Professor, Department of Computer Science, Shree Devi Institute of Technology, Mangaluru, Karnataka, India

Abstract—Digital payment channels have grown quickly, making card transactions part of daily commerce, yet the same growth has expanded opportunities for fraud. A key obstacle in building an automated detector is that legitimate transactions far outnumber fraudulent ones; a model trained naively leans toward the majority class and overlooks genuine fraud. This work proposes a machine-learning framework for credit card fraud detection that counters this imbalance with the Synthetic Minority Over-sampling Technique (SMOTE). Logistic Regression, Decision Tree, Random Forest and XGBoost, four supervised classifiers, are trained and compared on Accuracy, Precision, Recall, F1-score and ROC-AUC. In addition, the predicted fraud probability is translated into a three-level risk indicator (Low, Medium, High), showing how a model score can feed a practical decision workflow. XGBoost gave the best overall outcome among the four, with an F1-score of 86.57% on unseen test data.

Index Terms—Class Imbalance, Credit Card Fraud Detection, Machine Learning, Random Forest, Risk Classification, SMOTE, XGBoost.

I. INTRODUCTION

The move toward online banking, e-commerce and card-based payment services has sharply increased the number of electronic transactions processed daily. This brings speed and convenience but also opens new routes for unauthorized and deceptive use of payment credentials. Undetected fraud causes direct monetary losses for both cardholders and financial institutions, and repeated incidents slowly erode public trust in digital payments.

Machine learning lets a model learn behavioural patterns directly from historical transaction data, moving beyond static rule-based checks. Doing so is not simple, because fraudulent transactions usually form a tiny fraction of the dataset. A classifier trained on such skewed data can show a misleadingly high accuracy while still missing a large share of the fraud that matters most.

This project tackles the imbalance problem directly by applying SMOTE to the training portion of the data before fitting, and by comparing four widely used classification algorithms. Beyond plain Accuracy, the comparison uses Precision, Recall, F1-score and ROC-AUC, which describe more fully how well each model distinguishes fraudulent from legitimate transactions. The application accepts transaction details, outputs a fraud prediction with its probability, and maps that probability to a simple risk category.

II. RELATED WORK

Recent studies have applied many machine-learning methods to credit card fraud detection. Sharma et al. [1] surveyed machine- and deep-learning techniques for this task, emphasising how learning-based approaches can identify fraudulent transactions.

Aditi et al. [2] explored advanced machine-learning techniques for detecting card fraud, offering a comparative basis for judging different classification approaches in the field.

Bin Sulaiman et al. [3] surveyed machine-learning approaches to card fraud detection and also considered privacy, including a direction based on federated learning.

Devika et al. [4] applied Logistic Regression to card fraud detection, giving a simple, lightweight method that works well as a baseline.

Dileep et al. [5] examined Decision Tree and Random Forest methods, showing how tree-based models can pick out suspicious transaction behaviour. Together, these works motivate the comparison carried out here between Logistic Regression, Decision Tree, Random Forest and XGBoost, combined with explicit treatment of class imbalance through SMOTE [6].

III. PROBLEM STATEMENT

Credit card fraud detection is fundamentally about telling fraudulent transactions apart from legitimate ones using the details captured when each transaction occurs. The main difficulty is the uneven class distribution: since legitimate transactions dominate, a conventionally trained classifier learns the majority pattern too heavily and ends up missing a considerable portion of fraud.

Fraud also seldom reveals itself through one clear signal; it typically emerges from several transaction attributes acting in combination. A capable detector must therefore consider many features together instead of depending on a single rule or threshold.

To address both problems, the developed system combines careful preprocessing, SMOTE-based imbalance correction, a comparison of four classification algorithms, and evaluation with several metrics instead of accuracy alone.

IV. PROPOSED METHODOLOGY

The workflow passes through the following stages:

Transaction Data → *Preprocessing* → *Train/Test Split* → *SMOTE on Training Data* → *Model Training* → *Model Evaluation* → *Fraud Prediction* → *Risk Classification*

