IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

Online Payment Fraud Detection Using Machine Learning

Akshay Ajalsonde[1], Om Auti[2], Vedant Ghadge[3], Ajay Kapse[4], M. N. Pimple[5], K. D. Jamkar[6], Dr. H. B. Jadhav[7]
Students, Department of Computer Engineering[1-4]
Faculty, Department of Computer Engineering[5-7]
Adsul's Technical Campus, Ahilyanagar, Maharashtra, India
[1] sanjayajalsonde3@gmail.com

Abstract : *The rapid growth of digital payments, including UPI, cards, net banking and e-wallets, has been accompanied by a sharp rise in online payment fraud. Rule-based filters adapt slowly and raise many false alarms, while fraud itself is extremely rare, so automated detection is a highly class-imbalanced learning problem. This paper presents a complete machine learning framework for online payment fraud detection covering data preprocessing, imbalance handling, model training, threshold selection and evaluation with metrics suited to rare events. Six classifiers (Logistic Regression, Decision Tree, K-Nearest Neighbors, Random Forest, Gradient Boosting and a Multi-Layer Perceptron) and an unsupervised Isolation Forest are compared on a simulated dataset of 100,000 transactions with 1.25% fraud. At the default threshold, Random Forest gave the best F1-score (0.380), precision (0.507) and ROC-AUC (0.837), with a PR-AUC (0.336) practically equal to that of the neural network (0.332), which achieved the highest recall (0.565) at the price of low precision. Further experiments show that SMOTE and random oversampling raise PR-AUC to 0.434 and 0.457, that the decision threshold changes the precision-recall balance more than the choice of model, and that the unsupervised detector is close to random on this data. The results confirm that ensemble methods are well suited to rare-event fraud detection and that accuracy alone is a misleading measure in this domain.*

Keywords: online payment fraud, fraud detection, machine learning, class imbalance, SMOTE, random forest, gradient boosting, neural network, anomaly detection, precision-recall

I. INTRODUCTION

A. Background

Digital payments have changed how people and businesses move money. Debit and credit cards, net banking, mobile wallets and the Unified Payments Interface (UPI) now process very large numbers of transactions every day, and e-commerce, food delivery, ticketing, utility bills and person-to-person transfers have all moved online. The speed and convenience that make these channels attractive also make them attractive to criminals. A fraudster only needs a stolen card number, a one-time password obtained by deception, or control of a victim's phone to move money in seconds, often across several accounts, before the victim or the bank notices.

Online payment fraud harms every party involved. Customers lose money and time and may lose confidence in digital channels; merchants face chargebacks, penalties and reputational damage; and banks and payment providers bear investigation costs, reimbursement and regulatory pressure. Because payment volumes keep growing, fraud prevention has to be automated: no team of human analysts can review millions of transactions. The central question is how a system can decide, within milliseconds, whether a given payment is genuine or fraudulent.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

B. Common Types of Online Payment Fraud

Online payment fraud takes many forms, and a detection system has to cope with several of them at once. The most common ones are summarized below.

1) Card-not-present fraud: Stolen card numbers, expiry dates and security codes are used for online purchases where the physical card is not required. The data is usually obtained through data breaches, skimming or phishing pages.

2) Phishing and vishing: Victims are tricked by fake messages, websites or phone calls into revealing card details, passwords or one-time passwords (OTPs), which the fraudster then uses to authorize payments.

3) Account takeover: An attacker gains control of a genuine account, for example through credential stuffing or a compromised device, and then changes details or transfers funds. The transactions look like they come from a legitimate customer.

4) SIM swap and device fraud: The victim's mobile number is moved to a SIM controlled by the attacker, who then receives OTPs and approves payments. Malicious applications that read messages or share the screen have a similar effect.

5) UPI collect-request and QR scams: Victims are persuaded to approve a collect request or scan a QR code believing they will receive money, and the money is debited instead.

6) Merchant and identity fraud: Fake merchants collect payments and disappear, while synthetic or stolen identities are used to open accounts that later cash out or act as mule accounts for moving stolen funds.

C. Why Machine Learning

Traditional fraud control uses fixed rules, such as blocking transactions above a certain amount, from a new country, or beyond a certain number per hour. Rules are simple and transparent, but fraudsters learn them and stay just below the limits, the rules must be updated by hand, and strict rules block many genuine customers. Machine learning (ML) instead learns the patterns that separate fraud from normal behaviour directly from historical transactions. It can combine hundreds of weak signals, detect interactions that no analyst would write as a rule, and be retrained as behaviour changes [2], [3], [17].

D. Challenges

Fraud detection is harder than a typical classification task for several reasons. (i) Extreme class imbalance: fraud is usually far below one percent of transactions, so a classifier can reach very high accuracy by predicting everything as genuine while detecting nothing [12]. (ii) Unequal costs: a missed fraud costs the transaction value plus investigation, while a false alarm annoys a customer and may lose the sale. (iii) Concept drift: fraudsters change tactics, so the data distribution changes over time [2]. (iv) Delayed and incomplete labels: a fraud label often arrives days or weeks later, when the customer disputes the payment. (v) Real-time constraints: a decision must be made within a few milliseconds. (vi) Privacy: transaction data is sensitive and rarely shared, which makes open research data scarce.

