Credit Card Fraud Detection Model-based Machine Learning Algorithms

Amira M. Idrees *, Nermin Samy Elhusseny, Shimaa Ouf

Abstract—**Fraud detection plays a crucial role in the modern banking sector, aiming to mitigate financial losses affecting both individuals and financial institutions. With a significant portion of the population regularly using credit cards, efforts to enhance financial inclusivity have led to increased card usage. Additionally, the rise of e-commerce has brought about a surge in credit card fraud incidents. Unfortunately, traditional statistical methods used for identifying credit card fraud are time-consuming and may not provide accurate results. As a result, machine learning algorithms have become widely adopted for effective credit card fraud detection. This study addresses the challenge of an imbalanced credit card dataset by employing three sampling strategies: cluster centroid-based majority under-sampling technique (CCMUT), synthetic minority oversampling technique (SMOTE), and oversampling technique. The training dataset is then used to train nine machine learning algorithms, including Random Forest (RF), k nearest neighbors (KNN), Decision Tree (DT), Logistic Regression (LR), Ada-boost, Extra-trees, MLP classifier, Naive Bayes, and Gradient Boosting Classifier. The performance of these approaches is assessed using metrics such as accuracy, precision, recall, f1 score, and f2 score. The dataset used in this study was obtained from the Kaggle data repository.**

Index Terms—**credit card, fraud detection, imbalanced dataset, machine learning algorithms**

I. Introduction

Credit cards have become an integral mode of payment in contemporary society. Amidst the rapid growth in credit card growth, credit card fraud is on the rise. While credit card transaction data exhibits some imbalance, fraud transaction data is considerably less uneven than regular transaction data [1], [2]. The global adoption of credit cards has initiated a shift towards financial inclusion, aiming to integrate marginalized individuals into the financial sector. This widespread usage has resulted in a surge in credit card users, consequently contributing to an elevated incidence of credit card fraud. Annually, over 10.7 million individuals fall victim to credit card fraud, the most prevalent form of identity theft [3]. The surge in credit card usage can be attributed to various factors, including convenience, purchasing capability, market expansion, user-friendly features, simplicity, credit card perks, price protection, purchase security, and travel benefits. In response, the payment industry is introducing an increasing number of digital payment solutions. However, this trend may lead to an increase in fraud, resulting in financial losses, and it is susceptible to online fraudulent activities. Card fraud remains a concerning aspect of the digital era due to the ease with which criminals can obtain credit cards for illicit purposes. Once they acquire your personal information, such as your credit card number, committing fraud becomes a simple task. The global incidence of credit card fraud and the associated per-dollar losses are projected from 2013 to 2027. The anticipated worldwide cost of credit card fraud is set to reach $38.5 billion by 2027, a significant increase from the $13.7 billion reported in 2013. In 2021, the estimated cost of credit card fraud was $32.4 billion, equivalent to 7.1 cents per $100 in volume [4], [5].

Financial fraud persists as a pervasive issue with extensive repercussions for the financial industry, businesses [6], education [7], corporate entities, and governmental institutions. The proliferation of credit card transactions has experienced a notable increase, driven by the expanding adoption of e-commerce in the business sector. The rise in fraudulent transactions is attributed to vulnerabilities inherent in existing e-banking systems. Detection and prevention are widely acknowledged as the most effective strategies for mitigating fraudulent activities. Implementing an additional layer of defense helps thwart attacks by fraudsters, and when preventive measures prove inadequate, the detection process comes into play. Therefore, detection enables the prompt identification and notification of a fraudulent transaction upon initiation. Transactions can be categorized into two types: authentic, which are lawful and valid, and fraudulent, involving deceit or deception. Credit card fraud is further classified into two types: interior fraud and exterior fraud. Internal fraud occurs when a bank employee is linked to a customer using a fabricated identity. External fraud involves criminals using stolen credit cards for unauthorized purchases. The primary goal of credit card fraud detection is to accurately identify instances of fraudulent activity through dataset analysis. The decision-making process is deemed challenging due to highly skewed and unbalanced datasets. Dataset providers acknowledge privacy and security concerns, often utilizing predominantly numerical columns with limited inclusion of alphabetic

Manuscript received June 23, 2023; revised July 30, 2024.

This work is funded by the Deanship of Research and Graduate Studies at King Khalid University through General Research Project under grant number GRP/50/45

Amira M. Idrees is a professor, College of Business, King Khalid University, Saudi Arabia. (Corresponding author; email: aidrees@kku.edu.sa).

Nermin Samy Elhusseny is a postgraduate student, Faculty of Commerce and Business Administration, Helwan University, Egypt.
(e-mail: nerminelhusseny5@gmail.com).

Shimaa Ouf is an Assistant Professor, Faculty of Commerce and Business Administration, Helwan University, Egypt.
(email:shimaaouf@commerce.helwan.edu.eg).

properties. Fraudulent transactions often resemble genuine ones, posing a significant challenge during the credit card identification process [8].

The contributions were made to the field of credit card fraud detection in this research by employing a comprehensive approach that integrated nine distinct machine learning techniques and three sampling techniques to tackle the challenges posed by highly imbalanced datasets. The contributions can be summarized as follows:

A. Diverse Machine Learning Techniques

A rich set of nine machine learning techniques was employed, ensuring a broad exploration of the model landscape. The diversity of these algorithms, including but not limited to Decision Trees, Random Forest, Extra Tree, and Decision tree , allowed a wide range of patterns inherent in credit card transaction data to be captured.

B. Imbalance Mitigation through Sampling Techniques

Recognizing the imbalanced nature of credit card fraud datasets, three state-of-the-art sampling techniques—Cluster Centroid, SMOTE (Synthetic Minority Over-sampling Technique), and Random Over-sampler—were incorporated. These techniques aimed to rectify the class imbalance, fostering better model generalization and reducing the risk of biased predictions towards the majority class.

C. Comprehensive Performance Evaluation Metrics

To thoroughly assess the effectiveness of the models, a set of robust performance metrics—accuracy, recall, precision, F1-score, and F2-score—was employed. These metrics provide a nuanced understanding of model performance, accounting for both false positives and false negatives—critical aspects in the context of credit card fraud detection.

D. Holistic Evaluation Approach

The research extends beyond a sole emphasis on accuracy, recognizing the nuanced requirements of fraud detection scenarios. By incorporating recall, precision, and F-scores, it provides a more comprehensive evaluation that considers the trade-offs between accurately identifying fraud cases and minimizing false positives.

E. Empirical Study on Sampling Techniques

An empirical study was conducted to analyze the impact of each sampling technique on the performance of the machine learning models. This analysis not only sheds light on the effectiveness of individual techniques but also provides insights into their suitability for credit card fraud detection.

The research is structured into eight sections: Introduction (Section 1), Related Work (Section 2), Problem Formulation (Section 3), Proposed Framework (Section 4), Methodology (Section 5), Implementation (Section 6), Experimental Results (Section 7), and Conclusion (Section 8).

II. RELATED WORK

Garg V., et al. [3] conducted a research study that explored various machine learning approaches, emphasizing the emerging field of auto-machine learning technology. The study aimed to comprehend the widely used auto-machine-learning technology by comparing manual and automated machine learning methods. The proposed techniques included pre-processing, oversampling, splitting the dataset into test and train subsets, feature selection, and AUTO ML, which played a crucial role in the model. The models were tested with Extra Trees, Random Forest, Linear Discriminant Analysis, Ada Boost, Logistic Regression, Decision Tree, Ridge Classifier, Gradient Boosting, KNN, SVM-Linear Kernel, Light Gradient Boosting, Naive Bayes, and Quadratic Discriminant Analysis. The research findings highlighted that Extra Trees outperformed other models in terms of accuracy, f-1 score, recall, time, and precision, achieving an impressive accuracy rate of 99.96%. The study suggested that software businesses could contribute essential data to facilitate fair and accurate comparisons and evaluate assessment techniques in the realm of auto-machine learning.