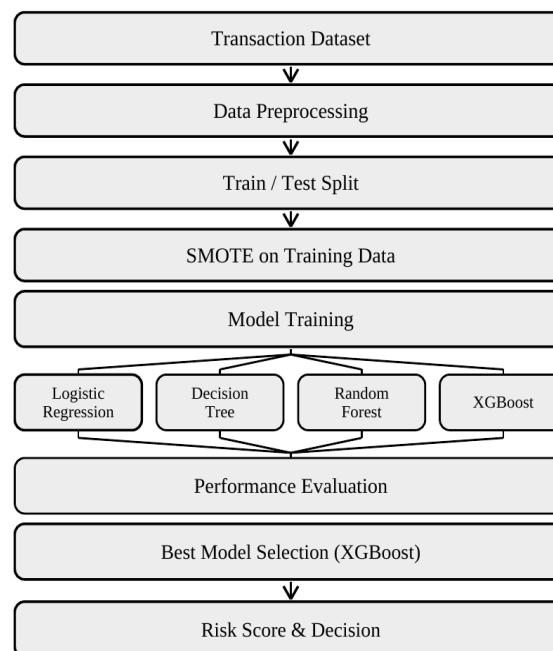


Fig. 1. System Workflow

The final system version (V3) uses sixteen transaction-level features:

- Amount
- Transaction Hour
- Merchant Category
- Foreign Transaction
- Location Mismatch
- Device Trust Score
- Velocity in Last 24 Hours
- Cardholder Age
- Average Transaction Amount
- Amount-to-Average Ratio
- Transactions in Last 7 Days
- Transactions in Last 24 Hours
- New Device
- Failed Attempts
- Distance from Usual Location
- Time Since Previous Transaction

Categorical attributes such as merchant category are encoded numerically in preprocessing, so the classification algorithms can work with them.

After preprocessing, the data is divided into training and test portions. SMOTE is used on the training portion alone. This leaves the test portion representative of genuine unseen transactions and

prevents synthetic samples from leaking into evaluation and inflating results.

Logistic Regression, Decision Tree, Random Forest [7] and XGBoost [8] are then fitted on the balanced training data, and their predictions on the held-out test set are compared using the measures described in the next section.

V. MODEL EVALUATION

Table I reports how the four classifiers in the final (V3) system version performed on the test set.

TABLE I TEST-SET PERFORMANCE OF THE CLASSIFIERS

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
Logistic Regression	96.80%	31.11%	93.33%	46.67%	99.305%
Decision Tree	98.40%	48.08%	83.33%	60.98%	91.341%
Random Forest	99.15%	78.26%	60.00%	67.92%	99.769%
XGBoost	99.55%	78.38%	96.67%	86.57%	99.956%

XGBoost achieved the highest F1-score of the four models and was chosen as the classifier for the final application. Its ROC-AUC [9] of 99.956% also points to strong separation of fraudulent and legitimate transactions on this dataset.

These figures come from this project only and should not be taken as a guarantee of the same results in a live banking setting, where transaction patterns and fraud tactics keep changing.

VI. RISK-BASED TRANSACTION ANALYSIS

Instead of stopping at a binary fraud/legitimate label, the application turns the fraud probability of the model into a risk score:

$$\text{Risk Score} = \text{Fraud Probability} \times 100$$

The score is then assigned to one of three prototype categories, loosely guided by probability-calibration considerations for imbalanced classifiers [10]:

Low Risk: 0–30 → Approve

Medium Risk: 31–70 → Additional Verification / OTP

High Risk: 71–100 → Block and Alert

These thresholds are design choices made for the prototype to illustrate the idea; they do not come from any established financial-industry standard. They exist to show how a raw probability can become an interpretable indicator that suggests an action for a transaction.

VII. RESULTS AND DISCUSSION

The four classifiers display clearly different trade-offs. Logistic Regression reached relatively high recall, catching most fraudulent transactions, yet its low precision means many of the transactions it flagged were actually legitimate.

Decision Tree scored better than Logistic Regression on accuracy but remained behind the ensemble methods on F1-score. Random Forest raised precision substantially, though its recall dropped below that of both Logistic Regression and XGBoost.

XGBoost recorded the top F1-score of 86.57%, with 78.38% precision, 96.67% recall and a ROC-AUC of 99.956% on the test set, and was therefore selected for the final application.

The finished application was also tried on representative inputs covering low-, medium- and high-risk cases, which confirmed the whole pipeline from raw input through classification, probability estimation and risk categorization.

VIII. IMPLEMENTATION ARCHITECTURE

The system is built as a series of data preparation, model inference and decision-support stages. Transaction attributes entered through the prototype interface are converted into the same feature representation used in training. This matters because a trained classifier expects identical feature order and numerical form at inference time.

