Welfare of Masses: Peer-Reviewed Multidisciplinary Research Journal (WMRJ)

Journal Website - www.journal.mmcc.edu.in/wmrj

Credit Card Fraud Detection Using Machine Learning: A Comparative Study of Logistic Regression and Random Forest with SMOTE-Based Class Balancing

Saurabh Dodke[1], Ms. Sneha Vanjari[2]

[1]Department of Data Science
Marathwada Mitra Mandal's College of Engineering
Pune, India
saurabhdodke9090@gmail.com

[2]Department of Information Technology
Marathwada Mitra Mandal's College of Engineering
Pune, India
snehavanjari@mmcoe.edu.in

ARTICLE INFO

Article history:

Received: *04 June 2026*

Received in revised form: *16 July 2026*

Accepted: *4 August 2026*

Keywords:
Credit card fraud detection, machine learning, SMOTE, logistic regression, random forest, class imbalance, supervised learning,

ABSTRACT

Credit card fraud remains a persistent and costly problem for global financial systems, with losses estimated in the billions of dollars annually. Effective automated fraud detection is essential for protecting financial institutions and consumers, but such systems must contend with severely imbalanced transaction data, since fraudulent transactions typically represent only a small fraction of total transaction volume. This paper presents a systematic comparison of two supervised learning approaches, Logistic Regression and Random Forest, for detecting fraudulent credit card transactions. The Synthetic Minority Over-sampling Technique (SMOTE) is applied to the training data to mitigate class imbalance prior to model fitting. Experiments are conducted on the widely used European credit card transaction dataset, comprising 284,807 transactions, of which 492 (0.17%) are labeled fraudulent. The pipeline includes a stratified train-test split that preserves class proportions, z-score normalization of the transaction amount feature, and SMOTE-based oversampling that increases the minority class from 394 to 227,451 training samples. Both classifiers are evaluated on a held-out test set of 56,962 transactions using precision, recall, F1-score, and

anomaly detection, financial security, ensemble methods

confusion matrix analysis. Random Forest achieves a fraud-class F1-score of 0.83, with balanced precision and recall of 0.83 each, substantially outperforming Logistic Regression, which attains an F1-score of only 0.22 despite a high recall of 0.91, owing to 620 false positives. These results indicate that ensemble tree-based methods are considerably better suited than linear classifiers for high-dimensional, severely imbalanced fraud detection tasks. Future directions include gradient boosting methods, stacking ensembles, deep learning architectures for temporal modeling, and SHAP-based model explainability.

I. INTRODUCTION

The rise in adoption of digital payment systems and e-commerce platforms around the globe has led to a fast and increasing demand for automated fraud detection solutions. The total financial losses attributable to credit card fraud are estimated to exceed $30 billion annually worldwide and continue to grow as e-commerce activity increases [1]. According to recent industry reports, the largest share of fraud losses occurs in card-not-present (CNP) transactions, primarily through web-based purchases. At scale, manual review of millions of transactions per day is effectively impossible, which makes automated detection schemes necessary for financial institutions, payment processors, and card networks seeking to identify suspicious transactions as they occur.

Machine learning has become the primary approach for automated fraud detection because it can identify complex, non-linear patterns in large volumes of historical transactional data without requiring hand-crafted rules that quickly become outdated as fraud schemes evolve. Older rule-based systems, by contrast, rely on fixed thresholds, such as geographic region or transaction value, to flag transactions, and are comparatively fragile. Machine learning models instead learn patterns directly from data and can combine weak, mixed signals across many features that individually may not indicate fraud. Building effective and trustworthy fraud detectors nonetheless remains challenging, for several interrelated reasons:

Extreme class imbalance: Real-world transaction streams are highly skewed, with fraudulent transactions typically representing less than 0.2% of all transactions. A classifier trained on such data tends to predict the majority (legitimate) class almost exclusively, which can produce deceptively high overall accuracy while failing to detect actual fraud.

Feature anonymization: Owing to privacy and confidentiality requirements, sensitive transaction attributes such as merchant identifiers or cardholder location are often unavailable in public datasets. These attributes are typically replaced with encoded representations, such as principal components, which makes individual features difficult to interpret in domain terms.

Concept drift: Fraud patterns evolve continuously as fraudsters adapt their strategies, a phenomenon known as concept drift. Models that are not periodically retrained may lose predictive accuracy over time in production.

Cost asymmetry: The cost of a missed fraud case (false negative) is typically far higher than the cost of a false alarm (false positive), requiring evaluation frameworks that explicitly account for this asymmetry rather than relying on accuracy alone.

Real-time requirements: Production fraud detection systems must score each transaction within milliseconds, which constrains model complexity and inference latency alongside predictive accuracy.