Nadim, A.H. et al. [8] conducted a study testing six machine learning models for credit card fraud detection, including XG-Boost, Linear Discriminant Analysis (LDA), Logistic Regression, Support Vector Machine, Random Forest (RF), K-Nearest Neighbors (KNN), and Logistic Regression. The research utilized accuracy, sensitivity, precision, and specificity as evaluation metrics. The dataset, derived from European cardholder data, encompassed two-day transactions in September 2013, totaling 284,807 entries. To balance the dataset, the positive class—fraud cases, constituting 0.172 percent of transaction data—was emphasized. Four metrics, TPR, TNR, FPR, and FNR, were employed for performance evaluation. Random Forest outperformed XG-Boost (98.4%), logistic regression (97.7%), support vector machines (97.5%), linear discriminant analysis (97.4%), k-nearest neighbors (96.9%), and classification and regression trees (58.6%). Based on the study's findings, the authors intend to incorporate a genetic algorithm, rigorous feature selection, and layered classifier approaches in their future research

Uchhana N et al. [9] conducted a comparative analysis of various machine learning algorithms for credit card fraud detection, including support vector machines (SVM), logistic regression, naive Bayes, K-nearest neighbors (KNN) classifiers, and random forest. The random forest algorithm achieved the highest score, followed by the K-nearest neighbors (KNN) algorithm. The evaluation metric used in this study is the Matthews Correlation Coefficient (MCC), which ranges from -1 to 1, with 1 being the best possible score. The Random Forest algorithm demonstrated the highest MCC score of 0.89 according to their findings. The maximum achievable MCC score with randomly selected parameters for the Random Forest algorithm is 0.848. The researchers then utilized Random Forest and Grid Search methodologies to create a new model by adjusting parameters, followed by a comparative analysis to identify the most favorable configurations. The MCC value obtained for the new algorithmic solution is 0.89.

Chakshu. V et al. [10] employed support vector machines, naive Bayes, and logistic regression as analytical tools in their study, aiming to conduct a comprehensive analysis of business data derived from credit card histories with the primary goal of developing robust fraud detection algorithms. The proposed model's performance is evaluated using metrics such as accuracy, precision, recall, and F1-measure. The results suggest that the Support Vector Machine (SVM) kernel exhibits superior efficacy in credit card fraud detection, achieving an accuracy rate exceeding 97.2% as indicated by a Receiver Operating Characteristic (ROC) graph. However, the precision rate drops to 25% when restricting the analysis to only 10% of the dataset. Notably, the algorithm demonstrates a 30% increase in accuracy when provided with the entire dataset. The study proposes the incorporation of alternative algorithms to further enhance the model.

Sadineni, P.K. [11] utilized support vector machines (SVM), decision trees, artificial neural networks (ANN), logistic regression, and random forests as analytical techniques in their study. The evaluation criteria included accuracy, precision, and false alarm rate to assess the performance of these techniques. Principal component analysis (PCA) was employed to eliminate extraneous variables and retain essential ones, such as transaction time, amount, class, and other relevant factors. The dataset, sourced from Kaggle, consisted of 150,000 transactions. Evaluation relied on analyzing true positive, true negative, false positive, and false adverse outcomes. Results indicate that Random Forest and Decision Tree achieved accuracy rates of 99.21% and 98.47%, respectively. SVM achieved an accuracy rate of 95.16%, while logistic regression achieved 95.55%. The ANN outperformed both models with a 99.92% accuracy rate. While the ANN model produces accurate outcomes, its training process is arduous and costly. SVM demonstrates exceptional performance even with limited-sized datasets. Logistic regression excels with unprocessed, unaltered data, while the decision tree method performs well with sampled and pre-processed data. The Random Forest algorithm is highly suitable for handling both categorical and continuous data.

Ignatius. J et al. [12] utilized a dataset sourced from Kaggle, comprising 150,000 transactions for their experiment. Evaluation of machine learning methods for detecting fraudulent transactions involved analyzing true positive, true negative, false positive, and false adverse outcomes. Results demonstrated that Random Forest and Decision Tree achieved accuracy rates of 99.21% and 98.47%, respectively. The support vector machine (SVM) attained an accuracy rate of 95.16%, while logistic regression demonstrated a slightly higher accuracy rate of 95.55%. In contrast, the artificial neural network (ANN) outperformed both models with an impressive accuracy rate of 99.92%. While the ANN model produces accurate outcomes, its training process poses significant difficulties and incurs substantial costs. Support vector machines (SVM) demonstrated exceptional performance even with datasets of limited size. Logistic regression excels with unprocessed, unaltered data, while the decision tree method exhibits enhanced performance with sampled and pre-processed data. The Random Forest algorithm is a highly suitable choice for handling both categorical and continuous data. Findings regarding the Isolation Forest and Local Outlier Factor algorithms indicate that Isolation Forest exhibits superior performance in detecting credit card fraud, achieving a peak accuracy rate of 97%, while the local outlier factor measured at 76%.

Anand, H., R. et al. [13] conducted a comparative analysis to assess the performance of the Isolation Forest and Local Outlier Factor algorithms in detecting credit card fraud. The study revealed that the Isolation Forest algorithm outperformed the Local Outlier Factor algorithm, achieving a peak accuracy rate of 97%, while the Local Outlier Factor reached 76%. The accuracy of the Local Outlier Factor (LOF) algorithm was 99.67%, with a total of 935 errors. In contrast, the Isolation Forest (IF) algorithm achieved 99.76% accuracy with a lower number of mistakes, totaling 659. Despite the Isolation Forest's ability to achieve an accuracy rate exceeding 99.6% with only a fraction of the available dataset, its precision is limited to 28%. However, this analysis is conducted using the complete dataset. The accuracy of the Isolation Forest was observed to be 99.76%, resulting in a total of 659 errors. In comparison, the Local Outlier method had 935 errors and an accuracy of 99.61%. On the other hand, the Support Vector Machine method achieved an accuracy rate of 70%. Therefore, it is evident that the Isolation Forest outperforms both the Local Outlier and the Support Vector Machine in terms of performance.

Rout, M., [14] evaluated LR, Random Forest, and Naive Bayes classifiers, utilizing accuracy, precision, F1, recall, and MCC, with a focus on F1 and MCC. The study used a dataset comprising 284,807 European cardholder transactions in September 2013. Due to the low fraud rate of 0.173%, SMOTE oversampling was applied. The experiment comprised three phases. Addressing the unbalanced dataset involved employing both a conventional model and the SMOTE approach. Random Forest outperformed XG-Boost (99.95%), logistic regression (90.93%), and naive Bayes (90.92%). Conventional models incorporated SMOTE, AdaBoost, and soft voting. A random forest with a decision tree surpassed the competition at 99.94%. Naive Bayes, Logistic Regression, XG-Boost, and 99.92% accurate models followed. Notably, the Naive Bayes + decision tree model exhibited inferior F1 and recall scores compared to Random Forest. Finally, the rate decreased, with AdaBoost being surpassed by Random Forest. In summary, XG-Boost achieved 99.95%, logistic regression 99.93%, random forest 99.96%, and naive Bayes 99.92%. Naive Bayes lagged, while Random Forest and XG-Boost models showed improvements in recall, F1, and MCC. The conventional model with AdaBoost outperformed logistic regression and naive Bayes, which had comparable F1 scores. Although the evaluation score increased marginally, the research suggested that future machine learning models could explore deep learning models, and alternative feature selection and dataset imbalance methods might enhance results.

V Kumar K S et al. [15] conducted research utilizing logistic regression, naive Bayes, decision trees, and artificial neural networks (ANN) to develop a prediction model for fraud detection. The dataset comprised European cardholder transactions from September 2013, with an imbalance in the statistics due to a higher number of fraudulent cases (492) compared to the total transactions (284,807). Principal Component Analysis (PCA) transformed the data into a numerical dataset. Logistic regression achieved the highest accuracy at 94.84%, followed by decision trees at 92.88%, and naive Bayes at 91.62%. The artificial neural network (ANN) exhibited the highest accuracy of 98.69%. The findings were tabulated using a confusion matrix, emphasizing the importance of low-false-positive algorithms for achieving the research goals.