Merchant category, the categorical field, is one-hot encoded during preprocessing, while numerical attributes stay numerical so the models can use them directly. The preprocessing objects are saved

alongside the trained model, so the application repeats exactly the transformation applied during training rather than building a different representation at prediction time.

XGBoost serves as the final classifier. It takes the processed transaction vector and outputs the probability of the fraud class. This probability is shown together with the predicted class, so the application conveys more than a bare binary label, and it is then converted into the prototype risk score for the decision layer.

Fig. 2 outlines the prediction path of the final application. A transaction goes through preprocessing and encoding and then the trained XGBoost model. The fraud probability obtained is turned into a numerical risk score and placed in the Low, Medium or High category. The actions shown in the diagram are prototype actions that illustrate how model output might fit into a transaction-review workflow.

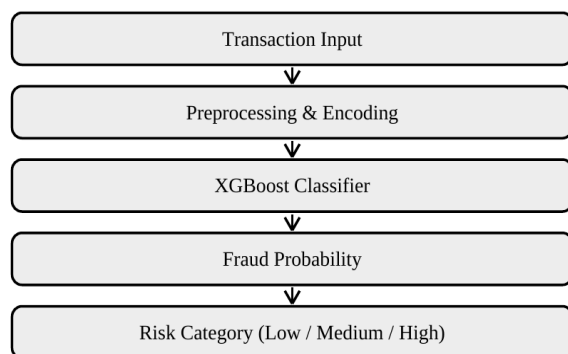


Fig. 2. Model Prediction and Risk Classification Flow

IX. EXPERIMENTAL SETUP AND FEATURE ANALYSIS

The V3 dataset has sixteen transaction-level attributes chosen to capture both immediate transaction properties and behavioural context. Amount and transaction hour give the basic monetary and time characteristics of an event, and merchant category names the type of merchant. Foreign transaction and location mismatch flag geographic irregularity, while device trust score expresses how trustworthy the device is considered.

The other attributes add behavioural context. Velocity in the last 24 hours, transactions in the last 24 hours and transactions in the last 7 days capture activity over different time windows. Average

transaction amount and the amount-to-average ratio compare the present transaction with a behavioural baseline. New device, failed attempts, distance from usual location and time since the previous transaction supply further signals that can help separate unusual activity from normal spending.

Data is split into training and test subsets before any model fitting, and SMOTE touches only the training subset. Keeping synthetic minority examples out of the test data means the final evaluation rests on observations that played no part in building the balanced training distribution. All four classifiers are scored on the same held-out test set, so the reported values can be compared directly within this experiment.

Evaluation relies on Accuracy, Precision, Recall, F1-score and ROC-AUC. Accuracy is the overall share of correct predictions; Precision is the share of predicted fraud cases that are truly fraudulent; Recall is the share of actual fraud the classifier detects. F1-score merges Precision and Recall through their harmonic mean, which helps when classes are imbalanced, and ROC-AUC summarizes how well the model separates the two classes across thresholds. The results reveal differing precision-recall trade-offs. Logistic Regression reached 93.33% recall but only 31.11% precision, so it caught many fraud cases while raising more false alarms. Random Forest gave 78.26% precision with lower recall of 60.00%, and Decision Tree gave 83.33% recall with 48.08% precision. XGBoost paired 78.38% precision with 96.67% recall and so achieved the highest F1-score of the four models.

X. LIMITATIONS AND FUTURE SCOPE

The results hold for the dataset and setup of this academic project and should not be read as production-level banking performance. Live payment systems handle constantly changing transaction streams and evolving fraud tactics, so a production deployment would need continuous monitoring, retraining, calibration, access controls and validation on current transaction distributions.

A further limitation is that the prototype takes manually entered transactions for demonstration. In practice, records would flow automatically from a payment or banking system. The application shows the machine-learning decision process without

reproducing the full infrastructure, authentication, authorization or transaction-processing stack of a financial institution.

The three risk bands are likewise prototype design choices. They show how a probability can be linked to an operational response, but their numerical boundaries are not offered as banking-industry standards. Future work can study probability calibration, cost-sensitive learning, threshold selection based on fraud and false-positive costs, and validation on larger, more diverse datasets.

Possible extensions include explainable AI methods such as SHAP to show which attributes drive an individual prediction, as well as real-time transaction ingestion, model monitoring, drift detection, stronger authentication for flagged transactions and periodic retraining. Such additions would bring the prototype closer to a complete fraud-analytics architecture while keeping the comparative modelling approach of this study.