E. Objectives and Contributions

The objectives of this work are: (1) to build a complete and reproducible ML pipeline for online payment fraud detection; (2) to compare several supervised classifiers and one unsupervised detector using metrics suited to imbalanced data; (3) to study how different imbalance-handling techniques affect results; (4) to analyse how the decision threshold controls the trade-off between caught fraud and false alarms; and (5) to discuss what is needed to move such a system towards real deployment. The paper contributes a systematic experimental comparison, a threshold and imbalance analysis, and practical guidelines.

F. Organization of the Paper

Section II reviews related work. Section III describes the proposed system, dataset, algorithms and metrics. Section IV presents and discusses the results. Section V covers deployment considerations, Section VI lists limitations, and Section VII concludes the paper and suggests future work.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

II. LITERATURE REVIEW

A. Rule-Based and Statistical Approaches

Early fraud detection systems were expert systems of hand-written rules combined with statistical profiles of customer behaviour. Surveys of data mining-based fraud detection show that the field moved from such rules to learning-based methods as transaction volumes and fraud sophistication grew [17]. Rules are still used in production, but mainly as a fast first filter and as a way to encode business policy and regulation.

B. Supervised Machine Learning

Supervised methods learn from labelled examples of fraud and genuine transactions. Bhattacharyya et al. compared logistic regression, support vector machines and random forests on real credit card data and found that the random forest generally performed best [3]. Random forests [5] and gradient boosted trees [13], including highly optimized implementations such as XGBoost [6], are popular because they handle non-linear relationships, mixed feature types and missing values with little preprocessing and score a transaction in microseconds. Neural networks can learn complex patterns, but need more data and tuning and are harder to explain.

C. Handling Class Imbalance

Because fraud is rare, many studies focus on rebalancing. Random undersampling removes genuine transactions, making training fast but discarding information, and it shifts the predicted probabilities, which can be corrected by calibration [1]. Random oversampling duplicates fraud cases and risks overfitting. SMOTE creates synthetic minority samples by interpolating between a fraud case and one of its nearest fraud neighbours [4]. Cost-sensitive learning gives errors on the fraud class a larger weight instead of changing the data. He and Garcia give a broad review of these and related methods for imbalanced learning [12].

D. Unsupervised and Hybrid Approaches

When labels are missing or delayed, anomaly detection can flag transactions that differ from normal behaviour [14]. Isolation Forest isolates points with random splits, so that anomalies need fewer splits [7]. Unsupervised scores are often useful as additional features for a supervised model, and hybrid approaches combining both have been reported to improve detection [15].

E. Deep Learning and Sequence Models

Fraud depends on behaviour over time, for example a sudden burst of small payments followed by a large one. Recurrent models such as LSTM networks [8] can learn such sequential patterns from a customer's history. They are more costly to train and deploy, and they are harder to explain to investigators and regulators, so explainability tools such as SHAP [16] are commonly used with complex models.

F. Evaluation of Imbalanced Problems

Accuracy is misleading when 98 to 99.9 percent of cases are genuine. The ROC curve [10] is widely used, but with extreme imbalance it can look optimistic because the large number of true negatives keeps the false-positive rate small. Saito and Rehmsmeier showed that the precision-recall curve is more informative in this situation [9]. Dal Pozzolo et al. also stress that fraud detection should be judged by how well the highest-ranked alerts match what investigators can actually review, and that evaluation must respect the time order of data [2].

TABLE I

Summary of Related Work Used in This Study

Ref.	Technique / Topic	Main Idea	Relevance to This Work
[1]	Undersampling and calibration	Undersampling shifts probabilities; correct by calibration	Motivates threshold analysis
[2]	Realistic fraud learning	Delayed labels, drift, ranking-based metrics	Evaluation and deployment issues

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 3, November 2026 Impact Factor: 8.2

[3]	Classifier comparison	Random forest compared with SVM and logistic regression	Supports use of ensembles
[4]	SMOTE	Synthetic minority samples by interpolation	Imbalance experiment
[5], [6], [13]	Tree ensembles	Bagging and boosting of decision trees	Core models
[7],[14]	Anomaly detection	Isolation Forest, survey of anomalies	Unsupervised baseline
[8],[16]	LSTM and SHAP	Sequence learning and explanations	Future work
[9],[10], [12]	Metrics and imbalance	PR curves, ROC, imbalanced learning	Evaluation design

G. Research Gap

Many papers report a single high score on one benchmark and give little guidance on how the choice of metric, resampling method and decision threshold changes the picture. This paper does not claim a new algorithm. Its aim is a clear, reproducible comparison that shows how these choices interact on one imbalanced problem and what a practitioner should check before trusting a number.

III. PROPOSED SYSTEM AND METHODOLOGY

A. System Architecture

Fig. 1 shows the proposed pipeline. Transactions arrive as a stream and pass through preprocessing and feature engineering. The training branch applies imbalance handling and trains and tunes the models. At run time, the trained model assigns each transaction a fraud score, a threshold converts the score into an action (allow, send for manual review, or block), and confirmed fraud labels from investigators and customer disputes are fed back for periodic retraining.