Manohar S et al. [16] conducted a study employing support vector machines, random forests, and decision trees for credit card fraud detection. Principal Component Analysis (PCA) was utilized to identify characteristics explaining at least 95% of the variation. However, the PCA feature selection technique did not reveal significant associations or variations with the class column for detecting fraudulent transactions. The researchers utilized 2013 European cardholder transaction histories from Kaggle, comprising thirty-one columns: thirty features and one class variable. Essential components included temporal, quantitative, and transactional aspects. Support Vector Machine, Decision Tree, and Random Forest demonstrated accuracy rates of 99.8%, 99.7%, and 99.7%, respectively. Despite the current models having high accuracy, precision was low. Therefore, there is an emphasis on improving the model to achieve optimum results with high precision in credit card transaction fraud detection.

Sadgali I. et al. [17] conducted research utilizing supervised machine learning, employing Support Vector Machine, Random Forest, Decision Tree, and K-Nearest Neighbors. The study focused on a single dataset comprising fraudulent transactions, consisting of 60,000 transactions with twelve criteria, including transaction and customer data. The dataset exhibited strong skewness, with 99.72% of transactions being non-fraudulent, aiming to replicate real-life transaction circumstances to create a dataset mimicking financial data. Machine learning on the training and test datasets estimated the Mean Squared Error (MSE) for each strategy. Support Vector Machine (SVM) demonstrated MSE values ranging from 0.0021 to 0.0024. The Random Forest, K-Nearest Neighbors, and Decision Tree algorithms had MSE values of 0.0026 to 0.0028, 0.0028 to 0.0029, and 0.0027 to 0.0031, respectively. Support Vector Machines outperformed Decision Trees (78.1%), Random Forests (82.5%), and K-Nearest Neighbors (97.1%) in accuracy and MSE. The study aimed to identify the best adaptive credit card fraud detection solutions.

Husejinovic, A. [18] conducted research utilizing C4.5 decision trees, naive Bayes, and bagging ensembles to predict outcomes in both authentic and fraudulent credit card transactions. The performance of these algorithms was assessed based on precision, recall, and the area under the precision-recall curve (PRC). The dataset comprised credit card transactions made by European cardholders in September 2013, with 492 fraudulent transactions out of a total of 284,807. The results indicated that the Bagging ensemble technique, utilizing a C4.5 decision tree as the learner, achieved the highest Precision-Recall Curve (PRC) class 1 rate of 0.825. The C4.5 decision tree algorithm exhibited a fraud prediction accuracy of 92.74%. The PRC area rates for the zero class ranged from 0.999 to 1.000, showcasing the success of the machine learning techniques in differentiating the binary class 0 in the dataset. Class 1 precision-recall curve (PRC) values for this study were Naive Bayes classifier 0.080, C4.5 decision tree 0.745, and Bagging ensemble learner 0.825. While the C4.5 decision tree and bagging techniques effectively differentiated binary class 1, the Naive Bayes approach required revision. The best-performing C4.5 decision tree algorithm accurately identified all predicted fraudulent transactions at a rate of 92.74%. Confusion matrices summarized the algorithm performance, where Class 0 represented positive cases, and Class 1 represented negative cases. Naive Bayes demonstrated 99.9% accuracy, 97.8% recall, and a 1.000 PRC area in Class 0, while C4.5 and Bagging achieved 1.000 accuracy, recall, and PRC area. The PRC Area highlighted that bagging, utilizing a C4.5 decision tree as the learner, yielded the most favorable results, with distribution rates of 1.000 for Class 0 and 0.825 for Class 1. The accuracy rates for Class 0 and Class 1 in the C4.5 decision tree model were 1.000 and 0.927, respectively.

Trivedi, N.K. et al. [19] conducted a research evaluation of multiple machine learning algorithms, including support vector machine, decision tree, K-nearest neighbors, logistic regression, random forest, naive Bayes, and gradient boosting classifier. The assessment utilized metrics such as accuracy, precision, recall, F1-score, and false positive rate (FPR). The dataset comprised 284,807 transactions from European cardholders provided for machine learning analysis. The findings of the research highlighted that the random forest approach outperformed other algorithms across accuracy, precision, recall, and F1-score. The specific metrics for the random forest algorithm were as follows: accuracy - 94.9991%, precision - 95.9887%, recall - 95.1234%, F1 score - 95.1102%, and FPR - 3.9875%. The performance of the naive Bayes algorithm was not disclosed, but the model accurately classified 91.888% of cases, with 91.201% of positive forecasts being accurate. True positive predictions constituted 91.98% of all positive cases, resulting in an F1 score of 91.7748% and an FPR of 4.778. The classification model metrics for other algorithms were as follows: First model: 90.448% accuracy, 92.8956% precision, 93.112% recall, 92.112% F1-score, and 3.9785% FPR. Second model (SVM): 93.963% accuracy, 93.228% precision, 93.005% recall, 93.479% F1-score, and 3.889% FPR.Third model (KNN): 94.999% accuracy, 94.5891% precision, 92.008% recall, 91.003% F1-score, and 3.998% FPR. Decision trees: 90.998% accuracy, precision, recall, F1-score, and FPR. Fifth model (GBM): 94.001% accuracy, precision, recall, F1-score, and FPR. Consequently, the research suggests that the random forest method might be the preferred choice for achieving a balance between quality and comprehensiveness. Future iterations of this proposed technique could be tested using large real-time datasets and diverse machine learning methods.

Joshi A. et al. [20] conducted a research study to assess the performance of the local outlier factor, the isolation forest algorithm, and K-means clustering on skewed credit card fraud data. The evaluation metrics employed included the balanced classification rate, PR-AUC, Matthew's correlation coefficient, accuracy, specificity, and sensitivity. The dataset experiment comprised two parts. First, the dataset was divided into three ratios:

1. With 60 percent of the dataset allocated for training and 40 percent for testing, Isolation Forest achieved an accuracy of 99.7787%, while K-Means clustering yielded local outlier factor values of 99.6752% and 53.9978%.
2. In the scenario of a 70% training set and 30% testing set, Isolation Forest demonstrated 99.7799% accuracy, outperforming the local outlier factor with 99.6804% accuracy. K-means clustering achieved 53.8756% accuracy.
3. In the case of an 80% training set and 20% testing set, Isolation Forest reached 99.7928% accuracy, while the local outlier factor had 99.6804% accuracy. K-means clustering resulted in 53.904% accuracy.

Overall, isolated forests consistently outperformed the other two methods across different scenarios. The research also explored how categorization strategies for unbalanced datasets, including over- and under-sampling, might enhance algorithm performance. This phase delved into the hyperparameter configuration of the algorithm, aiming to find optimal settings for improved performance on class-imbalanced datasets. K-fold cross-validation was employed to evaluate machine learning models, with Isolation Forest consistently surpassing the local outlier factor and K-means clustering techniques. The study suggests that a future investigation could focus on exploring meta-classifiers and meta-learning methods specifically tailored for extremely skewed credit card fraud data. Additionally, exploring ensemble approaches and modular algorithm combinations within a big data-driven ecosystem could facilitate further system testing with additional datasets. In Table I below, the results of the machine learning techniques utilized in the previous studies have been presented:

TABLE I
RESULTS OF THE EMPLOYED TECHNIQUES IN PREVIOUS STUDIES

REF.	TECHNIQUES	ACCURACY	RECALL	PRECISION	MCC	F1-SCORE
Garg V., et al. [3]	ET	99.96%	79%	94.6%	86%	86%
	RF	99.95%	78%	94%	85%	85%
	IDA	99.93%	73%	85%	79%	78%
	ADA	99.92%	70%	82%	75%	75%
	LR	99.91%	60%	81%	69%	69%
	DT	99.91%	75%	74%	74%	74%
	RIDGE	99.89%	42%	82%	58%	55%
	GBC	99.89%	41%	77%	54%	50%
	KNN	99.84%	5%	81%	21%	10%
	SVM	99.82%	0%	0%	-.0002%	0%
	Lightbm	99.51%	53%	21%	33%	29%
	NB	99.26%	62%	13%	29%	22%
	QDA	97.58%	86%	5%	22%	10%
Nadim, A.H., et al. [8]	RF	98.6%	___	0.997	___	___
	XGB	98.4%	___	0.994	___	___
	LR	97.7%	___	0.996	___	___
	SVM	97.5%	___	0.996	___	___
	LDA	97.4%	___	0.995	___	___
	KNN	96,9%	___	0.991	___	___
	CART	58.6%	___	0.94	___	___
Uchhana, N. et al [9]	RF	___	0.90	___	0.848	1.00
	KNN	___	0.8	___	0.793	1.00
	LR	___	0.8	___	0.761	1.00
	NB	___	0.9	___	0.761	0.98
	SVM	___	0.92	___	0.558	1.00
Chakshu. V, Chand.S [10]	SVM Kernel	97.2%	___	30%	___	___
Sadineni, P.K.[11]	ANN	99.92%	___	99.57%	___	___
	RF	99.21%	___	92.34%	___	___
	DT	98.47%	___	84.98%	___	___
	LR	95.55%	___	83.76%	___	___
	SVM	95.16%	___	88.42%	___	___
Ignatius. J, et al. [12]	Isolation forest	97%	1.00	1.00	___	1.00
	Local Outlier Factor	76%	1.00	1.00	___	1.00

