University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR)

ISSN: 2714-3627

A Journal of the Department of Computer Science, University of Ibadan, Ibadan, Nigeria

Volume 14 No. 1, June, 2025

journals.ui.edu.ng/uijslictr
http://uijslictr.org.ng/

University of Ibadan
Journal of Science and Logics in ICT Research

Comparative Analysis of Machine Learning Models for Fraud Detection in Imbalanced Credit Card Transaction Datasets

[1]✉Enehikhare C. C. and [2]Odumuyiwa V.

[1,2]Department of Computer Sciences, University of Lagos, Akoka, Nigeria
[1]christianah.eke@gmail.com [2]vodumuyiwa@unilag.edu.ng

Abstract
Fraud detection in imbalanced datasets presents a major challenge in financial domains, particularly in credit card fraud detection. This paper presents a comparative analysis of popular machine learning models—Logistic Regression, Decision Trees, Random Forest, and XGBoost applied to real and simulated fraud datasets. Various data preprocessing techniques, such as SMOTE, Random Sampling, were employed to address class imbalance. The results indicate that ensemble models, particularly Random Forest and XGBoost, outperformed traditional models, achieving near-perfect F1-scores (0.999) and accuracies (0.999) across both datasets. These findings provide insight into model effectiveness in fraud detection tasks and offer a foundation for developing robust, adaptive fraud detection systems.

Keywords: *Credit card fraud, machine learning, imbalanced datasets, SMOTE, Random Forest, XGBoost, Random Sampling.*

1. Introduction

Credit card fraud is one of the common cybercrime practices [1] and it poses a significant threat to financial institutions, retailers, and consumers alike, resulting in billions of dollars in losses annually. In addition to immediate losses, the brand name may suffer from a decline in customer trust brought on by the deception. Because of these increased losses, financial institutions and card issuers are in constant search for new methods and innovations in the detection and prevention of payment card fraud.

Traditional rule-based fraud detection systems have become inadequate in identifying emerging fraud patterns, especially in datasets characterized by extreme class imbalance, where fraudulent transactions are heavily outnumbered by legitimate ones. Given that credit card fraud directly affects financial institutions in losses, clients confidence, and regulatory compliance, detecting such fraud is a very essential function for financial institutions to perform. Machine learning algorithms offer a promising solution by learning intricate patterns from past transactions to detect fraud. Unfortunately, machine learning models have a lot of difficulty because of the intrinsic imbalance in credit card transaction datasets, where fraudulent transactions are a small minority compared to valid purchases. Dealing with imbalanced datasets introduces additional complexity, as most models tend to favor the majority class, leading to high false negatives. Detection of fraudulent transactions is a minority class, which is difficult to identify using traditional machine learning algorithms, since they are biased toward the majority class. This can lead to misclassification of fraudulent transactions, which is an unwanted condition for creating high false negatives due to class imbalance. This may cause significant financial losses as well as damage to the integrity of the fraud detection system.

The inverse relation between recall and precision adds still another level of complexity to the issue. High precision would ensure the rate of false positives is low enough to avoid annoying actual consumers and adding unnecessary costs to operations. On the other hand, the false negative measure should be kept low by ensuring a high recall rate to minimize fraud passing through undetected. This research seeks to tackle the trade-off between precision and recall and detect fraudulent credit card transactions with high

Enehikhare C. C. and Odumuyiwa V. (2025). Comparative Analysis of Machine Learning Models for Fraud Detection in Imbalanced Credit Card Transaction Datasets. *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR),* Vol. 14 No. 1, pp. 25 - 37

efficiency in very imbalanced credit card datasets through a comprehensive analysis of different machine learning models, data preprocessing techniques, and ensemble methods,

In this study, the effectiveness of various machine learning models (Logistic Regression, Decision Trees, Random Forest, and XGBoost) in detecting credit card fraud is explored. We also assess how synthetic oversampling techniques like SMOTE and Random sampling can help mitigate class imbalance, enhancing the models' ability to detect fraudulent transactions.

2. Literature Review

Financial fraud is said to occur when fraudulent activities such as unauthorized transactions, identity theft, or money laundering are performed within financial systems [2]. Institutions employ fraud detection technologies to scan historical datasets of transactions and detect suspicious activities. The datasets typically contain transaction records characterized by diverse attributes such as the size of a transaction, the time it was initiated, the sender and receiver information, and the type of transaction (a fraudulent transaction or a non-fraudulent transaction) [3].

A dataset is said to be balanced when the number of examples in each class (i.e., fraudulent and non-fraudulent transactions) is nearly equal. In fraud detection, it implies that the dataset has nearly an equivalent proportion of fraudulent and legitimate transactions. Balanced datasets are more appropriate for conventional machine learning algorithms since they enable balanced training without any bias towards a particular class. In most real-world financial fraud detection problems, the dataset is imbalanced in the sense that one class (non - fraudulent transactions) overwhelmingly dominates the other class (fraudulent transactions). Under normal circumstances, fraudulent transactions form a minute proportion of total transactions, even as low as 1% in some cases. This poses an issue to machine learning algorithms because when trained with such data, they will be biased towards the majority class, and therefore their fraud detection rate is compromised [4].

Several machine-learning techniques have been explored in the context of fraud detection. Traditional models like Logistic Regression and Decision Trees have been effective in detecting simple fraud patterns, but they often struggle with nonlinear data relationships [5].

Cheng & Xiong [6] suggested a novel technique to optimize the support vector machine based on cuckoo search algorithm in order to improve its credit card fraud detection. Cuckoo search algorithm improves the classification performance through the optimization of support vector machine kernel function parameters (C, g). Results show that CS-SVM is better than SVM in Accuracy, Precision, Recall, F1-score, AUC, and better than Logistic Regression, Random Forest, Decision Tree and Naive Bayes.