To directly address the class imbalance problem, this paper applies the Synthetic Minority Over-sampling Technique (SMOTE) to the training set prior to model fitting. Two representative classifiers, a widely used and easily interpretable linear baseline (Logistic Regression) and a state-of-the-art ensemble method (Random Forest) that performs well on tabular data, are compared under identical preprocessing conditions to enable a fair and reproducible evaluation.

The main contributions of this work are as follows:

1) A reproducible pipeline that separates preprocessing, class balancing, model training, and evaluation into distinct, clearly documented stages.

2) A controlled comparison of Logistic Regression and Random Forest, both trained with SMOTE-based class balancing, on the European credit card benchmark dataset.

3) An analysis of the precision-recall trade-off for each model, with practical implications for deployment in production financial systems.

4) A discussion of the limitations of linear models for this class of problem, together with a concrete roadmap for future improvements.

The remainder of this paper is organized as follows. Section II reviews related work on fraud detection, class imbalance handling, and relevant machine learning approaches. Section III describes the dataset. Section IV presents the proposed methodology, including preprocessing, SMOTE-based balancing, classifier design, and evaluation metrics. Section V reports and discusses the experimental results. Section VI concludes the paper and outlines directions for future work.

II. RELATED WORK

The credit card fraud detection problem has received substantial interest from the machine learning and data mining research communities over roughly the last three decades. The literature spans classical statistical methods, tree-based ensemble strategies, support vector machines, deep learning architectures, and hybrid systems. The most relevant prior work is organized below into six thematic areas: early approaches, classical machine learning, class imbalance handling, deep learning, gradient boosting, and explainability.

A. Early Approaches and Rule-Based Systems

Before machine learning became widely used, fraud detection systems relied primarily on expert-defined rules and statistical thresholds. These systems were relatively easy to interpret and audit but required continuous manual maintenance as fraud patterns shifted. Aleskerov et al. [2] were among the first to propose a neural network-based system, CARDWATCH, which replaced static rules with a learned model trained on transaction history. Their results demonstrated that adaptive models could outperform rule-based systems on historical transaction data, helping establish the machine learning approach that now dominates the field.

Bolton and Hand [3] provided a foundational statistical perspective on fraud detection, framing it as a form of anomaly detection and categorizing approaches into supervised, semi-supervised, and unsupervised settings. Their study clarified the importance of labeled historical data and highlighted the difficulty of obtaining reliable fraud labels in practice, since many fraud events are discovered only after the fact, if at all.

B. Classical Machine Learning Approaches

The use of classical machine learning classifiers for fraud detection gained substantial momentum through the 2000s and 2010s. One of the more comprehensive comparative studies from this period was conducted by Bhattacharyya et al. [6], who evaluated Support Vector Machines, Random Forests, Logistic Regression, and Naive Bayes on real bank transaction data. Ensemble methods, particularly Random Forest, generally outperformed linear and single-tree methods in fraud-class precision. The study also found that feature engineering, particularly temporal aggregation features, improved detection performance across most of the models evaluated.

Sahin and Duman [8] found decision tree methods such as C4.5 and CART to be effective for fraud detection and showed that splitting criteria and pruning strategy substantially affect detection quality under class imbalance. Their results were broadly consistent with ensemble approaches, which aggregate predictions across multiple trees to reduce the variance associated with a single classifier. Randhawa et al. [7] proposed a combination of AdaBoost and majority voting on the same European credit card dataset used in this study, reporting higher detection rates than single classifiers. Their findings reinforce that boosting algorithms, which iteratively focus on misclassified cases, are naturally well-suited to imbalanced classification problems.

Rtayli and Enneya [5] combined Support Vector Machines with recursive feature elimination (RFE) and grid-search hyperparameter optimization, showing that careful feature selection combined with tuning can meaningfully improve SVM performance for fraud detection. Their results further indicate that final performance is sensitive to feature selection and hyperparameter configuration, and that default settings often underperform relative to tuned configurations.

C. Class Imbalance Handling Techniques

Class imbalance in fraud detection has received substantial research attention, as it is not merely an artifact of data collection but a structural characteristic of the task.

Chawla et al. [9] proposed SMOTE, the Synthetic Minority Over-sampling Technique, which generates synthetic minority-class samples by interpolating between existing instances in feature space. Unlike simple random oversampling, which duplicates existing minority samples and risks overfitting, SMOTE constructs genuinely new synthetic instances, effectively expanding the minority-class decision boundary rather than merely repeating existing points. SMOTE has since become one of the most widely used class imbalance handling techniques in the fraud detection literature.