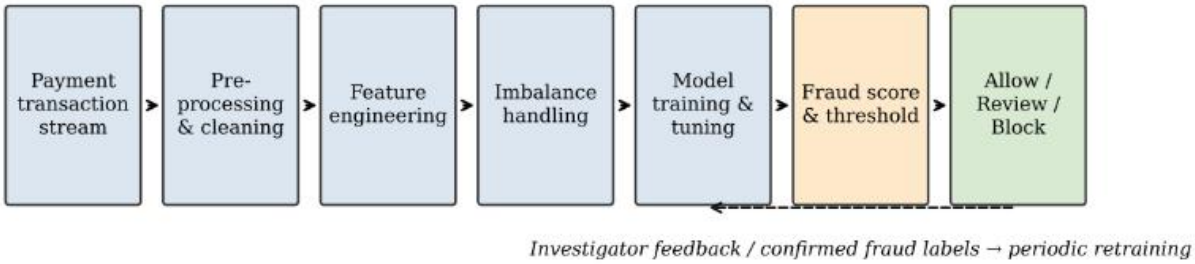


Fig. 1 Architecture of the proposed online payment fraud detection system

B. Dataset

Real payment data is confidential and rarely shared, so a simulated dataset was generated with the make_classification function of Scikit-learn [11] using a fixed random seed for reproducibility. It contains 100,000 transactions and 30 numerical features: 12 informative features, 6 redundant features that are linear combinations of the informative ones, and 12 noise features. Each class is built from several clusters, the classes deliberately overlap (class separation 0.9), and 0.5% of the labels are flipped at random to imitate labelling errors. The fraud rate is 1.25% (1,250 fraud cases), so the data is heavily imbalanced, though less so than some real systems.

The data was split with stratified sampling into 70% for training (70,000 records, 875 fraud) and 30% for testing (30,000 records, 375 fraud). The test set was never resampled and was not used for training or feature scaling, so it represents the real-world fraud rate. Table II summarizes the data.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

TABLE II
Summary of the Experimental Dataset

Item	Training Set	Test Set
Transactions	70,000	30,000
Fraud cases	875	375
Genuine cases	69,125	29,625
Fraud rate	1.25%	1.25%
Features	30	30

Because the features are anonymous numbers, the study shows how methods behave on a hard, imbalanced problem but cannot tell which real-world attributes matter. A real deployment would use the fields shown in Table III.

TABLE III
Typical Features Used in Real Payment Fraud Systems

Feature Group	Examples
Transaction	Amount, currency, payment method, merchant category, channel (app, web, QR)
Time	Hour of day, day of week, time since the previous transaction
Customer behaviour	Average and maximum amount, count and total in the last hour, day and week
Device and network	Device identifier, IP address, location, new-device flag, distance from usual location
Account	Account age, recent change of phone number, password or address, number of beneficiaries added
Graph and risk	Link to known fraudulent beneficiaries, merchant risk score, number of accounts sharing a device

C. Preprocessing and Feature Scaling

All features were standardized to zero mean and unit variance using statistics computed on the training set only, then applied to the test set. This avoids data leakage and is needed for distance-based and gradient-based models such as KNN, logistic regression and the neural network, while tree-based models are not affected by scaling. In a real system, categorical attributes would also be encoded, missing values imputed, skewed amounts log-transformed, and behavioural aggregates computed over sliding time windows.

D. Handling Class Imbalance

Five training strategies were compared with the same Random Forest:

- **No resampling:** the original imbalanced training data (70,000 records).
- **Class weighting:** errors on the fraud class are weighted in inverse proportion to the class frequency in each tree's bootstrap sample.
- **Random undersampling:** all 875 fraud cases are kept and genuine cases are randomly reduced to a 5:1 ratio, giving 5,250 records.
- **Random oversampling**: fraud cases are randomly duplicated until they equal 20% of the genuine count (82,950 records).

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

- **SMOTE:** synthetic fraud cases are created until fraud equals 20% of the genuine count (82,950 records). For a minority sample x and one of its k = 5 nearest minority neighbours x', a synthetic point is x_new = x + λ (x' − x), with λ drawn uniformly from [0, 1] [4].

SMOTE was implemented directly with a nearest-neighbour search because the imbalanced-learn package was not available. Undersampling was used as the default strategy for the model comparison in Section IV-A because it makes training fast. All resampling was applied to the training data only.

E. Classification Algorithms

Six supervised models were trained with Scikit-learn [11]. Their main settings are listed in Table IV.

1) Logistic Regression: A linear model that estimates the probability of fraud with the sigmoid function:

$$P(y = 1 \mid x) = 1 / (1 + e^{-(w^T x + b)}) \qquad (1)$$

The weights w and bias b are learned by minimizing the log-loss. The model is fast and easy to explain, but it can only draw a linear boundary between the classes.

2) Decision Tree: A tree of yes/no questions on features. At each node the split that most reduces the Gini impurity is chosen:

$$Gini = 1 - \Sigma\, p_k^2 \qquad (2)$$

where p_k is the share of class k in the node. Trees are easy to read as rules but a single tree is unstable and overfits unless its depth is limited.

3) K-Nearest Neighbors: A transaction is classified by the majority class among its k closest training transactions in feature space, using Euclidean distance:

$$d(x, z) = \sqrt{\Sigma\, (x_j - z_j)^2} \qquad (3)$$

KNN needs no training, but it stores all training data and its prediction cost grows with the data size. It also suffers when many features are noisy.