Ali et al. [7] presented a systematic literature review (SLR) of existing literature regarding machine learning (ML)-based fraud detection. The review made use of the Kitchenham method, which employs well-defined procedures to extract and synthesize the relevant articles and then report the results obtained. Based on the specified search strategies from prominent electronic database libraries, different studies were considered. Upon inclusion/exclusion criteria, 93 articles were chosen, synthesized and analyzed. The review presents an overview of commonly used ML techniques used to detect fraud, the most commonly utilized fraud type, and evaluation metrics. The reviewed articles showed that SVM and ANN are common ML algorithms used to identify fraud. The review also showed that credit card fraud is the most commonly fraud type solved by using ML techniques. The article then ultimately generates primary issues, areas of study missing, and inadequacies in financial fraud detection areas and lists possible domains of future research.

Vanini et al. [8] developed three models: machine learning-based fraud detection, economic optimization of machine learning output, and a risk model for estimating the fraud risk with the inclusion of countermeasures. The models were tested against real data. According to the authors, their machine learning model reduces the expected and unexpected losses in the three combined payment channels by 15% compared to a benchmark constructed from static if-then rules. Improving the machine-learning model reduces the expected losses by an additional 52%. The outcomes are sustained at a low false positive rate of 0.4%. The three models' risk framework is hence viable from both a business and risk perspective. The framework is, however, limited in the sense that it operates in a

single direction, i.e., from machine learning methods in fraud detection to statistical risk modelling. The feedback process from the risk model to the triage model and back from the triage model to the fraud detection model is a challenging problem that can be solved using reinforcement-learning methods. With this kind of feedback loop, the whole risk-management system becomes a learning system.

The strengths and weakness of the existing works are presented in Table1. While current studies have investigated a variety of machine learning techniques for fraud detection like traditional models, optimization techniques like CS-SVM, and new frameworks based on risk models, there are still some research gaps. One of the significant gaps is the lack of comparative analysis among more than one model under various data scenarios, especially class imbalance, which is present in fraud detection. Most works focus on one dataset or model alone without robustly testing performance across different balancing methods and model forms. Second, although SMOTE and sampling techniques are typically considered, limited works consider their effect across different datasets with identical test metrics.

Table1: Summary of Strengths and Weaknesses of Recent Studies on Machine Learning in Fraud Detection

Paper	**Strengths**	**Weaknesses**
Sakharova [5]	Explores traditional models like Logistic Regression and Decision Trees for fraud detection.	Struggles with nonlinear data relationships, making it less effective for complex fraud patterns.
Cheng & Xiong [6]	Introduces a novel technique to optimize SVM using the cuckoo search algorithm, improving fraud detection performance	Focuses mainly on credit card fraud, limiting the generalizability to other fraud types
Ali et al. [7]	Systematic Literature Review (SLR) synthesizing 93 articles on ML-based fraud detection, providing an overview of commonly used techniques (SVM, ANN) and fraud types (credit card fraud).	Does not explore in-depth limitations of applying the identified techniques in real-world scenarios.
Vanini et al. [8]	Developed a machine learning-based fraud detection model that reduces expected losses and maintains a low false-positive rate. The model is tested against real data, providing practical business applications.	Risk framework operates in a one-directional manner; feedback loops between fraud detection and risk models are not explored.
Sandhya et al. [9]	Developed an innovative approach to fraud detection in transaction data streams, extracting behavioral patterns from historical customer transaction data.	Focuses primarily on transaction data streams; lacks a broader exploration of other fraud detection approaches.
Btoush et al. [10]	Reviewed machine learning/deep learning techniques for credit card cyber fraud detection, offering insights into the suitability of these methods.	Focuses on credit card cyber fraud, potentially limiting the scope of fraud detection methods to only this area

3. Research Methodology

In this study, the effectiveness of four machine learning models—Logistic Regression, Decision Tree, Random Forest and XGBoost—in detecting credit card fraud was compared. Figure1 presents a flowchart of our methodology. Two datasets were used in the experiments in the three scenarios considered: first, where both datasets were imbalanced; second, after both datasets were balanced using SMOTE technique; and third, after both datasets were balanced using Random Sampling technique. The metrics used for evaluation were Recall, Precision, F1-score and Accuracy.

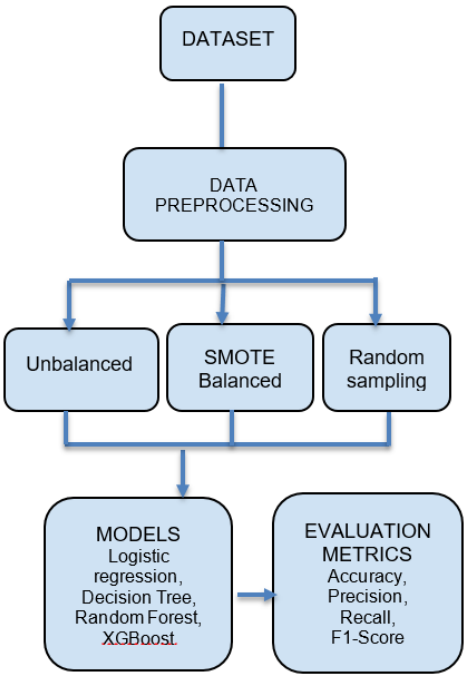

Figure1: Flowchart of the Credit Card Fraud Detection Methodology

3.1 Datasets

Two datasets were used in this study. Dataset 1[11] is a real-world credit card transaction dataset containing over 550,000 transactions and Dataset 2 [12] is a simulated dataset containing 1,000 transactions. Both datasets label transactions as either fraudulent or legitimate. The real-world dataset was obtained from European customers' anonymous credit card transactions, while the simulated dataset comprises transactions of 1000 clients' credit cards when they make purchases from a pool of 800 businesses. Brandon Harris's Sparkov Data Generation Github tool was used to generate this.