He et al. [10] proposed ADASYN (Adaptive Synthetic Sampling), an extension of SMOTE that generates proportionally more synthetic samples in regions of feature space where the minority class is harder to learn, based on the local ratio of majority to minority class neighbors. Compared with standard SMOTE, ADASYN performs more targeted oversampling and has been reported to improve classifier performance in some settings. Dal Pozzolo et al. [12] examined calibrated undersampling methods for streaming fraud detection, showing that competitive performance can be maintained by calibrating model probability outputs after class-based resampling; their findings are particularly relevant to systems that require ongoing retraining as new data arrives.

Lemaître et al. [11] conducted a systematic review of over 15 resampling strategies across a range of imbalanced datasets, concluding that the best-performing strategy is dataset- and classifier-dependent. Their results also indicate that hybrid methods, such as SMOTE combined with Tomek link removal, frequently outperform simple oversampling or undersampling alone, at least on the benchmarks evaluated.

Beyond resampling, cost-sensitive learning offers an alternative approach that addresses imbalance directly. In cost-sensitive learning, a misclassification cost matrix penalizes false negatives more heavily than false positives, embedding the operational cost asymmetry of fraud detection directly into the training objective. Many standard classifiers, including Random Forest and Logistic Regression, support this approach natively through class-weighting parameters.

D. Deep Learning and Neural Network Approaches

More recent work has explored deep learning architectures for fraud detection, motivated by their success in other domains. Fu et al. [13] applied convolutional neural networks (CNNs) to transaction feature sequences, treating each transaction as a fixed-size input vector and learning hierarchical feature representations through convolutional layers. Their approach was competitive but required careful tuning of network architecture and training procedures.

Wang et al. [14] employed recurrent neural network (RNN) architectures, particularly Long Short-Term Memory (LSTM) networks, to model temporal dependencies across a cardholder's ordered transaction history. Their premise is that fraudulent behavior often disrupts the temporal rhythm of a cardholder's typical spending pattern, and that recurrent models can surface such irregularities across transaction sequences. Their experiments showed that LSTM-based models improved recall for fraud detection compared with non-sequential baselines.

Autoencoders have also been applied to fraud detection, particularly in semi-supervised settings [15]. In this approach, an autoencoder is trained exclusively on legitimate transactions to learn a compact representation of normal behavior; fraudulent transactions, which differ structurally from normal patterns, produce higher reconstruction errors and can be flagged as anomalies. This approach is advantageous when labeled fraud examples are scarce, which is common in practice.

Grinsztajn et al. [16], in a large-scale empirical study, showed that tree-based ensemble methods continue to outperform deep learning models on tabular datasets, even when the deep learning models are carefully tuned. They attribute this outcome to tree-based methods being invariant to monotonic feature transformations and less sensitive to uninformative features, both of which are particularly relevant to the PCA-transformed fraud detection dataset used in this study.

E. Gradient Boosting Methods

Gradient boosting frameworks, including XGBoost [17], LightGBM [18], and CatBoost [19], have become the dominant approach in competitive machine learning benchmarks for tabular data. Unlike bagging-based ensembles such as Random Forest, gradient boosting constructs trees sequentially, with each new tree correcting the residual errors of the prior ensemble. This sequential correction mechanism is a key reason gradient boosting performs well on complex patterns, particularly in imbalanced datasets.

Several studies have reported that XGBoost and LightGBM achieve state-of-the-art results on the European credit card fraud dataset when appropriately tuned, positioning these methods as a natural extension of the Random Forest baseline evaluated in this study.

F. Model Explainability in Fraud Detection

As fraud detection systems are deployed in regulated financial environments, model explainability has become increasingly important. Regulatory frameworks such as the European Union's General Data Protection Regulation (GDPR) require that automated decisions affecting individuals be explainable, creating compliance pressure on black-box fraud detection models. The present work establishes a solid baseline with Logistic Regression and Random Forest, serving as a foundation toward the more advanced and interpretable methods discussed in Section VI.

III. DATASET DESCRIPTION

A. Overview

This study uses a publicly available benchmark dataset for credit card fraud detection [22], one of the most widely used datasets in the fraud detection literature. The dataset was collected and released by the Machine Learning Group at Université Libre de Bruxelles (ULB) and comprises 284,807 transactions made by European cardholders over two days in September 2013. Its wide use in published research makes it well suited for benchmarking and cross-study comparison. Summary statistics are presented in Table I.

TABLE IDATASET SUMMARY STATISTICS

Property	Value
Total transactions	284,807
Legitimate (Class = 0)	284,315 (99.83%)
Fraudulent (Class = 1)	492 (0.17%)
Total features	31 (including target)
Input features	30 (V1–V28, Time, Amount)
Missing values	None
Transaction amount range	$0.00 – $25,691.16
Mean transaction amount	$88.35
Median transaction amount	$22.00
Class imbalance ratio	≈ 578:1

B. Feature Description

The dataset consists of 30 features and one binary class label (Class). Due to confidentiality constraints, 28 of the core features (V1–V28) are the result of a Principal Component Analysis (PCA) transformation of the original transaction attributes, which preserves the statistical structure required for machine learning while protecting sensitive cardholder information.