Anand, H., et al.[13]	Isolation forest	99.75%	___	___	___	___
	Local Outlier	99.65%	___	___	___	___
	SVM	70%	___	___	___	___
Rout, M. [14]			Individual models			
	RF	99.96%,	0.83	0.95	0.8900	0.89
	XGBoost	99.95%	0.83	0.92	0.8726	0.87
	LR	99.93%	0.68	0.91	0.7876	0.78
	NB	99.92%	0.66	0.86	0.7497	0.74
			Soft voting			
	RF+DT	99.94%	0.80	0.88	0.8416	0.84
	NB+DT	99.92%	0.80	0.76	0.7802	0.78
	LR+DT	99.91%	0.80	0.74	0.7729	0.77
	XGBoost+DT	99.91%	0.80	0.72	0.7592	0.76
			AdaBOOST			
	RF	99.96%	0.84	0.95	0.8975	0.90
	XGBoost	99.95%	0.83	0.92	0.8764	0.88
	LR	99.93%	0.69	0.90	0.7884	0.78
	NB	99.92%	0.66	0.86	0.7497	0.74
V Kumar K S, et al. [15]	ANN	98.69%	98.98	___	___	___
	LR	94.84%	92.00	___	___	___
	DT	92.88%	86.34	___	___	___
	NB	91.62%	84.82	___	___	___
Manohar s, et al. [16]	SVM	99.8%	___	___	___	___
	RF	99.7%	___	___	___	___
	DT	99.7%	___	___	___	___
Sadgali, I., et al. [17]	SVM	99.7%	___	___	___	___
	KNN	97.1%	___	___	___	___
	RF	82,5%	___	___	___	___
	DT	78,9%	___	___	___	___
Husejinovic, A. [18]	C4.5D	___	1.000	___	___	___
	NB	___	0.978	___	___	___
	Bagging	___	1.000	___	___	___
Trivedi, N.K., et al. [19]	RF	94.99%	95.1234%	95.9887%	___	95.11
	KNN	94.99%	92.008%	94.5891%	___	91
	GBM	94%	93.001%	93.998%	___	93.99
	SVM	93.96%	93.005%	93.228%	___	93.47
	NB	91.88%	91.989%	91.201%	___	91.77
	DT	90.99%	91.996%	90.998%	___	92.77
	LR	90.44%	93.112%	92.8956%	___	91.11
Joshi, A., [20]	Isolation forest	99.7787%	___	___	___	___
		99.7799%	___	___	___	___
		99.7928%	___	___	___	___
	Local Outlier	99.6752%	___	___	___	___
		99.6804%	___	___	___	___
		99.6804%	___	___	___	___
	K-Means	53.9978%	___	___	___	___
		53.8756%	___	___	___	___
		53.9043%	___	___	___	___

Several of the prior studies discussed in Table I above overlooked the management of imbalanced datasets, despite the imperative to address such data imbalances for the generation of truthful and accurate results. Consequently, in addressing the imbalanced dataset, this proposed model assessed the outcomes of three sampling strategies:

Random over-sampler, cluster centroid-based majority under-sampling techniques (CCMUT), and SMOTE. Additionally, the significance of using the F2-score was disregarded, despite its relevance. The f2-score becomes pertinent when the cost of a false negative result is higher. This is due to the F2-score's emphasis on recall over precision, contrasting with the F1-score, which assigns equal value to both recall and precision.

TABLE II
THE ORIGINS OF DATA AND THE SCALE OF DATA EMPLOYED IN PREVIOUS STUDIES

REF.	DATASET SOURCE	DATASET SIZE
[3]	European cardholders September 2013European Cardholder	284,807 transactions
[9]	European cardholders September 2013European Cardholder	284,807 transactions
[10]	Not mentioned	Not mentioned
[11]	Kaggle	150000
[12]	European cardholders September 2013European Cardholder	Transactions
[13]	Kaggle	284807 transactions
[14]	European cardholders September 2013European Cardholder	Not mentioned
[8]	European cardholders September 2013European Cardholder	284,807 transactions
[15]	European cardholders September 2013European Cardholder	284,807 transactions
[16]	European cardholders September 2013European Cardholder	284,807 transactions
[17]	Not mentioned	284,807 transactions
[18]	European cardholders September 2013European Cardholder	60.000 transactions in across 12 attributes.
[19]	European cardholders September 2013European Cardholder	284,807 transactions
[20]	European cardholders September 2013European Cardholder	284,807 transactions

III. FORMULATION OF PROBLEMS

The dataset utilized for training the proposed models comprises 284,807 entries conducted by European cardholders over a span of two days in September 2013. Across the entire dataset, there were 492 fraudulent transactions and 284,315 legitimate transactions, with only 0.172% of the total transactions identified as fraudulent.

A. Imbalanced Dataset

The creation of a training dataset is essential for assisting algorithms in discerning specific features. However, using the initial informative dataset for this purpose is deemed ineffective for apparent reasons. The global occurrence of fraudulent transactions is identified to be less than 0.1% of the total transactions, resulting in highly imbalanced classes. Without proper acknowledgment, a machine learning algorithm focused on accurately classifying legitimate transactions could display outstanding performance, achieving a 99% accuracy rate but neglecting misclassifications in the minority class. Fraudulent activities tend to evolve over time to avoid detection. Therefore, it is crucial that the predictive model for credit card fraud detection is not static. It should not be constructed once and left unchanged without updates.

To address this, cluster centroid-based majority under-sampling techniques, random-over sampling, and SMOTE are employed to generate a training dataset. This approach ensures an even distribution of classes, fostering a balanced representation. Such a balanced class distribution prompts algorithms to effectively detect fraudulent transactions, aligning with the overarching goal of the procedure.

Detecting credit card fraud is commonly framed as a binary classification problem with imbalanced data, where fraud instances constitute a small fraction, often less than 0.1%, of the total dataset [1]. The challenge lies in identifying the minority class within a large volume of data. This situation, known as "data imbalance “, in machine learning, occurs when the distribution of classes or labels in a dataset is highly skewed, posing a significant problem in classification tasks [17]. Imbalanced datasets can hinder classifier performance, causing the minority class to be misclassified as noise or outliers.

To tackle this issue, our model employed three methods to address the uneven distribution: cluster centroid-based majority under-sampling technique (CCMUT), synthetic minority over-sampling technique (SMOTE), and random over-sampler. However, SMOTE comes with two limitations. Firstly, replicating minority-class instances in the dataset increases the risk of overfitting. Moreover, the learning process is prolonged, especially when the original dataset is both extensive and imbalanced compared to our dataset. In situations where data availability is constrained, this approach proves effective. Random under-sampling, a technique that randomly removes instances from a dataset, is one strategy employed. However, this introduces a potential risk of losing valuable examples.

To address this concern, a proposed solution involves integrating unsupervised learning and supervised learning by employing a clustering tool (CCMUT) for sampling. This hybrid approach aims to reduce the likelihood of discarding relevant data from the majority class.