3.2 Data Preprocessing

To address missing values and inconsistencies, we employed data cleaning techniques, followed by oversampling methods like SMOTE to balance the datasets. Feature engineering was also performed to create new attributes, such as the age of merchants and distance between merchant and home location. The datasets were split into training and testing sets using stratified sampling to ensure that both legitimate and fraudulent transactions were represented in each subset. Three different split ratios (80:20, 70:30 and 60:40) were experimented with.

Handling Imbalanced Data: the following preprocessing techniques were used to handle the imbalance of the two-datasets used:

1. *Random Over-sampling*: This involved increasing the number of fraud cases to match the non-fraud cases. It involves randomly duplicating instances from the minority class (the less represented class) to balance the class distribution, This technique improved precision but at the cost of recall.
2. *SMOTE (Synthetic Minority Over-sampling Technique):* This technique performed the best in balancing precision and recall. It is an oversampling technique to equalize the class distribution of the dataset by generating synthetic minority class samples. SMOTE attempts to solve this issue of imbalance by creating new minority instances between the current instances. It generates the virtual training records by linear interpolation for the minority class. These synthetic training records are constructed by randomly selecting one or more of the k-nearest neighbors for each minority class example. After oversampling, the data is reconstructed, and various classification models can be applied to the processed data. It is a powerful technique for addressing class imbalance, particularly useful in scenarios where the minority class is of great interest such as fraud detection, anomaly detection, and facial recognition [13].

This particular method involves the following steps: First, for a positive class sample X_i, it computes its distance from other positive class samples. Next, it randomly selects a sample X_j from the k-nearest neighbor samples of the positive class. Finally, it generates new samples using the following formula:

$$X_m = X_i + \text{rand}(0,1) \times (X_j - X_i)$$

Where in the provided context:

1. m represents a newly generated sample.
2. i represents the original positive class sample for which a new sample is being generated.
3. j represents a randomly selected sample from the k-nearest neighbors of X_i within the positive class [14].

3.3 Machine Learning Models

Logistic regression is a widely-used statistical technique primarily employed for binary classification tasks, predicting the probability of

an instance belonging to a specific class. Despite its given name, logistic regression is applied to classification, not regression. It models the relationship between independent features and the dependent target using the logistic function, which takes real values and maps it to a number between 0 and 1, and hence is suitable for estimating probability. The formula for the logistic function is:

$$\sigma(z) = 1 / (1 + e \wedge -z)$$

Where:
$\sigma(z)$ is the output of the logistic function.
e is the base of the natural logarithm.
z is the linear combination of the features and their respective weights, also known as the log-odds or the logits.

The log-odds (logit) z can be expressed as:

$$z=b_0+b_1x_1+b_2x_2+...+b_nx_n$$

Where:

1. $b_0,b_1,...,b_n$ are the coefficients (weights) associated with each feature.
2. $x_1,x_2,...,x_n$ are the values of the features [14].

*Decision Tree***:** The Decision tree algorithm, a supervised learning method used for regression and classification tasks, builds a tree data structure where each internal node is a feature, each decision is a branch depending on the feature, and each leaf node is the result (class label or regression value) that is predicted. The rule for a decision tree for classification is to determine the best attribute to split on at each node in order to maximize information gain. The decision tree algorithm selects the attribute that provides the maximum information gain at each node for splitting. This recursive process continues until the stopping criterion is met, i.e., the maximum tree depth, insufficient samples in a node, or when no split can provide significant improvement in information gain.

The primary goal of the decision tree algorithm is to attain the highest information gain and the attribute with the highest information gain is chosen for initial split. This information gain is computed using the following formula: Information Gain equals Entropy of the entire dataset minus the weighted average of the entropy of each feature. In simpler terms, the algorithm evaluates how much uncertainty (entropy) is reduced by splitting based on a particular attribute compared to the initial uncertainty in the dataset [15].

Random Forest: The Random Forest (RF) algorithm constructs a "forest" composed of multiple decision trees. Within the RF classifier class, when a node is split, the best feature from a randomly chosen subset of features is picked by the algorithm, rather than always selecting the most important feature. This random selection process introduces a significant amount of variability, leading to the creation of a more robust and effective model [16][17][18].

*XGBoost***:** Is an ensemble technique that combines several weak decision tree models to create a powerful prediction model. It is the application of the gradient boosting decision tree algorithm. By iteratively training new decision trees on the residual errors of the prior trees, XGBoost aims to gradually enhance the model's overall performance.
The process includes:

1. Initializing the model with a single decision tree or with a constant value.
2. The residual errors are calculated as differences between the actual and predicted values for all instances within the training data set.
3. The new decision tree will be fitted to the computed residual errors by minimizing a regularized objective function.
4. The model is updated by adding the new decision tree with a learning rate (shrinkage factor) in order to control the contribution of the new tree.
5. Run the second to the fourth step until a given maximum number of iterations or the application of a certain stopping criterion.

One of the notable advantages of XGBoost is that it is efficient in dealing with sparse data, which makes it a very appropriate method for datasets with high dimensions. It also builds in a variety of advanced techniques that prevent overfitting, among them regularization, column subsampling, and row subsampling. The sum of a loss function serves as XGBoost's objective function, which measures the difference between the actual and predicted values such as logistic loss for classification and squared loss for regression, and

the regularization term, which is used to control the complexity of the model. This term includes L1 and L2 regularization, which has the effect of shrinking the weights of less important features, thereby controlling overfitting.