4) Random Forest: An ensemble of T decision trees, each trained on a bootstrap sample of the data and using a random subset of features at every split [5]. The final fraud probability is the average of the trees:

$$P(y = 1 \mid x) = (1 / T)\, \Sigma\, P_t(y = 1 \mid x) \qquad (4)$$

Averaging many different trees reduces variance and overfitting. Here T = 200.

5) Gradient Boosting: Trees are added one at a time, each fitted to the errors (negative gradient of the loss) of the current model [13]:

$$F_m(x) = F_{(m-1)}(x) + \eta\, h_m(x) \qquad (5)$$

where h_m is the new tree and η the learning rate. Boosting often gives high accuracy, but it is more sensitive to noisy labels than bagging.

6) Multi-Layer Perceptron: A feed-forward neural network with two hidden layers of 64 and 32 neurons and ReLU activation, trained with the Adam optimizer, ending with a sigmoid output that gives the fraud probability.

7) Isolation Forest: An unsupervised detector trained without labels on the whole training set [7]. Points that are isolated by few random splits get a high anomaly score. The 1.25% most anomalous test transactions were flagged, corresponding to the known fraud rate, which is an optimistic assumption because the true rate is usually unknown.

TABLE IV: Model Settings Used in the Experiments

Model	Main Settings
Logistic Regression	L2 penalty, maximum 1,000 iterations
Decision Tree	Gini criterion, maximum depth 8
KNN	k = 5, Euclidean distance
Random Forest	200 trees, other settings default

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

Gradient Boosting	100 trees (default), learning rate 0.1, depth 3
Neural Network (MLP)	Hidden layers (64, 32), ReLU, Adam, maximum 300 iterations
Isolation Forest	200 trees, contamination 0.0125

All models used a fixed random seed (42). No hyperparameter search was performed, so the results show out-of-the-box behaviour and tuning would likely improve every model.

F. Evaluation Metrics

With 1.25% fraud, a model that labels every transaction genuine reaches 98.75% accuracy and detects no fraud, so accuracy is not reported. The metrics are defined from true positives (TP, fraud correctly caught), false positives (FP, genuine blocked), false negatives (FN, fraud missed) and true negatives (TN):

$$\text{Precision} = TP / (TP + FP) \quad (6)$$

$$\text{Recall} = TP / (TP + FN) \quad (7)$$

$$F1 = 2 \cdot \text{Precision} \cdot \text{Recall} / (\text{Precision} + \text{Recall}) \quad (8)$$

Precision is the share of raised alerts that are real fraud, which relates to the burden on customers and investigators. Recall is the share of all fraud that is caught, which relates to the money saved. The F1-score balances the two. Two threshold-independent measures describe how well the model ranks transactions: ROC-AUC, the area under the curve of true-positive rate against false-positive rate [10], and PR-AUC, the area under the precision-recall curve, estimated here by average precision. For a random classifier PR-AUC equals the fraud rate (0.0125), while ROC-AUC equals 0.5, which makes PR-AUC the more demanding measure on imbalanced data [9].

G. Experimental Procedure

The experiments were run in Python 3 with Scikit-learn [11] on a CPU-only machine. The procedure was: (1) generate and split the data; (2) scale the features; (3) resample the training set; (4) train each model; (5) score the untouched test set and compute the metrics at a threshold of 0.5; (6) repeat training with five-fold stratified cross-validation on the training set (resampling inside each fold) to measure stability; (7) compare the imbalance strategies; (8) vary the threshold; and (9) record training and prediction time.

IV. RESULTS AND DISCUSSION

A. Overall Performance of the Classifiers

Table V lists the results on the test set of 30,000 transactions (375 fraud) with the threshold at 0.5, and Fig. 2 compares precision, recall and F1-score. Table VI gives the corresponding confusion-matrix counts.

TABLE V: Performance of the Classifiers on the Test Set

Model	Precision	Recall	F1-Score	ROC-AUC	PR-AUC
Logistic Regression	0.167	0.211	0.186	0.734	0.111
Decision Tree	0.082	0.355	0.134	0.666	0.050
KNN	0.255	0.261	0.258	0.773	0.166
Random Forest	0.507	0.304	0.380	0.837	0.336
Gradient Boosting	0.277	0.360	0.313	0.818	0.277
Neural Network (MLP)	0.090	0.565	0.155	0.811	0.332

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

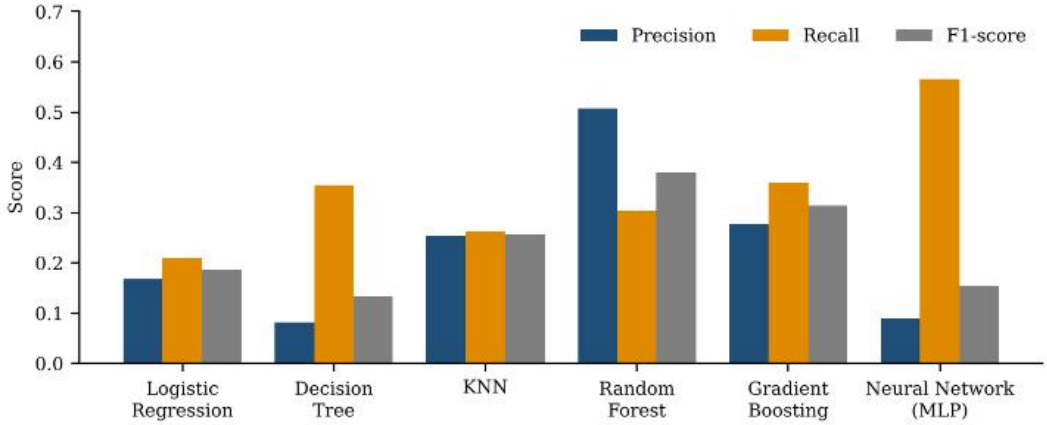


Fig. 2 Precision, recall and F1-score of the six classifiers at a threshold of 0.5