B. The Performance Metrics

The performance of the proposed model undergoes evaluation through nine assessment techniques: Extra Tree, Random Forest, AdaBoost, Decision Tree, Gradient Boosting, KNN, MLP, Naive Bayes, and Logistic Regression. These techniques utilize diverse performance metrics such as accuracy, precision, recall, F1-score, and F2-score to gauge the effectiveness of the model. The descriptions of these metrics are as follows:

Accuracy

It calculates the percentage of true positives and true negatives among all cases, indicating the resemblance between the matrix data and the actual data. A higher score in this context signifies greater similarity [21].

$ACC= (TP+TN\ (TP+TN) + (FP+TN))$ (1)

Precision (P)

This metric compares the number of true positives to the total number of positives predicted by the model. It computes the proportion of accurate positive predictions [25].

P= (PT/PT+FP) (2)

Recall

The concept of recall refers to the proportion of correctly identified positive items out of the total number of items classified as being positive. The recall is calculated as follows [22]:

Recall =TP/ (TP+FN) (3)

The F1-Score

The F1 score is a metric that calculates the weighted mean of precision and recall. It considers both false negatives and false positives, making it particularly useful.

F1 Score =2*(Precision *Recall)/(Precision +Recall) (4)

The F2 score

The F2 score can be defined as the weighted harmonic average of precision and recall, with consideration for a specific threshold value. The F2 score places greater emphasis on recall compared to precision, in contrast to the F1 score, which assigns equal importance to recall and precision. The F2 score is calculated as follows [24]:

F2-Score = ((1+22)* Precision * Recall)/ (22*Precision + Recall) (5)

IV. PROPOSED FRAMEWORK

The efficacy of nine machine learning techniques random forest, gradient boost classifier, MLP classifier, extra tree classifier, naive bayes, Ada-boost classifier, k-nearest Classifier, Decision Tree, and Gradient Boost Classifier was assessed within the framework proposed. Performance metrics, including recall, precision, F1-score, F2-score, and accuracy, were employed to gauge the effectiveness of these techniques. The illustration below outlines the process of identifying credit card fraud using machine learning (ML) techniques. This graphic delineates the crucial phases of the modified framework specifically designed for the purposes of this study. Figure 1 illustrates the diverse processes engaged in identifying credit card (CC) fraud through the application of machine learning techniques. The following steps were applied.

This section provides a comprehensive analysis of the results achieved. It details the experimental proposed model employed to implement supervised machine learning algorithms on credit card data and evaluate their performance. The implementation of the solution operates as follows:

Step 1: Importing all the necessary libraries: Exploratory data processing and charting make use of NumPy, Pandas, OS, and Seaborn. NumPy serves as the core Python module for scientific computing.

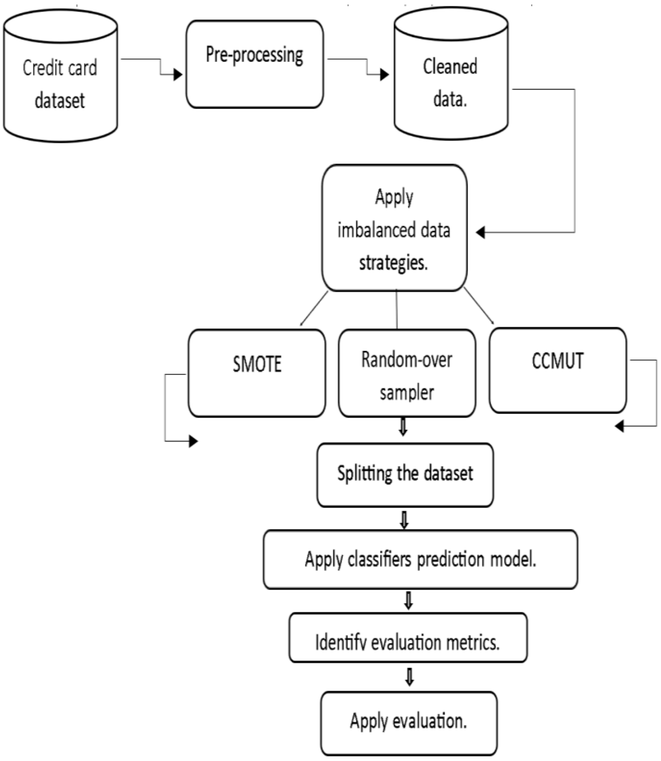

Fig. 1. The proposed framework for credit card fraud detection

V. METHEDOLOGY

Pandas, on the other hand, is an open-source data analysis and manipulation toolkit known for its speed, efficiency, scalability, and user-friendly interface. The OS module in Python provides functions for tasks such as creating and deleting directories (folders), accessing their contents, modifying, identifying the current directory, and more. Additionally, Seaborn, a Python toolkit, is employed for crafting static, animated, and immersive graphics.

Step 2: Importing data into Data Frame and required libraries.

Step 3: Balancing the dataset using Random Over Sampler, Under-Sampling (CCMUT), and SMOTE.

Step 4: Partition the dataset into training and testing subsets, followed by the model input.

Step 5: Applying all models with cluster centroid based-majority under-sampling technique.

Step 6: Applying all models with SMOTE.

Step 7: Applying all models with random over-sampler.

The performances of certain machine learning algorithms have been examined, and they are categorized into the following: Classification and Ensemble Learning algorithms.

The classification algorithms are as follows:

A. Naïve bayes

Naive Bayes, a supervised machine learning approach, is employed to address classification issues. The method is effective with limited training data, estimating necessary parameters for classification. It utilizes Bayes' theorem to calculate the probability of the true class for classification [25].

B. Decision tree

The Decision Tree serves as a machine learning technique applicable for both classification and regression problem-solving. In this structure, each internal node signifies an attribute test, branches indicate test outcomes, and leaf nodes represent classes or results. The reliance of decision trees on input data stems from their algorithmic complexity and sequential nature, where even minor changes can influence the tree's structure [25].

C. Logistic regression

Logistic regression, an algorithm designed for classification, is employed to categorize data into discrete outcomes [19]. This method is utilized in constructing a regression model when the dependent variable is categorical. Originating in 1958 by David Cox [20], logistic regression encompasses three types: (1) binary, suitable for a binary response variable; (2) multinomial, applicable when the dependent variable has more than two non-ordered categories; and (3) ordinal, used for ordered categories [25].

D. K-nearest neighbor (KNN)

K-nearest neighbor (KNN) is a classification algorithm rooted in analogy learning. In the presence of a new, unfamiliar sample, the classifier explores the pattern space to identify the K-nearest neighbors in close proximity to the new sample. It then assigns the class of the new pattern based on the closest pattern class, employing a supervised approach. The algorithm relies on distance as a metric to determine closeness [25].

Ensemble Learning Group:

A. Random Forest (RF)

Is an ensemble learning classifier comprising numerous decision trees. It determines the class by aggregating the outputs of each decision tree within the ensemble [25].

B. The Extra Trees algorithm

The choice of an extra-tree classifier is motivated by its clear interpretation, straightforward properties, and its ease of conversion to "if–then" rules. The selection of the extra-tree method is particularly advantageous for its randomizing property when handling numerical inputs. This feature proves highly beneficial in scenarios with a large number of numerical features, often resulting in improved accuracy in such situations [26].

C. MLP

The units are organized into a series of layers, with each layer containing a specific number of identical units. Every unit in one layer is connected to each unit in the subsequent layer, making the network fully connected. The initial layer is the input layer, and its units take on the values of the input features. The final layer is the output layer, housing one unit for each output value of the network, whether it be a single unit for regression or binary classification or K units for K-class classification. The layers situated between these are referred to as hidden layers, as their specific computations are not predetermined and are discovered during the learning process [27].

D. Gradient Boosting

The Gradient Boosting Machines algorithm aims to optimize a cost function by iteratively selecting a function within the function space. This chosen function consistently moves in the negative gradient direction to enhance the overall optimization process [25].

E. Ada-boost

AdaBoost's objective is to enhance classification performance by combining various weak learners or classifiers (hi(x)), where hi(x) denotes an individual classifier. Each weak learner is trained using a basic set of training samples, each assigned a weight. These sample weights are adjusted iteratively. AdaBoost sequentially trains weak learners, assigning a weight to each, reflecting the robustness of the respective weak learner [28].