XGBoost supports parallel and distributed computing. Therefore, as the size of data keeps increasing, XGBoost will enable computation with large datasets by enhancing hardware capabilities [19].

The primary challenge was the imbalanced nature of the datasets, where the number of non-fraud cases significantly outnumbered the fraud cases.

3.4 Performance Metrics for Fraud Detection

The performance of machine learning models in credit card fraud detection is evaluated with a variety of metrics, especially because most fraud datasets are imbalanced. Under such conditions, straightforward accuracy becomes misleading. The following are the critical performance metrics used here:

1. Accuracy

Accuracy is total correct predictions divided by total predictions made. Though it provides a quick general idea, it isn't always reliable in fraud detection due to class imbalance (fraud cases are much fewer than legitimate ones).

$$\text{Accuracy} = \{TP + TN\}/\{TP + TN + FP + FN\}$$

In an imbalanced dataset, a model that assigns a probability of being legitimate to each transaction with high accuracy can be useless in fraud detection despite high accuracy.

Cheng & Xiong [8] noted that even though their CS-SVM was extremely accurate, they emphasized other statistics like precision and recall for enhanced understanding in the context of fraud.

2. Precision

Precision is the ratio of correctly anticipated fraud instances to the total anticipated fraud instances. Precision computes the number of positive true predictions.

$$\text{Precision} = \{TP\}/\{TP + FP\}$$

Ali et al. [3] highlighted precision as one of the required metrics in their systematic review of ML-based fraud detection, especially when false positives are costly.

3. Recall (Sensitivity or True Positive Rate)

Recall calculates the proportion of actual fraud cases correctly anticipated by the model.

$$\text{Recall} = \{TP\}/\{TP + FN\}$$

High recall guarantees the majority of frauds are detected. It is important because not detecting a fraudulent transaction has a very high cost. Vanini et al. [8] optimized recall to ensure low fraud-related losses through payment channels, illustrating the metric's ability in business risk reduction.

4. F1-Score

The F1-score is the harmonic mean of precision and recall. It provides a single value that truly represents the trade-off between precision and recall.

F1 Score: 2×Precision×Recall/Precision + Recall

4. Results and Discussion

4.1 Results

4.1.1 Performance of the ML models on the Fraud Dataset (Dataset 1)

The performance of the models as seen in the Tables 2 to 7 was evaluated using precision, recall, F1-score, and Accuracy as metrics.

Table 2 presents the result of the models when trained and tested on the imbalanced dataset. Logistic Regression performed very poorly with zero recall, precision, and F1-score for all ratios tested, namely 80/20, 70/30, and 60/40. However, it kept a high accuracy. Decision Tree showed consistent performance across different splits, with recall values ranging from 0.6525 to 0.6772, and F1-scores around 0.6373 to 0.6507. Random Forest model had higher precision of 0.8859 at 80/20 split but showed variability in recall (ranging from 0.5202 to 0.6549), leading to fluctuating F1-scores. XGBoost performed best compared to other models. Its recall, precision, and F1-score were high across all ratios tested with similar high accuracy levels compared to the other model. Figure 2 presents a visualization of this result.

Table 3 presents the result on the SMOTE balanced dataset. All models showed significant improvement. Decision Tree and Random Forest achieved near-perfect scores across all metrics, demonstrating exceptional performance on balanced data. Logistic Regression improved considerably but still lagged behind the other models in overall performance. XGBoost showed outstanding results, with recall, precision, and

F1-scores close to the perfect score. It proved to be resilient for balanced data and outperformed Logistic Regression and Random Forest while its results was very close to Decision Tree. Figure3 presents the visualization of this result.

Table2: Results on the Imbalanced Dataset 1

	Split ratio	Recall	Prec.	F1	Acc.
LR	80/20	0.00	0.00	0.00	0.9943
	70/30	0.00	0.00	0.00	0.9943
	60/40	0.00	0.00	0.00	0.99443
DT	80/20	0.6772	0.6258	0.6507	0.99621
	70/30	0.6549	0.6263	0.6403	0.99616
	60/40	0.6525	0.6227	0.6373	0.99613
RF	80/20	0.5202	0.8859	0.6480	0.99705
	70/30	0.6549	0.6263	0.6403	0.99616
	60/40	0.6525	0.6227	0.6373	0.99613
XG	80/20	0.6621	0.8571	0.7471	0.99766
	70/30	0.6469	0.8467	0.7335	0.99755
	60/40	0.6463	0.8406	0.7308	0.99751