TABLE VI" Confusion Matrix Counts on the Test Set

Model	TP	FP	FN	TN
Logistic Regression	79	394	296	29231
Decision Tree	133	1481	242	28144
KNN	98	287	277	29338
Random Forest	114	111	261	29514
Gradient Boosting	135	352	240	29273
Neural Network (MLP)	212	2147	163	27478

Random Forest achieved the best precision (0.507), F1-score (0.380) and ROC-AUC (0.837). It raised 225 alerts, of which 114 were real fraud, so about half of its alerts were correct, and it produced the fewest false alarms (111) of all six models. Gradient Boosting was next in ranking quality (ROC-AUC 0.818, PR-AUC 0.277) but raised 352 false alarms, more than three times as many. The Decision Tree caught 133 fraud cases, yet it also raised 1481 false alarms (precision 0.082), which would be unacceptable for customers. Logistic Regression and KNN, the simpler models, were clearly weaker at ranking transactions.

The neural network behaved differently from the rest. Its PR-AUC (0.332) is almost the same as that of Random Forest (0.336), meaning that it ranks transactions about equally well, and at the default threshold it caught the most fraud (recall 0.565, 212 of 375 cases). However, it did so by raising 2359 alerts, of which only 212 were fraud (precision 0.090). This shows that the F1-score at one fixed threshold can hide how close two models really are, and it prepares the threshold analysis in Section IV-E.

B. ROC and Precision-Recall Curves

Fig. 3 shows the curves for all models. In the ROC plot, Random Forest, Gradient Boosting and the neural network lie close together and above the other models, and the plot suggests good performance because the false-positive rate stays small for any model when genuine transactions are so numerous. The precision-recall plot separates the models better. Random Forest and the neural network keep the highest precision over most of the recall range, Gradient Boosting follows, and the other three models fall quickly towards the baseline of 0.0125 (the fraud rate). At 20% recall, for example, Random Forest and the neural network still have a precision of roughly 0.7 to 0.8 and Gradient Boosting about 0.5 to 0.6, while KNN is at about 0.5 and Logistic Regression and the Decision Tree are at only 0.1 to 0.2.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

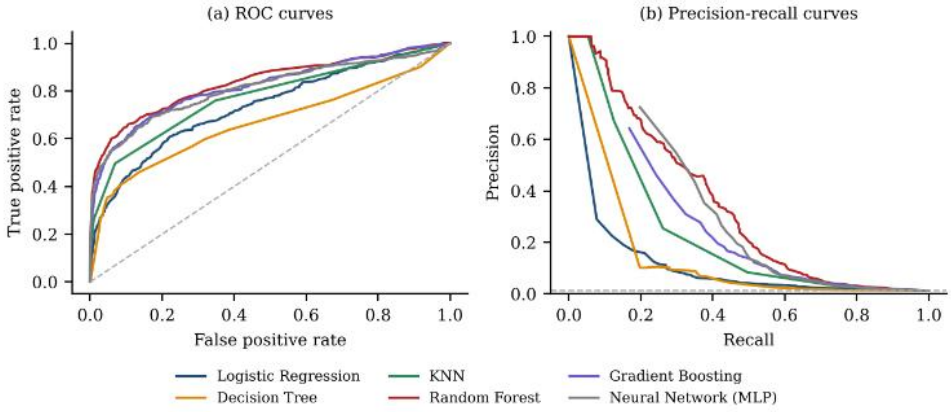


Fig. 3 (a) ROC curves and (b) precision-recall curves of the six classifiers on the test set

C. Stability Across Cross-Validation Folds

To check that the ranking is not an accident of one split, each model was also evaluated with five-fold stratified cross-validation on the training data (Table VII). Random Forest again had the highest F1-score (0.348 ± 0.033), and its PR-AUC (0.324 ± 0.021) is within one standard deviation of the neural network (0.335 ± 0.027), so these two cannot be separated. Both were clearly above Gradient Boosting, KNN, Logistic Regression and the Decision Tree. The standard deviations are small compared with the gaps between the top group and the rest, so that part of the ranking is stable.

TABLE VII

Five-Fold Cross-Validation Results (Mean ± Standard Deviation)

Model	Precision	Recall	F1-Score
Logistic Regression	0.136 ± 0.023	0.171 ± 0.020	0.151 ± 0.021
Decision Tree	0.071 ± 0.009	0.346 ± 0.019	0.117 ± 0.012
KNN	0.246 ± 0.031	0.265 ± 0.047	0.254 ± 0.036
Random Forest	0.506 ± 0.027	0.265 ± 0.031	0.348 ± 0.033
Gradient Boosting	0.245 ± 0.023	0.303 ± 0.026	0.271 ± 0.023
Neural Network (MLP)	0.102 ± 0.004	0.577 ± 0.010	0.174 ± 0.006

Note that the cross-validation values are slightly lower than the test-set values for Random Forest, as can be expected, because each fold is trained on only 80% of the training data.