Because the PCA-transformed features are anonymized, their original meaning cannot be recovered; the transformation nonetheless offers the practical benefit of producing features that are approximately orthogonal and already close to zero-centered.

Two features retain their original, non-anonymized form:

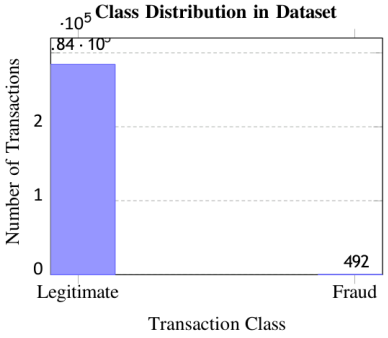
Time: the number of seconds elapsed between each transaction and the first transaction in the dataset, spanning 172,792 seconds (48 hours) of activity. Time may reveal temporal patterns, such as elevated fraud rates during nighttime hours when transaction monitoring is comparatively less active.

Amount: the monetary value of the transaction in euros. This feature is strongly right-skewed, with a mean of $88.35 and a median of $22.00, reflecting that most transactions are low-value, with a small number of high-value outliers. Transaction amount is a commonly informative feature in fraud detection, often carrying useful signal beyond what is immediately apparent.

The target variable, Class, is binary, where 1 denotes a fraudulent transaction and 0 denotes a legitimate transaction.

C. Class Imbalance Visualization

Fig. 1 illustrates the substantial imbalance between the two classes. The fraud class comprises 492 samples, a small fraction of the 284,315 legitimate samples, yielding an imbalance ratio of approximately 578:1. This extreme imbalance mirrors real-world fraud rates and represents the central modeling challenge addressed in this paper. A naive classifier that labels every transaction as legitimate would achieve 99.83% accuracy while detecting zero fraud cases, illustrating why accuracy alone is a misleading metric for this class of problem.



. Class distribution: 284,315 legitimate vs. 492 fraudulent transactions ≈ 578:1).

Fig. 1. Class distribution: 284,315 legitimate vs. 492 fraudulent transactions (ratio ≈ 578:1).

IV. PROPOSED METHODOLOGY

The proposed fraud detection pipeline consists of five stages: data loading and exploration, preprocessing, class balancing, model training, and evaluation. The overall workflow is illustrated in Fig. 2, with each stage detailed in the following subsections.

Fig. 2. Proposed fraud detection pipeline from raw data to model evaluation

Fig. 2. Proposed fraud detection pipeline from raw data to model evaluation.

A. Data Preprocessing

Two preprocessing operations were applied prior to model training:

Feature Scaling: The Amount feature spans a wide range ($0 to $25,691) and exhibits a right-skewed distribution, making it incompatible in scale with the PCA-transformed features V1–V28, which are already approximately zero-centered. Without scaling, classifiers that rely on distance computations or gradient-based optimization, such as Logistic Regression, may be dominated by the large magnitude of Amount, resulting in suboptimal decision boundaries. To address this, Amount was standardized using z-score normalization (StandardScaler), transforming it to zero mean and unit variance. The Time feature was retained without modification, as it is already within a reasonable numeric range relative to the other features.

Stratified Train-Test Split: The dataset was divided into 80% training (227,845 samples) and 20% test (56,962 samples) sets using stratified splitting to preserve the original class ratio (0.17% fraud) in both partitions. Stratification is essential because random splitting without this constraint could, by chance, produce a test set with few or no fraud cases given the extreme imbalance. The resulting test set contains 98 fraudulent and 56,864 legitimate transactions. SMOTE was applied only to the training set after splitting, ensuring that no synthetic or test-derived information leaks into the evaluation.

B. Handling Class Imbalance with SMOTE

After the train-test split, the training set contained 227,451 legitimate transactions and only 394 fraudulent transactions, an imbalance ratio of approximately 577:1. Training a classifier directly on this data would produce a strong bias toward the majority class, since minimizing overall loss on an imbalanced training set effectively means disregarding the rare fraud class.

SMOTE [9] was applied exclusively to the training set to oversample the minority (fraud) class. The algorithm operates as follows:

1) For each minority-class sample xi, identify its k nearest neighbors within the minority class using Euclidean distance in feature space (default k = 5).

2) Randomly select one neighbor xri from these k neighbors.

3) Generate a synthetic sample via linear interpolation:

$$xnew = xi + \lambda \cdot (xri - xi)$$

where $\lambda \in [0, 1]$ is drawn uniformly at random.