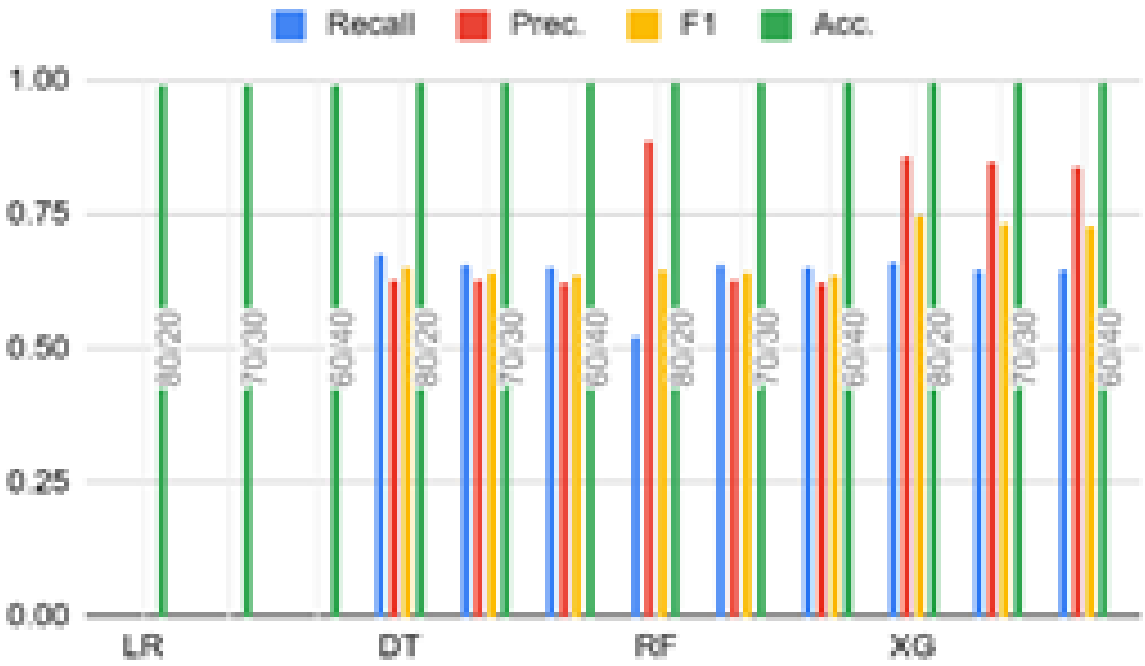

Figure 2: Comparison of Four Machine Learning Models (LR=Logistic Regression, DT=Decision Tree, RF= Random Forest, and XG=XGBoost) on four Evaluation Metrics on Dataset 1 when unbalanced

Table3: Results on the Balanced Dataset 1 using SMOTE

	Split ratio	Recall	Prec.	F1	Acc.
LR	80/20	0.7584	0.94067	0.839	0.85530
	70/30	0.7588	0.94067	0.8400	0.85551
	60/40	0.7589	0.9088	0.8401	0.994432
DT	80/20	0.9979	0.9958	0.9968	0.9968
	70/30	0.9980	0.9960	0.9970	0.9970
	60/40	0.9979	0.9957	0.9968	0.9968
RF	80/20	0.9326	0.9854	0.9583	0.9594
	70/30	0.93247	0.985057	0.95804	0.959164
	60/40	0.9337	0.9856	0.9589	0.9601
XG	80/20	0.994695	0.993482	0.994089	0.994085
	70/30	0.99449	0.9935	0.99399	0.993992
	60/40	0.994439	0.9935	0.993969	0.993966

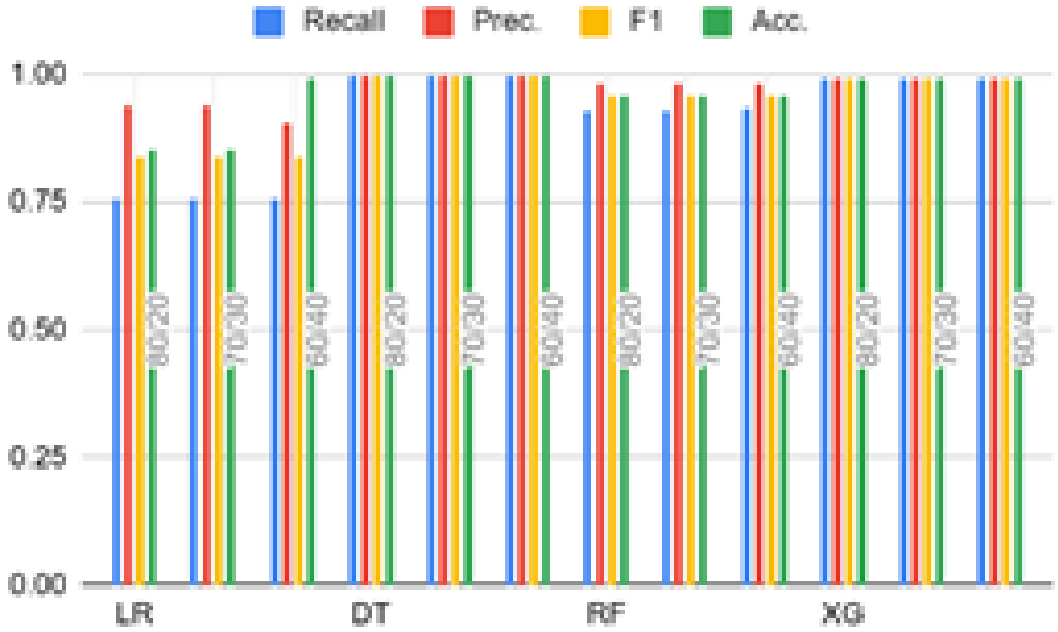

Figure3: Comparison of Four Machine Learning Models (LR=Logistic Regression, DT=Decision Tree, RF= Random Forest, and XG=XGBoost) on four Evaluation Metrics on Dataset 1 after balanced with SMOTE.

Table4: Results on the Balanced Dataset 1 using Random Sampling

	Split ratio	Recall	Prec.	F1	Acc.
LR	80/20	0.7588	0.9414	0.8403	0.855837
	70/30	0.7566	0.9396	0.8382	0.853996
	60/40	0.7587	0.9415	0.8403	0.85584
DT	80/20	1.0	0.9979	0.9989	0.998974
	70/30	1.0	0.9978	0.9989	0.998947
	60/40	1.0	0.997618	0.998808	0.998806
RF	80/20	0.919989	0.990459	0.953924	0.955564

	70/30	0.925687	0.988629	0.956123	0.95752
	60/40	0.933599	0.989271	0.960629	0.961737
XG	80/20	0.998722	0.982637	0.990614	0.990537
	70/30	0.998546	0.981957	0.990182	0.990099
	60/40	0.998474	0.983194	0.990775	0.990703

Table4 shows the performance of the models when trained and tested on the Random Sampling balanced dataset. Comparing Random sampling and SMOTE balancing technique, the models achieved more F1 score and Accuracy with SMOTE.