VI. IMPLIMENTATION

The implementation stages will be explained as follows:

A. Dataset

The dataset encompasses 284,807 instances of transactions conducted by European cardholders over a span of two days in September 2013. Across the entire dataset, there were 492 fraudulent transactions and 284,315 legitimate transactions, with only 0.172% of the total transactions identified as fraudulent. The dataset includes thirty-one latent attributes denoted as V1 to V28, with their values kept confidential. A transaction labeled with 0 is considered legal, while a value of 1 signifies a transaction classified as fraudulent Principal Component Analysis (PCA) generated a set of thirty-one features, including "time," "amount," and "class." The additional twenty-eight columns in the dataset are labeled V1 through V28, representing corresponding features. In PCA, only the variables "time" and "amount" have been retained to ensure user identity and personal information confidentiality. The 'Time' feature indicates the temporal duration in seconds between each transaction and the initial one in the dataset. The "Amount" feature represents the transaction amount, suitable for scenarios dependent on examples and sensitive to price [8].

B. Data pre-processing

Data preprocessing has been executed on the dataset. Data preprocessing involves the preparation of raw data for analysis through various techniques like cleaning, integration, reduction, and transformation. This essential procedure ensures that the data is appropriately prepared for subsequent analysis. The primary objective of data preprocessing is to enhance the quality and suitability of the data for a specific task.

C. Apply imbalanced strategies

This proposed framework addresses the issue of imbalanced datasets through the utilization of three strategies: random over-sampler, cluster centroid-based majority under-sampling technique (CCMUT), and SMOTE. As discussed earlier.

The proposed model utilizes a banking credit card fraud dataset obtained from Kaggle. This dataset consists of credit card transactions carried out by European cardholders in September 2013, spanning a period of two days. The dataset contains information on a total of 284,807 transactions, with

492 transactions identified as fraudulent. Notably, the dataset demonstrates a substantial class imbalance, where the positive class, indicating fraudulent transactions, constitutes only 0.172% of the total transactions.

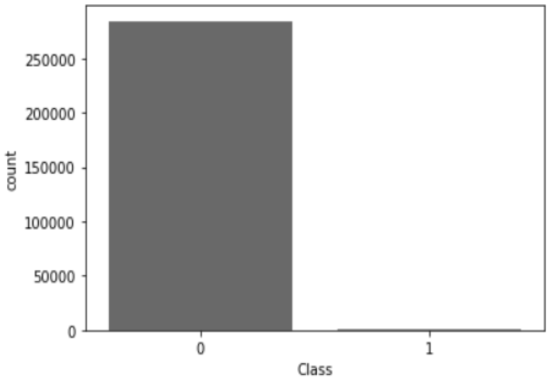

Fig. 2. Imbalanced Dataset

Figure 2 illustrates a significant imbalance, with the positive class constituting only 0.172% of all transactions, amounting to 492 fraudulent transactions. Consequently, the proposed model employs three sampling techniques—random over-sampler, cluster centroid-based majority under-sampling technique (CCMUT), and synthetic minority over-sampling technique (SMOTE)—to address the imbalanced dataset. Figures 3, 4, and 5 depict the approach to handling the imbalanced dataset using cluster centroid-based under-sampling technique (CCMUT), random over-sampling method, and synthetic minority over-sampling majority (SMOTE), respectively).

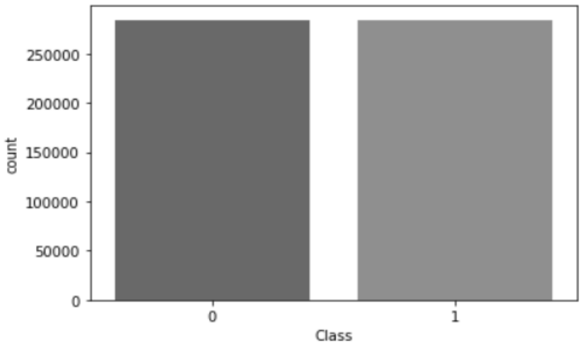

Fig. 3. Balanced dataset using SMOTE

Figure 3 illustrates the application of the Synthetic Minority Over-sampling Technique (SMOTE) to attain dataset balance. Renowned for its capability to generate synthetic instances of fraudulent transactions, SMOTE helps balance the class ratio by creating synthetic transactions through this chosen method [13].

Figure 4 illustrates the application of the random over-sampler technique, employed to address the challenge of class imbalance in the dataset. The recommended approach involves augmenting the total count of the minority class by randomly replicating existing instances [22].

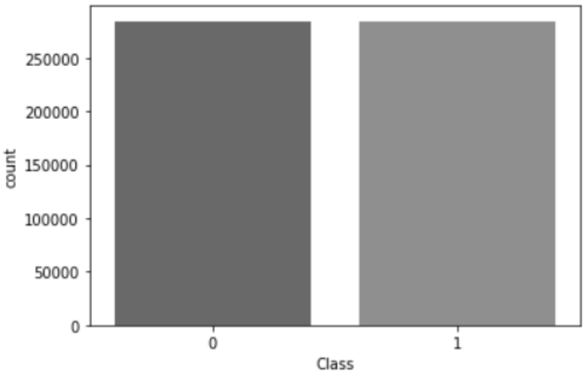

Fig. 4. Balanced dataset using random over-sampler

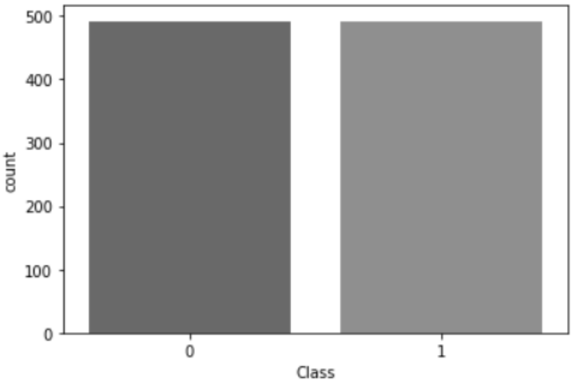

Fig. 5. Balanced dataset cluster centroid-based majority under-sampling technique

Figure 5 demonstrates the attainment of dataset balance through the under-sampling strategy, specifically the Cluster Centroids method. This under-sampling approach can be implemented alongside random under-sampling, where transactions are randomly removed from the training set of the majority class [23]. However, the random removal of transactions poses a significant risk of eliminating crucial or necessary instances, potentially reducing the effectiveness of the detection process. To address this issue, a clustering-based framework is introduced, clustering instances with comparable properties. The primary objective of this strategy is to eliminate dataset instances without doing so randomly, avoiding the potential loss of valuable examples that may contain information crucial for accurate conclusions and predictions [24].

D.Splitting the dataset

In the proposed model, the dataset has undergone partitioning into two distinct subsets: a training dataset and a testing dataset. This division was accomplished through two different ratios:

1.Utilizing both a random over-sampler and cluster centroid-based majority under-sampling techniques, the dataset was segmented into an 80% training set and a 20% testing set.

2. Employing the Synthetic Minority Over-Sampling Technique (SMOTE) to address class imbalance, the dataset was divided into a 75% training set and a 25% testing set.

1. Apply classifier prediction model

In this phase, nine supervised machine learning techniques were utilized, including Random Forest, MLP classifier, Naïve Bayes, Decision Tree, Gradient Boosting, AdaBoost, KNN, Logistic Regression, and Extra Trees classifier, among others. The effectiveness of these nine machine learning techniques was evaluated using a diverse set of performance metrics, including recall, accuracy, precision, F1-score, F2-score, and recall.

1. Identify evaluation metrics

The nine machine learning techniques employ various performance metrics, including accuracy, precision, recall, F1-score, and F2-score, to evaluate the model's effectiveness.