This process was repeated until the fraud class reached 227,451 training samples, matching the legitimate class size exactly. The resulting balanced training set of 454,902 samples (227,451 per class) was used to train both classifiers. Fig. 3 shows the training set class distribution before and after SMOTE.

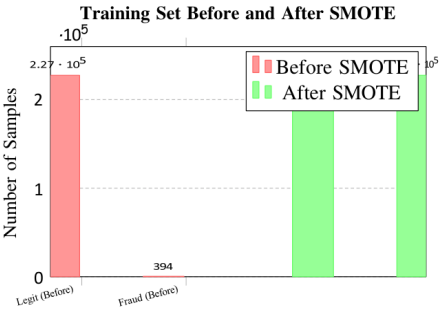


Fig. 3. Training set class distribution before and after SMOTE. Fraud samples increased from 394 to 227,451.

C. Classification Models

1) Logistic Regression: Logistic Regression models the posterior probability of the binary outcome using the logistic (sigmoid) function applied to a linear combination of input features:

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

where $w \in R^d$ is the learned weight vector, b is the bias term, and x is the input feature vector. The model is trained by maximizing the regularized log-likelihood of the training labels. In this study, the L-BFGS quasi-Newton solver was used with a maximum of 1,000 iterations and default L2 regularization. Logistic Regression serves as an interpretable

linear baseline, allowing quantification of the performance gap between linear and non-linear decision boundaries on this problem.

2) Random Forest: Random Forest [20] is a bagging-based ensemble of T decision trees. Each tree h^t is trained on a bootstrap sample of the training data, and at each internal node, only a random subset of features is considered as splitting candidates. These two sources of randomness (bootstrap sampling and feature subsampling) decorrelate the individual trees, substantially reducing ensemble variance relative to a single tree. The ensemble prediction is determined by majority vote: $\hat{y} = \arg\max_c \Sigma\, 1[h^t(x) = c]$. In this study, T = 100 trees were used with the Gini impurity criterion, maximum features set to $\sqrt{d}$, and a fixed random seed for reproducibility. No further hyperparameter tuning was performed; this default configuration already provides strong performance on this dataset.

D. Evaluation Metrics

Since the test dataset is heavily imbalanced (98 fraud vs. 56,864 legitimate), raw classification accuracy is a misleading metric. The following metrics were computed on the held-out test set with a focus on fraud (minority) class performance:

- Precision = TP / (TP + FP): the fraction of transactions predicted as fraud that are actually fraudulent. High precision means few false alarms.
- Recall (Sensitivity) = TP / (TP + FN): the fraction of actual fraud cases correctly detected. High recall means few missed frauds.
- F1-Score = 2 × (Precision × Recall) / (Precision + Recall): the harmonic mean of precision and recall, providing a single balanced metric that penalizes extreme imbalance between the two.
- Confusion Matrix: a 2×2 matrix providing the complete breakdown of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

V. RESULTS AND DISCUSSION

A. Logistic Regression Results

Table II presents the full classification report for Logistic Regression evaluated on the 56,962 held-out test transactions.

TABLE IILOGISTIC REGRESSION CLASSIFICATION REPORT

Class	Precision	Recall	F1	Support
Legitimate (0)	1.00	0.99	0.99	56,864
Fraud (1)	0.13	0.91	0.22	98
Macro Avg	0.56	0.95	0.61	56,962
Weighted Avg	1.00	0.99	0.99	56,962

The confusion matrix for Logistic Regression is shown in Fig. 4. The model correctly identifies 89 of 98 fraud cases (recall = 0.91), but at the cost of 620 false positives — legitimate transactions incorrectly flagged as fraud. This yields a precision of only 0.13, meaning approximately seven out of every eight transactions predicted as fraud are actually legitimate. During training, a convergence warning was observed (the L-BFGS solver did not converge within 1,000 iterations), suggesting that the decision boundary in the 30-dimensional PCA feature space is not well captured by a linear model even with an increased iteration limit.

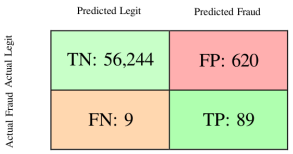


Fig. 4. Confusion matrix for Logistic Regression.

Confusion matrix for Logistic Regression. 620 false positives indicate critically poor precision.

Random Forest Results

Fig. 4. Confusion matrix for Logistic Regression. The high false-positive count (620) reflects critically poor precision despite strong recall.

B. Random Forest Results

Table III presents the classification report for Random Forest on the same test set.