4.1.2 Performance of the ML models on the Card Dataset (Dataset 2)

Table 5presents the result for the imbalanced dataset. Logistic Regression's best performance was 0.9991 in accuracy, with an F1-score of 0.6580 at a 70/30 split. Decision Tree and Random Forest showed perfect or near-perfect scores in accuracy. The XGBoost model's performance was high on this imbalanced dataset, giving 0.9995 in accuracy and 0.8353 as the F1-score when the split is 60/40, hence proving to be robust for handling imbalance with much efficiency. Figure4 presents the visualization of this result.

Table5: Results on the Imbalanced Dataset 2

	Split ratio	Recall	Prec.	F1	Acc.
LR	80/20	0.5789	0.8461	0.6875	0.999093
	70/30	0.5352	0.8539	0.6580	0.999045
	60/40	0.5555	0.8536	0.6531	0.999075
DT	80/20	0.6841	0.7647	0.7222	0.999093
	70/30	0.7464	0.7361	0.7412	0.999105
	60/40	0.7407	0.7567	0.7486	0.999148
RF	80/20	0.7578	0.9113	0.8275	0.999456
	70/30	0.7464	0.7361	0.7412	0.999105
	60/40	0.7407	0.7567	0.7486	0.999148
XG	80/20	0.74736	0.91025	0.82080	0.999438
	70/30	0.753521	0.92241	0.82945	0.999468
	60/40	0.751323	0.94039	0.83529	0.999492

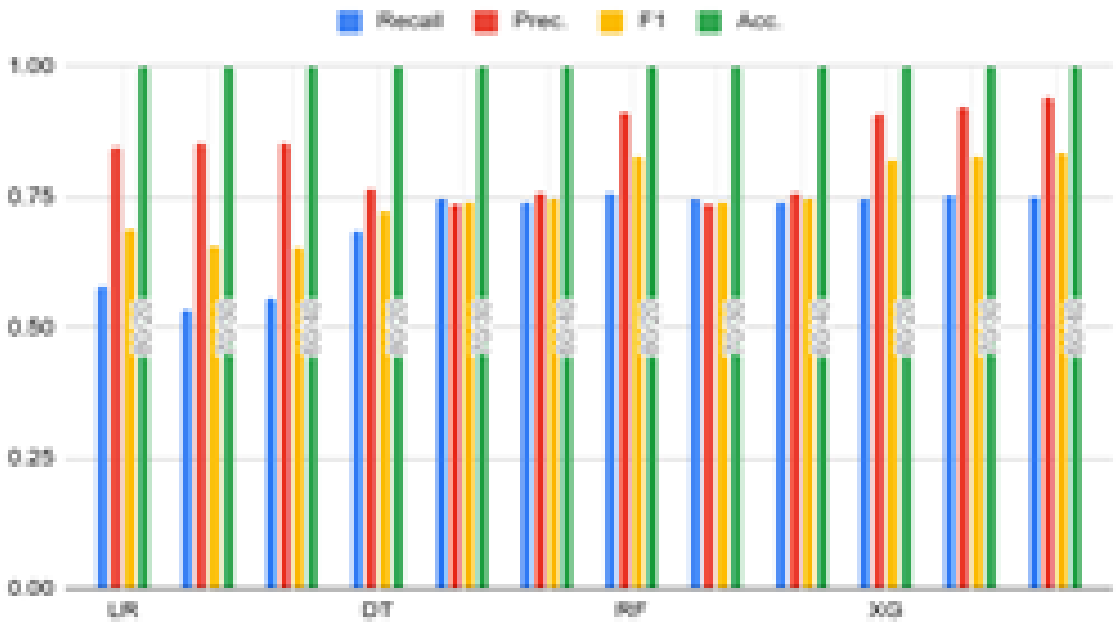

Figure 4: Comparison of Four Machine Learning Models (LR=Logistic Regression, DT=Decision Tree, RF= Random Forest, and XG=XGBoost) on four Evaluation Metrics on Dataset 2 when unbalanced

Table 6: Results on the Balanced Dataset 2 using SMOTE

	Split ratio	Recall	Prec.	F1	Acc.
LR	80/20	0.9368	0.9812	0.9585	0.95945
	70/30	0.9363	0.9796	0.9574	0.95843
	60/40	0.9360	0.9798	0.9574	0.95843
DT	80/20	0.9991	0.9973	0.9982	0.99823
	70/30	0.9986	0.9972	0.9979	0.99794
	60/40	0.9983	0.9967	0.9975	0.99754
RF	80/20	0.9914	0.9991	0.9952	0.99526
	70/30	0.9919	0.9990	0.9954	0.99551
	60/40	0.9922	0.9991	0.9957	0.99573
XG	80/20	1.0	0.9996	0.9998	0.9998
	70/30	1.0	0.9994	0.9997	0.99973
	60/40	1.0	0.9995	0.9997	0.99975

Table 6 presents the result for the SMOTE balanced dataset. The result shows that all models improved drastically. Decision Tree and Random Forest recorded almost perfect scores in all metrics, proving to be very effective for handling balanced data. Logistic Regression gained a bit, though always outperformed by the other models. With a 60/40 split, it had an accuracy of 0.9595 and an F1-score of 0.9585. XGBoost had perfect recall, precision, and F1 scores with a 60/40 split, which proves its capability to handle balanced data with ease. Figure 5 presents the visualization of this result.