D. Effect of Imbalance-Handling Strategy

Table VIII compares the five strategies with the same Random Forest, and Fig. 4 shows PR-AUC and ROC-AUC. Two observations stand out.

TABLE VIII: Random Forest With Different Imbalance-Handling Strategies

Strategy	Train Size	Precision	Recall	F1 (0.5)	PR-AUC	Best F1
No resampling	70,000	1.000	0.035	0.067	0.358	0.411
Class weight (balanced)	70,000	1.000	0.019	0.037	0.365	0.411
Random undersampling (5:1)	5,250	0.507	0.304	0.380	0.336	0.408
Random oversampling	82,950	1.000	0.056	0.106	0.457	0.497
SMOTE (k=5)	82,950	0.829	0.181	0.298	0.434	0.474

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

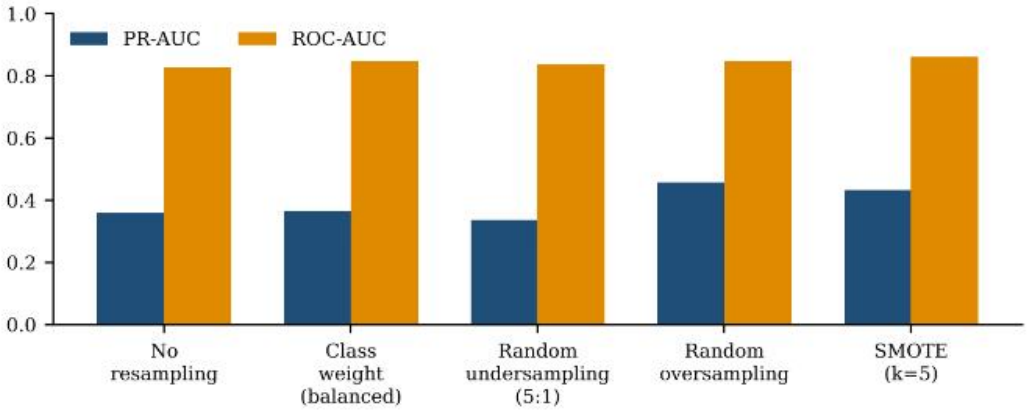


Fig. 4 PR-AUC and ROC-AUC of Random Forest under different imbalance-handling strategies

First, at the default threshold of 0.5 the strategies that keep many genuine cases and add little to the fraud class (no resampling, class weights, and to a large degree oversampling) are very precise (precision 0.83 to 1.00) but find very little fraud (recall 0.02 to 0.18). This is the typical sign of a model whose probabilities are pushed towards the majority class by the 1.25% prior. Undersampling moves the probabilities up, which is why its F1 at 0.5 is the highest of the group (0.380), but this is a property of the threshold, not of better learning.

Second, the threshold-independent PR-AUC tells a different story. Random oversampling (0.457) and SMOTE (0.434) rank the transactions best, no resampling and class weights are close behind (0.358 and 0.365), and undersampling is the lowest (0.336), which is consistent with it discarding more than 92% of the genuine training records. The last column shows the best F1 that can be reached by choosing the threshold. Oversampling (0.497) and SMOTE (0.474) are best again, while the other methods reach about 0.41. These best-F1 values were selected on the test set and are therefore optimistic. In practice the threshold must be chosen on a separate validation set.

E. Effect of the Decision Threshold

A classifier outputs a fraud probability, and the threshold decides which transactions are flagged. Table IX and Fig. 5 show the Random Forest (trained with undersampling) at six thresholds. A low threshold of 0.2 catches 70.7% of the fraud but only 4.8% of the alerts are real. A high threshold of 0.7 gives 91.9% precision but catches only 9.1% of the fraud. The F1-score peaks near 0.5, but this depends on the relative importance given to precision and recall.

TABLE IX

Random Forest Performance at Different Decision Thresholds

Threshold	Precision	Recall	F1-Score	Alerts Raised
0.2	0.048	0.707	0.089	5547
0.3	0.129	0.568	0.210	1654
0.4	0.306	0.459	0.367	563
0.5	0.507	0.304	0.380	225
0.6	0.683	0.189	0.296	104
0.7	0.919	0.091	0.165	37

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

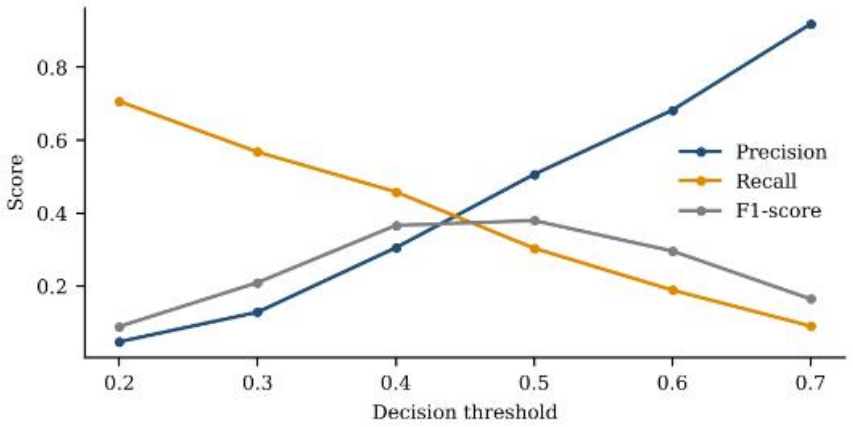


Fig. 5 Precision, recall and F1-score of Random Forest as the decision threshold changes