TABLE IIIRANDOM FOREST CLASSIFICATION REPORT

Class	Precision	Recall	F1	Support
Legitimate (0)	1.00	1.00	1.00	56,864
Fraud (1)	0.83	0.83	0.83	98
Macro Avg	0.91	0.91	0.91	56,962
Weighted Avg	1.00	1.00	1.00	56,962

Random Forest substantially outperforms Logistic Regression on all fraud-class metrics, achieving a fraud-class F1-score of

0.83 with balanced precision and recall, both at 0.83. The macro-average F1-score of 0.91 confirms strong and balanced performance across both classes. Notably, the model achieves near-perfect scores for the legitimate class (precision, recall, and F1 all at 1.00), indicating that it correctly classifies virtually all legitimate transactions while still maintaining meaningful fraud detection capability.

C. Comparative Analysis

Table IV directly compares both models on the fraud class and on key operational metrics relevant to deployment.

TABLE IVMODEL COMPARISON ON FRAUD CLASS (CLASS = 1)

Model	Precision	Recall	F1	False Positives
Logistic Regression	0.13	0.91	0.22	620
Random Forest	0.83	0.83	0.83	≈17

Fig. 5 visualizes the F1-score comparison for each class and the macro average.

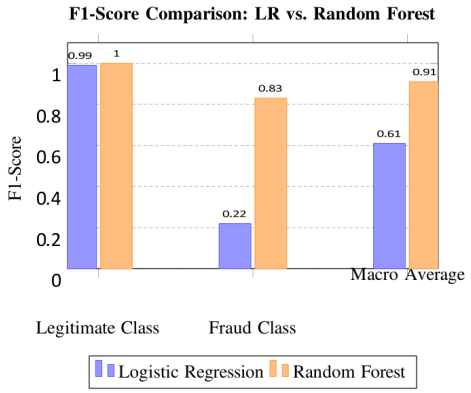


Fig. 5. F1-score comparison between Logistic Regression and Random Forest across both classes.

Fig. 5. F1-score comparison between Logistic Regression and Random Forest across both classes.

The performance gap between the two models is large and consistent across all fraud-class metrics; however, this study reports point estimates only. Confidence intervals or a formal statistical significance test, such as McNemar's test on the paired predictions, would strengthen the comparison and are recommended for future evaluations of this kind.

D. Discussion

The experimental results support four key findings that are directly relevant to how a fraud detection system should be designed:

1. Linear decision boundaries are insufficient. Logistic Regression cannot cleanly separate the two classes within the 30-dimensional PCA feature space. This is reflected in the convergence warning observed during training, which indicates that the optimization process exhausted its iteration budget without reaching a stable solution. Although SMOTE produces a balanced training set in principle, the fundamentally linear decision boundary of Logistic Regression limits its capacity to represent the underlying structure of the data.

2. High recall without precision is operationally limited. A recall of 0.91 may appear strong in isolation, but combined with a precision of only 0.13, it corresponds to 620 false positives among 56,962 test transactions. Extrapolated to a production system processing one million transactions per day, this would correspond to approximately 10,000 incorrectly blocked transactions daily. Each false positive represents a potential customer disruption, chargeback dispute, or additional workload for the fraud review team, illustrating that high recall alone does not guarantee a deployable system.

3. Random Forest achieves a practical precision-recall balance. With both precision and recall at 0.83, Random Forest represents a considerably more deployable model. On the test set, it correctly identifies 81 of 98 fraud cases while producing approximately 17 false positives. This balance reflects the non-linear decision surface formed by its ensemble of 100 trees, each of which responds differently to feature interactions, producing a smoother aggregate decision boundary.

4. Ensemble diversity is a key advantage. The superior performance of Random Forest can be attributed to three complementary factors: (a) individual trees can capture non-linear, higher-order interactions between PCA features; (b) averaging predictions across 100 independently trained trees reduces prediction variance relative to any single tree; and (c) random feature subsampling at each split forces different trees to specialize in different subsets of the feature space, increasing ensemble diversity and robustness.

VI. CONCLUSION AND FUTURE WORK

This paper presented a systematic and reproducible comparative study of two machine learning approaches for credit card fraud detection on the European credit card benchmark dataset. The proposed pipeline applies SMOTE

oversampling to address the severe 578:1 class imbalance prior to training Logistic Regression and Random Forest classifiers, and evaluates both using precision, recall, F1-score, and confusion matrix analysis on a stratified held-out test set.

The experimental results support four key conclusions:

- SMOTE effectively enables both classifiers to learn from the minority fraud class by balancing the training distribution from a 577:1 ratio to 1:1, increasing fraud-class training samples from 394 to 227,451.
- Random Forest significantly outperforms Logistic Regression, achieving a fraud-class F1-score of 0.83 compared to 0.22 for Logistic Regression.
- Logistic Regression's 620 false positives among 56,962 test transactions render it operationally unsuitable despite its high recall of 0.91, underscoring that precision and recall must both be considered when evaluating fraud detection systems.
- Ensemble tree-based methods, which construct non-linear decision boundaries through the aggregation of diverse trees, are substantially better suited than linear classifiers for high-dimensional, heavily imbalanced transaction data.