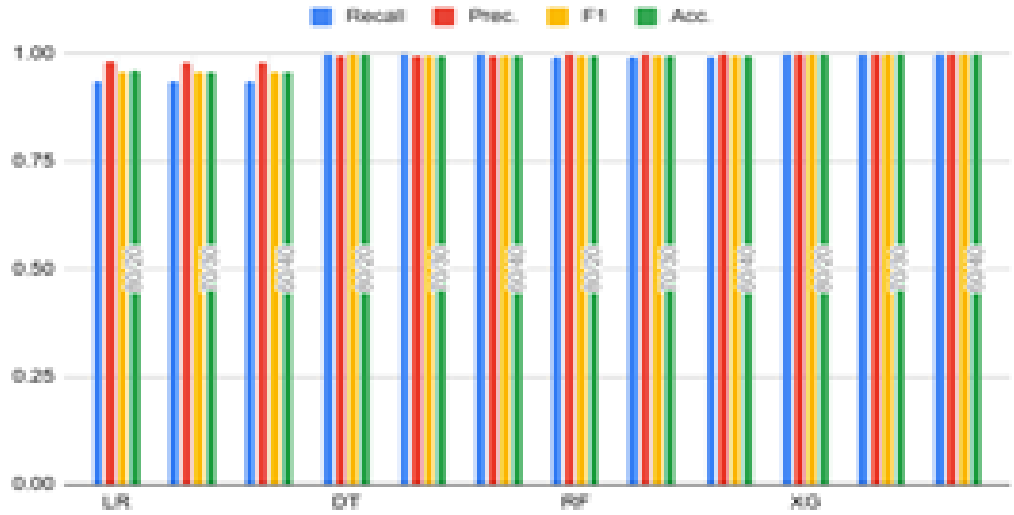

Figure 5: Comparison of Four Machine Learning Models (LR=Logistic Regression, DT=Decision Tree, RF= Random Forest, and XG=XGBoost) on four Evaluation Metrics on Dataset 2 after balanced with SMOTE

Table7: Results on the Balanced Dataset 2 using Random Sampling

	Split ratio	Recall	Prec.	F1	Acc.
LR	80/20	0.9128	0.9762	0.9434	0.945329
	70/30	0.9148	0.9753	0.9441	0.945898
	60/40	0.9188	0.9745	0.9458	0.947441
DT	80/20	1.0	0.999492	0.999746	0.999746
	70/30	1.0	0.999443	0.999721	0.999721
	60/40	1.0	0.999283	0.999641	0.999641
RF	80/20	0.997547	0.99949	0.998518	0.998519
	70/30	1.0	0.99954	0.99977	0.99977
	60/40	1.0	0.999564	0.999782	0.999782
XG	80/20	1.0	0.999746	0.999873	0.999873
	70/30	1.0	0.999734	0.999867	0.999867
	60/40	1.0	0.999728	0.999864	0.999864

Table 7 presents the result of the four models when trained on the Random Sampling balanced dataset. Logistic Regression achieved high recall of 0.91, precision of 0.97, and F1-score of 0.94 across all split ratios, with accuracy between 94-95%. Decision Tree delivered a perfect recall of 1.0, a near-perfect precision of 0.999 and F1-scores of 0.9997, with accuracy consistently above 99.9%. Random Forest maintained near-perfect recall scores between 0.99-and 1.0, precision of 0.999, and F1-scores between 0.998 and 0.999, with accuracy consistently near 99.9%. XGBoost showed perfect recall of 1.0, exceptional precision of 0.9997, and F1-scores of 0.9998, with accuracy reaching close to 99.99%.

4.2 Discussion

The overview of the results using both datasets reflects that ensemble techniques like XGBoost and Random Forests are bound to turn out successful in identifying fraud in an imbalanced dataset. These techniques can learn complex patterns and relationships within the data – a key aspect of fraud detection. The results indicate that XGBoost performed best among the compared models, including Random Forest, for both the imbalanced and SMOTE-balanced datasets. On the imbalanced dataset 1, XGBoost proved resilient, with a F1-score of 0.75 and an accuracy of 0.9977 at 80/20 split. On the balanced dataset 1, XGBoost had a performance outcome with an accuracy of 0.9940 and an F1-score of 0.9940 at

80/20 split, On the imbalanced dataset 2, XGBoost proved resilient, with a F1-score of 0.84 and an accuracy of 0.9994 at 60/40 split and on the balanced dataset 2, XGBoost had a performance outcome with an accuracy of 0.9998 and an F1-score of 0.9998 at 80/20. Its superior performance displays its capability in managing imbalanced data with great precision and recall. XGBoost also performed better on the SMOTE-balanced dataset and gave perfect recall, near perfect precision hence enhancing its capacity to effectively handle balanced data. It indicates that XGBoost does not only handle imbalanced data effectively, but it also performs very well when datasets are balanced. It is evidenced from the results that using the proposed hybrid data augmentation approach, XGBoost is the better-performing and more robust model overall, as it consistently achieves higher F1 scores and accuracy across all datasets and splits.

Logistic Regression faces a problem when it has to work with imbalanced datasets, whereas Decision Tree, Random Forest, and XGBoost are more robust to this kind of imbalance, and balancing significantly increases their power. Decision Tree and Random Forest showed huge improvements with the SMOTE technique, with nearly perfect scores across all metrics, XGBoost is consistently on top. These results show that applying SMOTE in balancing the data significantly enhances the efficiency of fraud detection models. The SMOTE data balancing method generated higher precision, F1-score, and accuracy compared to Random Sampling data balancing method hence showing its superiority in data augmentation for an imbalanced dataset.