VII. EXPERIMENTAL RESULTS

Following the resolution of the imbalanced dataset issue, the dataset was divided into two subsets: a training dataset and a testing dataset. Two ratios were employed for this partitioning, (80:20) using a cluster centroid-based majority under-sampling technique and a random over-sampler, and (75:25) using the Synthetic Minority Over-Sampling Technique (SMOTE). Nine machine learning techniques were then applied. To assess the proposed model, the effectiveness of these techniques was evaluated using various performance metrics, including F2 score, F1 score, accuracy, recall, and precision. Tables III, IV, and V present the outcomes of the nine machine learning techniques, comparing their performance under the cluster centroid-based majority under-sampling technique (CCMUT), SMOTE, and random over-sampler. The accuracy results of the nine classifiers, utilizing the cluster centroid-based majority under-sampling method (CCMUT), SMOTE, and random over-sampler, were compared in Figures 6, 7, and 8.

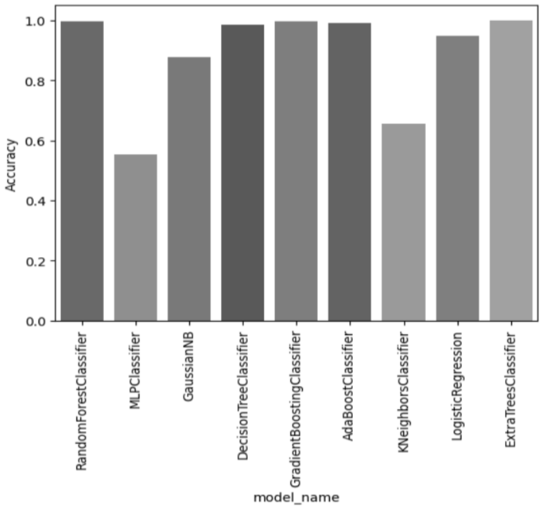

Fig. 6. Comparison of accuracy results of the nine classifiers using cluster centroid based-majority technique (CCMUT)

TABLE III
OBSERVATION RESULTS USING CLUSTER CENTROID

TECHNIQUES	ACCURACY	PRECISION	RECALL	F1-Score	F2-Score
Random Forest	99.49%	100	98.98	99.49	99.18
MLP CLASSIFIER	55.33	100	10.20	18.52	12.43
Naïve Bayes	87.82	100	75.51	86.05	79.39
Decision Tree	98.48	97.98	98.98	98.48	98.77
Gradient Boosting	99.49	98.99	100	99.49	99.79
Ada-Boost	99.49	100	98.98	99.49	99.59
KNN	65.48	61.54	81.63	70.18	76.62
Logistic Regression	94.92%	97.83	91.84	94.74	92.97
EXTRA Trees	100%	100%	100%	100%	100%

TABLE IV
RESULTS OF TECHNIQUES WITH SMOTE

TECHNIQUES	ACCURACY	PRECISION	RECALL	F1_score	F2-Score
Random Forest	99.99	99.98	100	99.99	99.99
MLP Classifier	98.17	99.21	97.11	98.15	98.52
Naïve bayes	86.87	98.97	74.54	85.03	78.40
Decision tree	99.83	99.74	99.92	99.83	99.88
Gradient boosting	99.87	99.82	99.93	99.87	99.90
Ad-boost	98.65	99.15	98.14	98.64	98.34
KNN	96.09	94.78	97.58	96.16	97
Logistic regression	97.37	98.19	96.52	97.35	96.85
EXTRA trees	99.99	99.98	100	99.99	99.99

TABLE V
RESULTS OF TECHNIQUES WITH RANDOM OVER-SAMPLER

TECHNIQUES	ACCURACY	PRECISION	RECALL	F1-Score	F2-Score
Random Forest	99.995	99.991	100	99.995	99.99
MLP CLASSIFIER	95	96.63	93.27	94.92	93.92
Naïve bayes	86.68	98.74	74.35	84.83	78.21
Decision tree	99.98	99.95	100	99.98	99.99
Gradient boosting	99.98	99.96	100	99.98	99.99
Ada-boost	98.61	98.78	98.44	98.61	98.50
KNN	99.89	99.79	100	99.90	99.95
Logistic regression	91.92	95.29	88.24	91.63	89.56
EXTRA trees	99.996	99.992	100	99.996	99.99

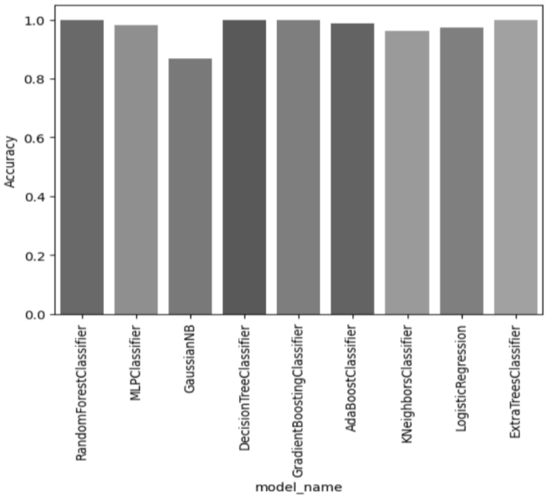

Fig. 7. Comparison of accuracy results of the nine machine learning classifiers using Smote

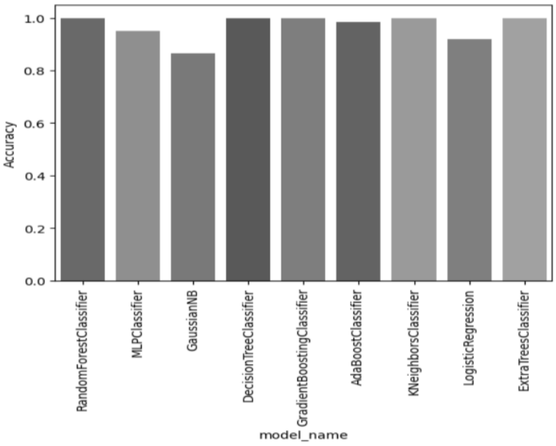

Fig. 8. Comparison of accuracy results of the nine machine learning classifiers using the random over-sampler technique

VIII. RESULTS AND DISCUSSION

The experimental findings presented in Tables III, IV, and V provide a comprehensive evaluation of diverse metrics applied to the credit card fraud dataset, employing nine supervised machine learning techniques. Specifically, in Table III, the adoption of the cluster centroid-based majority under-sampling approach (CCMUT) as a sampling technique allowed for a thorough assessment of the performance of these nine methods.

Notably, the results illuminate that among the various algorithms considered, the Extra Trees technique emerges as the standout performer across critical metrics, including accuracy, recall, precision, F1-score, and F2-score. This pronounced superiority of the Extra Trees algorithm underscores its efficacy in addressing the challenges posed by the imbalanced nature of the credit card fraud dataset. The significance of these findings lies in the identification of a robust and high-performing method, which can be pivotal in enhancing the reliability and accuracy of credit card fraud detection systems.

By comparing the results of tables III, IV, and V, it is observed that:

A. Random Forest

Cluster Centroid: Achieved an accuracy of 99.49%, perfect precision (100%), and high recall (98.98%).

SMOTE: Improved accuracy to 99.99% with near-perfect precision (99.98%) and perfect recall (100%).

Random Over-Sampler: Maintained near-perfect scores across all metrics (99.995% accuracy, precision, recall, F1-score, and F2-score). All three techniques show excellent performance, with Random Over-Sampler slightly outperforming the others.

B. MLP Classifier

Cluster Centroid: Demonstrated a lower accuracy of 55.33%, highlighting challenges in overall performance.

SMOTE: Improved accuracy to 98.17% with well-balanced precision (99.21%) and recall (97.11%).

Random Over-Sampler: Achieved a high accuracy of 95% with balanced precision (96.63%) and recall (93.27%). SMOTE performed better in accuracy and precision, while Random Over-Sampler showed a balanced improvement.

C. Naïve Bayes

Cluster Centroid: Achieved an accuracy of 87.82% with perfect precision (100%) and good recall (75.51%).

SMOTE: Displayed similar accuracy (86.87%) with excellent precision (98.97%) and a slightly higher recall (74.54%).

Random Over-Sampler: Maintained accuracy at 86.68% with excellent precision (98.74%) and a similar recall (74.35%). SMOTE and Random Over-Sampler show comparable performance, with a slight edge to SMOTE.