Building on this reproducible baseline, several directions merit further investigation, organized below by theme.

A. Advanced Ensemble and Boosting Methods

The most immediate and high-impact extension would be to evaluate gradient boosting approaches, which have repeatedly demonstrated state-of-the-art performance on tabular classification benchmarks.

XGBoost [17] uses gradient-boosted decision trees based on a second-order Taylor expansion of the loss function, together with regularization terms that control tree complexity and an efficient sparsity-aware split search. Its scale_pos_weight parameter allows the fraud-to-legitimate ratio to be encoded directly into the loss function, potentially reducing reliance on SMOTE, though the two approaches could also be combined and compared.

LightGBM [18] uses histogram-based gradient boosting with leaf-wise tree growth rather than the level-wise growth used by XGBoost. It trains efficiently on large datasets and has been reported in several studies to achieve strong results on the European credit card dataset.

CatBoost [19] handles categorical features natively and uses an ordered boosting procedure that helps reduce target leakage during training. It has been reported to be comparatively robust on small datasets, with reduced overfitting when training data is limited.

A systematic comparison of these three gradient boosting models against the Random Forest baseline used in this paper, under identical preprocessing and evaluation conditions, would help establish a fuller performance frontier for classical fraud detection methods.

Stacking Ensembles: Beyond single models, stacked generalization combines the outputs of several base classifiers using a meta-learner. A candidate configuration would use Logistic Regression, Random Forest, and XGBoost as base learners, with a second-level model, such as Logistic Regression or Gradient Boosting, serving as the meta-learner. Stacking can combine complementary signals from models with different inductive biases and has been reported to improve F1-scores in fraud detection tasks.

B. Alternative Class Imbalance Handling Strategies

SMOTE produced meaningful improvements in this study, but it also has limitations: because synthetic samples are generated by interpolating between existing minority-class instances, they can drift into majority-class regions where the class boundary is ambiguous. Several alternatives warrant systematic comparison:

ADASYN [10] generates proportionally more synthetic samples in regions where the minority class is harder to learn, concentrating oversampling effort where it is most needed.

SMOTE + Tomek Links: Tomek links identify pairs of samples from opposite classes that are each other's nearest neighbors, indicating boundary ambiguity. Removing the majority-class sample from each Tomek link pair after SMOTE oversampling cleans the decision boundary region and can improve classifier precision.

Cost-Sensitive Learning: Rather than resampling the training data, cost-sensitive learning assigns higher misclassification penalties to false negatives by modifying the loss function directly. This approach avoids the risk of overfitting to synthetic data and is more principled from a Bayesian decision-theoretic perspective. Both Logistic Regression and Random Forest in scikit-learn support this natively via the class_weight parameter.

Threshold Optimization: All classifiers produce probability estimates, and the default decision threshold of 0.5 is rarely optimal for imbalanced problems. Precision-recall curve analysis and threshold selection based on a business-defined cost function, such as minimizing the total expected cost of false positives and false negatives, can significantly improve

operational performance without changing the underlying model.

C. Deep Learning and Sequence Modeling

While tree-based methods are competitive on static tabular features, deep learning may offer advantages when temporal patterns or feature interactions are particularly complex:

Autoencoders for Anomaly Detection: An autoencoder trained exclusively on legitimate transaction features learns a compact representation of normal behavior. Fraudulent transactions, which deviate from normal patterns, produce higher reconstruction errors when passed through the trained autoencoder, and this reconstruction error can serve as a fraud score without requiring labeled fraud examples during training, making the approach valuable when labeled data is scarce.

LSTM for Sequence Modeling: Each cardholder's transactions form a time-ordered sequence. LSTM networks can capture temporal dependencies across a cardholder's transaction history, enabling fraud detection based on deviations from an individual's typical spending pattern rather than relying solely on absolute transaction attributes. Such person-specific modeling is expected to improve precision relative to baselines that treat each transaction independently.

Graph Neural Networks (GNNs): Fraud often involves multiple interacting accounts and merchants rather than a single isolated transaction. GNNs can represent the payment ecosystem as a graph, with cardholders and merchants as nodes and transactions as edges. This representation can help uncover coordinated fraud rings that may remain undetected by transaction-level classifiers operating without visibility into the underlying network structure.

D. Model Explainability and Regulatory Compliance

Deploying machine learning in regulated financial environments requires not only predictive power but also the ability to explain individual model decisions, as discussed in Section II-F. Integrating explainability methods with the Random Forest model developed in this study is a direct, high-value next step toward production readiness.