XGBoost amounts to be a powerful and trustworthy model in fraud detection for imbalanced datasets because of the superiority in different splits and balancing strategies concerning its metrics. The use of strong machine learning algorithms such as Random Forest or XGBoost together with effective techniques for dealing with imbalanced data like SMOTE enhances the accuracy of fraud identification in transactions. Further research in similar lines can explore advanced deep learning techniques and feature engineering in detail to improve the models' performance.

5. Conclusion

In this study, we compared the performance of several machine learning models for fraud detection in imbalanced credit card transaction datasets. Our findings show that ensemble models like Random Forest and XGBoost consistently outperform traditional models, even when dealing with heavily imbalanced datasets. Data preprocessing techniques, particularly SMOTE, played a key role in improving the performance of the models.

Although machine learning for credit card fraud detection has greatly improved, there are still a good number of challenges and limitations. One major challenge is that fraud patterns and techniques change rapidly; hence, making the models developed previously ineffective after some time. One active area of research is in the development of adaptive or online learning models that can update themselves continuously to improve performance with the availability of new data. This would imply that data quality problems may appear as missing or inconsistent features and thus affect the performance of machine learning models. Supplementing it with more sources of data, such as patterns of customer behavior or metadata on how transactions are made, might make a fraud detection system more accurate.

The other challenge is planning to make complex machine learning models, especially the deep neural networks interpretable. To build interpretable models or be equipped with techniques that could explain the decisions of fraud detection models are essential to build trust and comply with important regulations.

Future research should focus on developing adaptive models that can continuously learn from new fraud patterns, further enhancing fraud detection systems. In addition, future research directions can be towards exploring new architectures and algorithms designed especially for imbalanced datasets, like meta-learning or few-shot learning approaches, and focus more on feature engineering techniques to enhance the efficiency of developed models.

References

[1] Al-Khater, W. A., et al. (2020). Comprehensive review of cybercrime detection techniques. *IEEE Access*, 8, 137293–137311. https://doi.org/10.1109/

[2] Kennedy, K., et al. (2023). "Iterative cleaning and learning of big highly-imbalanced fraud data." *Journal of Big Data*, 10:106

[3] Aburbeian, A. H. M., & Ashqar, H. I. (2023). *Credit card fraud detection using enhanced random forest classifier for imbalanced data* [Preprint]. arXiv. https://arxiv.org/abs/2303.06514

[4] Baesens, B., Verbeken, B., Bravo, C., & Santafé, G. (2020). robROSE: A robust approach for dealing with imbalanced data in fraud detection [Preprint]. arXiv. https://arxiv.org/abs/2003.11915

[5] Sakharova, I. (2012). Payment card fraud: Challenges and solutions. 2012 IEEE International Conference on Intelligence and Security Informatics. https://doi.org/10.1109/isi.2012.6284315

[6] Cheng, J., & Xiong, Y. (2023). Multi-strategy adaptive cuckoo search algorithm for numerical optimization. *Artificial Intelligence Review*, 56(3), 2031–2055. https://doi.org/10.1007/s10462-022-10222-4

[7] Ali, A., et al. (2022). Financial fraud detection based on machine learning: A systematic literature review. *Applied Sciences*, 12(19), 9637. https://doi.org/10.3390/app12199637

[8] Vanini, P., Rossi, S., Zvizdic, E., & Domenig, T. (2023). Online payment fraud: From anomaly detection to risk management. *Financial Innovation*, 9(66). https://doi.org/10.1186/s40854-023-00470-w

[9] Sandhya, G., & Priya, S. R. (2023). Credit card fraud detection using machine learning algorithms. *Data Analytics and Artificial Intelligence*, 3(3), 130–133. https://doi.org/10.46632/daai.3.3.18

[10] Btoush, E. A. L. M., et al. (2023). A systematic review of literature on credit card cyber fraud detection using machine and deep learning. PeerJ Computer Science, 9, e1278. https://doi.org/10.7717/peerj-cs.1278

[11] Elgiriyewithana, N. (2023). Credit Card Fraud Detection Dataset 2023 [Data set]. Kaggle.https://www.kaggle.com/datasets/nelgiriyewithana/credit-card-fraud-detection-dataset-2023

[12] Shenoy, K., & Harris, B. (2020). Credit Card Transactions Fraud Detection Dataset. Kaggle. https://www.kaggle.com/datasets/kartik2112/fraud-detection

[13] Li, K., et al. (2014). An improved smote imbalanced data classification method based on support degree. 2014 International Conference on Identification, Information, and Knowledge in the Internet of Things. https://doi.org/10.1109/iiki.2014.14

[14] Zaki, M. J., & Meira, W. (2020). Data Mining and Machine Learning: Fundamental Concepts and Algorithms. *Cambridge University Press*

[15] Costa, V. G., & Pedreira, C. E. (2022). Recent advances in decision trees: an updated survey. *Artificial Intelligence Review*. *56*(5), 4765-4800

[16] Aktar, H., et al. (2021). Classification using Random Forest on imbalanced credit card transaction data. 2021 3rd International Conference on Sustainable Technologies for Industry 4.0 (STI). https://doi.org/10.1109/sti53101.2021.9732553

[17] Chen, T., & Guestrin, C. (2016). XGBoost: a Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16, 785–794. https://doi.org/10.1145/2939672.2939785

[18] Akanmu, I., & Odumuyiwa, V. (2022). Performance evaluation of machine learning models for detection of Mirai infection in IoT devices. *Journal of Scientific Research and Development*, *21*(1), 135-145.

[19] Igwilo, C. M., & Odumuyiwa, V. T. (2022). Comparative analysis of ensemble learning and non-ensemble machine learning algorithms for phishing URL detection. *FUOYE Journal of Engineering and Technology*, *7*(3), 305-312.