The same model therefore ranges from a detector that finds most fraud but floods investigators with alerts to one that is almost always right but misses most fraud, without any change in learning. The right threshold depends on cost. If the cost of a missed fraud is c_FN and the cost of a false alarm is c_FP, the expected cost is c_FN · FN + c_FP · FP, and the threshold should minimize this on validation data. Banks also have a limited review capacity, which sets an upper bound on the number of alerts per day. In addition, because undersampling changes the class prior from 1.25% to about 16.7%, the raw scores are not true probabilities and should be calibrated before being compared with business costs [1].

F. Feature Importance

Fig. 6 shows the ten most important features of the Random Forest, measured by the mean decrease in impurity. Feature 28 is the most important (0.086), followed by features 4, 7 and 17 (about 0.045 to 0.048). The importance is spread over many features rather than concentrated in a few, as expected from the way the data was generated, with 12 informative features and several redundant ones. With real data, such a ranking helps analysts understand which attributes drive the model. The impurity-based measure is biased towards features with many distinct values, so SHAP values [16] or permutation importance should be used to explain individual decisions.

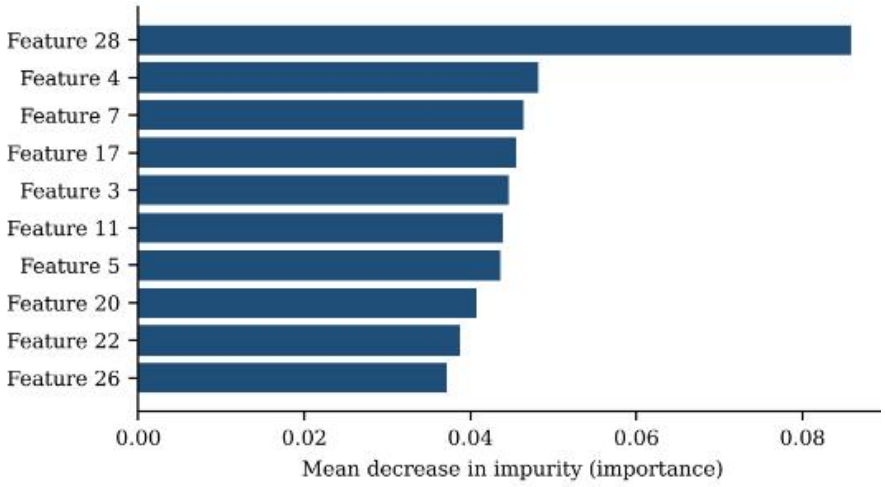


Fig. 6 Ten most important features of the Random Forest model

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

G. Unsupervised Detection

The Isolation Forest, trained without labels, achieved a ROC-AUC of only 0.554 and a PR-AUC of 0.015, which is close to the random baseline of 0.0125. It flagged 376 test transactions, of which only 8 were fraud. The reason is that in this dataset fraud is not a set of far-away outliers: fraud cases form clusters that overlap with genuine behaviour, so a distance- or isolation-based anomaly score cannot separate them. This does not mean unsupervised detection is useless in practice. It is valuable when no labels exist, for new fraud patterns, and as an extra input feature for a supervised model [15]. It does mean that labels, when available, bring a very large gain.

H. Computational Cost

Table X shows the time to train each model on the 5,250-record undersampled training set and the time to score 1,000 transactions on the same machine. All models score 1,000 transactions in under 10 milliseconds, which is fast enough for real-time use, although the numbers will differ on other hardware and with larger data and feature sets. Random Forest, Gradient Boosting and the neural network needed 2 to 3 seconds for training, so frequent retraining is practical for this data size.

TABLE X

Training and Prediction Time (CPU)

Model	Training Time (s)	Scoring Time per 1,000 (ms)
Logistic Regression	0.01	0.06
Decision Tree	0.09	0.08
KNN	0.00	8.37
Random Forest	1.80	7.35
Gradient Boosting	3.10	1.23
Neural Network (MLP)	2.36	0.56

I. Discussion

Several practical lessons follow from the experiments. First, ensembles of trees were the most reliable choice: Random Forest was the best or tied for best on every measure except recall, and it needs little preprocessing. Second, the neural network ranks fraud nearly as well, and with a suitable threshold it could be an equal alternative, but it requires scaled inputs and is harder to explain. Third, the imbalance strategy matters, but it should be judged by threshold-independent measures: undersampling looks good at 0.5 only because it moves the probabilities, while oversampling and SMOTE give the better ranking. Fourth, the threshold has at least as large an effect as the choice of algorithm, so it should be treated as a business decision based on costs and review capacity. Finally, labels are essential: the unsupervised detector was close to random on this data.