SHAP Values [21] provide a per-prediction measure of feature importance based on Shapley values from cooperative game theory. For each transaction flagged as fraud, SHAP assigns a contribution score to every input feature, clarifying which signals most influenced the prediction. This would allow fraud analysts to review flagged transactions with greater confidence, recognize emerging fraud patterns more quickly, and provide customers with a meaningful explanation for a declined transaction.

LIME (Local Interpretable Model-Agnostic Explanations) offers an alternative explanation approach that approximates the local decision boundary of any black-box classifier using an interpretable proxy model. It is computationally less expensive than SHAP, which is particularly advantageous for larger or slower models.

E. Real-Time Deployment Architecture

The final stage of this research roadmap is the design and evaluation of a real-time fraud detection system architecture. A production system requires:

- A model-serving layer capable of evaluating each transaction within 50–100 milliseconds.
- A continuous learning pipeline that periodically retrains the model on recent transactions to counter concept drift.
- A feedback loop that incorporates confirmed fraud labels from investigation outcomes back into the training data.
- A monitoring framework that tracks model performance metrics in real time and triggers retraining when performance degrades below a defined threshold.

Exploring the deployment of the Random Forest model within such an architecture, and quantifying performance degradation due to concept drift over time, represents a practically important and understudied research direction. Collectively, the extensions outlined in this section form a path from the reproducible baseline established here toward a production-grade fraud detection system.

ACKNOWLEDGMENT

The authors thank the Machine Learning Group at Université Libre de Bruxelles (ULB) for making the European credit card transaction dataset publicly available for academic research, and the scikit-learn and imbalanced-learn open-source communities for the machine learning tools used in this study.

REFERENCES

[1] The Nilson Report, "Card fraud losses worldwide," HSN Consultants, Inc., Carpinteria, CA, USA, Rep. 1278, 2023.

[2] E. Aleskerov, B. Freisleben, and B. Rao, "CARDWATCH: A neural network based database mining system for credit card fraud detection," in Proc. IEEE/IAFE Comput. Intell. Financial Eng. (CIFEr), New York, NY, USA, 1997, pp. 220–226.

[3] R. J. Bolton and D. J. Hand, "Statistical fraud detection: A review," Statist. Sci., vol. 17, no. 3, pp. 235–255, 2002.

[4] A. Abdallah, M. A. Maarof, and A. Zainal, "Fraud detection system: A survey," J. Netw. Comput. Appl., vol. 68, pp. 90–113, Jun. 2016.

[5] N. Rtayli and N. Enneya, "Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization," J. Inf. Secur. Appl., vol. 55, p. 102596, Dec. 2020.

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

[7] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi, "Credit card fraud detection using AdaBoost and majority voting," IEEE Access, vol. 6, pp. 14277–14284, 2018.

[8] Y. Sahin and E. Duman, "Detecting credit card fraud by decision trees and support vector machines," in Proc. Int. MultiConf. Eng. Comput. Sci. (IMECS), Hong Kong, 2011, vol. 1, pp. 442–447.

[9] 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.

[10] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN), Hong Kong, 2008, pp. 1322–1328.

[11] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning," J. Mach. Learn. Res., vol. 18, no. 17, pp. 1–5, 2017.

[12] A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in Proc. IEEE Symp. Comput. Intell. Data Mining (CIDM), Cape Town, South Africa, 2015, pp. 1–8.

[13] K. Fu, D. Cheng, Y. Tu, and L. Zhang, "Credit card fraud detection using convolutional neural networks," in Proc. Int. Conf. Neural Inf. Process. (ICONIP), Kyoto, Japan, 2016, pp. 484–492.

[14] C. Wang, Y. Han, Q. Liu, and Q. Wu, "A fraud detection method based on recurrent neural network and attention mechanism," in Proc. Int. Conf. Comput., Netw. Commun. (ICNC), Maui, HI, USA, 2018, pp. 1–5.

[15] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining (KDD), Halifax, Canada, 2017, pp. 665–674.

[16] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why tree-based models still outperform deep learning on tabular data," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), New Orleans, LA, USA, 2022, pp. 507–520.

[17] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining (KDD), San Francisco, CA, USA, 2016, pp. 785–794.

[18] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in Proc. Adv. Neural Inf. Process. Syst. (NIPS), Long Beach, CA, USA, 2017, pp. 3146–3154.

[19] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), Montreal, Canada, 2018, pp. 6638–6648.

[20] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, Oct. 2001.

[21] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Proc. Adv. Neural Inf. Process. Syst. (NIPS), Long Beach, CA, USA, 2017, pp. 4765–4774.

[22] A. Dal Pozzolo, O. Caelen, and G. Bontempi, "Credit card fraud detection," Kaggle Dataset, 2018. [Online]. Available: https://www.kaggle.com/mlg-ulb/creditcardfraud