XI. DEMONSTRATION SCENARIOS

The prototype was tested on representative transactions to check the complete route from user input to model output and risk category. These demonstrations show how the trained classifier responds to different attribute combinations; they are not meant as statistically estimated probabilities for a real banking population.

In the high-risk example, an amount of 500 was paired with a late transaction hour, an Electronics merchant category, a foreign-transaction indicator, a location mismatch, a low device trust score and fairly high transaction velocity. The application returned a fraud probability of about 98.89%, labelled the transaction Fraud and assigned the High Risk category, with the prototype action Block and Alert.

By contrast, the low-risk example used an amount of 50, a daytime hour, a Grocery merchant category, no foreign-transaction indicator, no location mismatch, a device trust score of 90 and low velocity. It received a fraud probability of about 0.01%, was labelled Legitimate and fell in the Low Risk category, with the prototype action Approve.

The medium-risk test used an amount of 376.74, transaction hour 19, Electronics as merchant category, foreign transaction and location mismatch both enabled, device trust score 27, velocity 1,

cardholder age 68, average transaction amount 460.06, amount-to-average ratio of roughly 0.819, 33 transactions in the previous 7 days, 5 in the previous 24 hours, no new device, 4 failed attempts, distance from usual location 42.59 and 162 units of time since the previous transaction. The resulting probability was about 51.29%, placing it in the Medium Risk category and showing the intermediate verification path.

These cases show why a probability output is valuable beyond a binary prediction: different feature combinations receive different levels of attention, giving the prototype a simple triage process. A real deployment would still need domain-specific rules, calibrated probabilities, authentication services, human review procedures and operational safeguards that lie outside this academic implementation.

XII. CONCLUSION

This paper described a machine-learning approach for identifying potentially fraudulent credit card transactions, combining data preprocessing, SMOTE-based imbalance correction [6] and a comparison of four classification algorithms.

The experiments confirm that the models offer different precision-recall trade-offs. XGBoost obtained the highest F1-score of the four and was therefore chosen for the final application.

Beyond the basic fraud/legitimate classification, the prototype adds a probability-based risk layer that turns a raw model score into an actionable category. Future work could add larger and more varied transaction datasets, real-time integration with live transaction streams, stronger authentication for flagged cases, better explainability and continued re-evaluation as fraud patterns evolve.

ACKNOWLEDGMENT

The authors sincerely thank Assistant Professor Vidwath for his guidance, support and valuable suggestions throughout the project and the preparation of this manuscript.

REFERENCES

- [1] S. Sharma, A. Kataria, J. K. Sandhu, and K. R. Ramkumar, "Credit Card Fraud Detection using

- Machine and Deep Learning Techniques," in 2022 3rd International Conference for Emerging Technology (INCET), pp. 1–7, 2022, doi: 10.1109/INCET54531.2022.9824065.
- [2] A. Aditi, A. Dubey, A. Mathur, and P. Garg, "Credit Card Fraud Detection Using Advanced Machine Learning Techniques," in 2022 5th International Conference on Computational Intelligence and Communication Technologies (CCICT), pp. 56–60, 2022.
- [3] R. Bin Sulaiman, V. Schetinin, and P. Sant, "Review of Machine Learning Approach on Credit Card Fraud Detection," Human-Centric Intelligent Systems, vol. 2, pp. 55–68, 2022, doi: 10.1007/s44230-022-00004-0.
- [4] M. Devika, R. K. Surapaneni, L. Sai Manohar, and N. Vijaya, "Credit Card Fraud Detection Using Logistic Regression," in 2022 Second International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE), 2022, doi: 10.1109/ICATIECE56365.2022.10046976.
- [5] M. R. Dileep, A. V. Navaneeth, and M. Abhishek, "A Novel Approach for Credit Card Fraud Detection using Decision Tree and Random Forest Algorithms," in 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), pp. 1026–1030, 2021, doi: 10.1109/ICICV50876.2021.9388431.
- [6] 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.
- [7] L. Breiman, "Random Forests," Machine Learning, vol. 45, pp. 5–32, 2001.
- [8] 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, pp. 785–794, 2016.
- [9] T. Fawcett, "An Introduction to ROC Analysis," Pattern Recognition Letters, vol. 27, no. 8, pp. 861–874, 2006.
- [10] 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.