The absolute scores are modest. Recall of 30% to 36% at the default threshold means most fraud would still be missed. This is expected because the simulated classes overlap strongly, 12 of the 30 features carry no signal, 0.5% of the labels are wrong, and no hyperparameter tuning or feature engineering was done. The ranking of methods and the trends are the main findings and not the absolute numbers, and the results should be confirmed on real transaction data before any claim about production performance is made.

V. DEPLOYMENT CONSIDERATIONS

1) Real-time scoring: A payment must be approved or declined within a few hundred milliseconds in total, leaving only a few milliseconds for the fraud model. Compact tree ensembles and logistic models meet this easily, while very large models need dedicated serving infrastructure. Feature computation, such as aggregates over recent transactions, usually takes more time than the model itself and needs a fast feature store.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 3, November 2026 Impact Factor: 8.2

2) Human-in-the-loop decisions: Rather than a simple allow or block, most systems use three outcomes: allow low-risk payments, send medium-risk payments for step-up authentication or manual review, and block only the highest-risk ones. This turns the threshold problem into two thresholds and reduces the harm of false alarms.

3) Concept drift and retraining: Fraud patterns change as criminals adapt, so model performance falls over time. The system should monitor precision, recall and the distribution of scores, retrain regularly on recent data, and use time-based validation, where the model is trained on the past and tested on the future, as recommended in [2].

4) Delayed labels: Confirmed fraud labels arrive after customer complaints, often with a delay of days or weeks. Recent transactions are therefore under-labelled, and investigators' feedback on reviewed alerts is a more timely but biased source.

5) Explainability: Investigators, customers and regulators need to know why a payment was blocked. Feature-importance tools and SHAP-style explanations [16] can provide reason codes, and simpler models may be preferred when explanations are required by policy.

6) Privacy and security: Transaction and device data is personal data and must be protected and used in line with applicable data-protection and banking rules. Data minimization, access control, encryption and anonymization of training data are standard measures. The model itself can also be attacked: fraudsters can probe the system and adapt, so error messages and decision details should not reveal the rules.

7) Fairness: Models should be checked so that they do not systematically flag customers from certain regions, devices or income groups without a legitimate reason, since false alarms have a real cost for the affected customers.

VI. LIMITATIONS

This study has several limitations that readers should keep in mind. (1) The data is simulated and anonymous, so the results cannot be taken as the performance on real payments, which have time structure, categorical fields, repeat customers and different kinds of fraud. (2) Only one dataset and one train-test split were used for the main tables, although cross-validation supports the main ranking. (3) The models were not tuned, and tuning may change the ranking of close models, in particular Random Forest and the neural network. (4) The best-F1 values in Table VIII were selected on the test set and are optimistic. (5) Time order, concept drift and delayed labels were not simulated. (6) Gradient boosting libraries such as XGBoost, sequence models such as LSTM, and graph-based methods were not evaluated. (7) The cost of errors was discussed but not quantified, because no real cost data was available.

VII. CONCLUSION AND FUTURE WORK

This paper presented a machine learning framework for online payment fraud detection and evaluated six supervised classifiers and one unsupervised detector on an imbalanced simulated dataset with 1.25% fraud. Random Forest gave the best F1-score, precision and ROC-AUC at the default threshold and the most stable results in cross-validation, with the neural network showing equal ranking quality and the highest recall. SMOTE and random oversampling improved the ranking quality over undersampling, the decision threshold changed the balance between caught fraud and false alarms more than the choice of model, and the unsupervised Isolation Forest was close to random on this data. The study also shows that accuracy is misleading, and that PR-AUC, recall at a chosen precision and the cost of errors are the measures that matter.

Future work will include: (i) evaluation on real or public benchmark transaction data with time-based splitting; (ii) hyperparameter tuning and comparison with XGBoost and other boosting libraries; (iii) cost-sensitive learning and threshold selection on a validation set using real cost estimates; (iv) sequence models such as LSTM and graph-based features to capture customer behaviour and mule networks; (v) hybrid supervised and unsupervised detection; (vi) concept-drift detection and online learning; and (vii) explainable AI and fairness analysis for deployment.

ACKNOWLEDGMENT

The authors thank [Institution Name] for guidance and support during this work.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

REFERENCES

[1] A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in Proc. IEEE Symp. Series on Computational Intelligence (SSCI), 2015, pp. 159–166.
[2] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: A realistic modeling and a novel learning strategy," IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 8, pp. 3784–3797, Aug. 2018.
[3] S. 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.
[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.
[5] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
[6] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794.
[7] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in Proc. IEEE Int. Conf. Data Mining (ICDM), 2008, pp. 413–422.
[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[9] 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, e0118432, 2015.
[10] T. Fawcett, "An introduction to ROC analysis," Pattern Recognition Letters, vol. 27, no. 8, pp. 861–874, 2006.
[11] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.
[12] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Trans. Knowl. Data Eng., vol. 21, no. 9, pp. 1263–1284, Sept. 2009.
[13] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, 2001.
[14] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Computing Surveys, vol. 41, no. 3, Art. 15, 2009.
[15] F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oble, and G. Bontempi, "Combining unsupervised and supervised learning in credit card fraud detection," Information Sciences, vol. 557, pp. 317–331, 2021.
[16] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems 30, 2017, pp. 4765–4774.
[17] C. Phua, V. Lee, K. Smith, and R. Gayler, "A comprehensive survey of data mining-based fraud detection research," arXiv preprint arXiv:1009.6119, 2010.