D. Decision Tree

Cluster Centroid: Achieved a high accuracy of 98.48% with balanced precision (97.98%) and recall (98.98%).

SMOTE: Maintained near-perfect scores across all metrics (99.83% accuracy, precision, recall, F1-score, and F2-score).

Random Over-Sampler: Also maintained near-perfect scores (99.98% accuracy, precision, recall, F1-score, and F2-score). SMOTE and Random Over-Sampler demonstrate similar excellent performance.

E. Gradient Boosting

Cluster Centroid: Achieved a high accuracy of 99.49% with high precision (98.99%) and perfect recall (100%).

SMOTE: Maintained excellent performance with high accuracy (99.87%) and near-perfect precision (99.82%) and recall (99.93%).

Random Over-Sampler: Similarly, maintained excellent performance with high accuracy (99.98%) and near-perfect precision (99.96%) and recall (100%). SMOTE and Random Over-Sampler exhibit comparable outstanding performance.

F. Ada-Boost

Cluster Centroid: Achieved high accuracy (99.49%) with perfect precision (100%) and high recall (98.98%).

SMOTE: Maintained high accuracy (98.65%) with well-balanced precision (99.15%) and recall (98.14%).

Random Over-Sampler: Also maintained high accuracy (98.61%) with well-balanced precision (98.78%) and recall (98.44%). Cluster Centroid and SMOTE show similar performance, while Random Over-Sampler is slightly behind.

G.KNN

Cluster Centroid: Showed a moderate accuracy of 65.48%, indicating potential challenges in overall performance.

SMOTE: Improved accuracy to 96.09% with balanced precision (94.78%) and recall (97.58%).

Random Over-Sampler: Achieved high accuracy (99.89%) with near-perfect precision (99.79%) and perfect recall (100%). Random Over-Sampler outperforms both Cluster Centroid and SMOTE significantly.

H.Logistic Regression

Cluster Centroid: Achieved a high accuracy of 94.92% with well-balanced precision (97.83%) and recall (91.84%).

SMOTE: Maintained high accuracy (97.37%) with balanced precision (98.19%) and recall (96.52%).

Random Over-Sampler: Achieved moderate accuracy (91.92%) with balanced precision (95.29%) and recall (88.24%). SMOTE outperforms the other two techniques in accuracy and precision.

I. EXTRA Trees

Cluster Centroid: Achieved perfect scores across all metrics (100% accuracy, precision, recall, F1-score, and F2-score).

SMOTE: Maintained near-perfect scores (99.99% accuracy, precision, recall, F1-score, and F2-score).

Random Over-Sampler: Also maintained near-perfect scores (99.996% accuracy, precision, recall, F1-score, and F2-score). All three techniques demonstrate exceptional and consistent performance.

In summary, each sampling technique (Cluster Centroid, SMOTE, and Random Over-Sampler) exhibits strengths and weaknesses across different machine learning techniques. The choice of the sampling technique should be made based on the specific requirements of the problem and the desired trade-offs between different evaluation metrics. Random Over-Sampler shows remarkable performance across various techniques, while SMOTE and Cluster Centroid demonstrate competitive results depending on the algorithm employed.

Random Forest showcased exceptional performance across all sampling methods, with Random Over-Sampler slightly outperforming others, maintaining near-perfect scores. The MLP Classifier faced challenges in overall performance, particularly under Cluster Centroid, but exhibited improvement with SMOTE and Random Over-Sampler. Naïve Bayes demonstrated comparable performance between SMOTE and Random Over-Sampler, with a slight advantage to SMOTE. Decision Tree, Gradient Boosting, and Ada-Boost consistently achieved high accuracy and precision, with slight variations among sampling methods. KNN exhibited significant improvement under Random Over-Sampler, outperforming Cluster Centroid and SMOTE by a considerable margin. Logistic Regression demonstrated balanced performance, with SMOTE exhibiting superior accuracy and precision.

Notably, Extra Trees consistently outperformed other techniques across all sampling methods, achieving perfect or near-perfect scores. These comprehensive results offer valuable insights into the effectiveness of different machine learning algorithms and sampling techniques for credit card fraud detection, aiding in informed decision-making for future model selections and implementations.

In the comprehensive comparison across all nine machine learning classifiers utilizing distinct sampling techniques, it becomes apparent that the random over-sampler method consistently attains the majority of expected outcomes. Following closely are the results from SMOTE, while the cluster centroid-based majority technique (CCMUT) exhibits commendable performance but lags slightly behind in comparison.

These results underscore the effectiveness of Extra Trees and the strategic choice of sampling methods, with the random over-sampler technique standing out as a particularly robust approach in the context of credit card fraud detection.

Upon meticulous examination of the results presented in Tables I, III, IV, and V, a rigorous comparative analysis was undertaken between the performance of machine learning techniques in the proposed model and the identical techniques utilized in prior studies, In comparing the outcomes of prior studies with our proposed model, it is important to note that while both utilized six common machine learning techniques (Random Forest, Decision Tree, Gradient Boosting, AdaBoost, KNN, and Logistic Regression), The shared techniques across both proposed model demonstrated superior performance in our proposed model, showcasing advancements in credit card fraud detection. The proposed model surpassed prior studies in performance metrics. The incorporation of the extra tree techniques in the proposed model, which yielded excellent results not explored in previous studies, represents a notable enhancement. This expansion in methodology contributes to the heightened efficacy of the proposed model, providing a more comprehensive and nuanced approach to credit card fraud detection. Thus, this comparative study underscores the innovation and scientific rigor of the proposed model, establishing it as an improvement in the field of machine learning for credit card fraud detection.

IX. CONCLUSION

This study undertakes a comparative analysis of various machine learning models for the detection of fraudulent transactions, including the Extra Trees Classifier, Random Forest, Logistic Regression, Naive Bayes, Gradient Boosting, MLP Classifier, Decision Tree, Ada-Boost, and KNN. The experiment comprises three stages: initial data preparation with data pre-processing techniques, addressing imbalanced datasets using Cluster Centroid-based Majority Under-Sampling Technique (CCMUT), Synthetic Minority Over-Sampling Technique (SMOTE), and Random Over-Sampler, and the application of supervised machine learning algorithms. Evaluation metrics such as accuracy, precision, recall, F1 score, and F2 score are employed, revealing that Extra Tree consistently outperforms other models in credit card fraud detection.

The study advocates for the integration of deep learning methodologies in future research endeavors, highlighting their superior performance in credit card fraud detection compared to traditional machine learning models. This suggests the practical applicability of deep learning methodologies in real-world contexts.

The use of independent under-sampling and over-sampling techniques is acknowledged to have limitations, impacting the efficacy of the detection system. Oversampling may lead to overfitting and overlapping, capturing noise and closely fitting the training data. In contrast, under-sampling may eliminate crucial information, blending valuable data with irrelevant noise and causing confusion. The study recommends the incorporation of the SMOTE technique into hybrid methods, suggesting combinations like SMOTE with ENN and SMOTE with Tomek. These hybrid approaches aim to achieve a balanced dataset, enhancing the overall effectiveness of the fraud detection system.

REFERENCES

[1] Z. Meng, Y., Xie, and J., Sun, “Detecting credit card fraud by generative adversarial networks and multi-head attention neural networks,” *IAENG International Journal of Computer Science*, vol. 50, no. 2, pp. 381-387, 2023.

[2] N. Elhusseny, S. Ouf, and A. Idrees, “Credit card fraud detection using machine learning techniques,” *Future Computing and Informatics Journal*, vol. 7, no.1, pp. 13-21, 2022.

[3] V. Garg, S. Chaudhary, and A. Mishra, “Analyzing auto ml model for credit card fraud detection,” *International Journal of Innovative Research in Computer Science & Technology (IJIRCST),* vol. 9, no. 3, pp. 31-36, 2021.

[4] T. Dang, T. Tran, L. Tuan, and M. Tiep, “Machine learning based on resampling approaches and deep reinforcement learning for credit card fraud detection systems,” *Applied Sciences*, vol. 11, no. 21, pp. 1-32, 2021.

[5] M. Sharaf, S. Ouf, and A. Idrees, “Risk assessment approaches in banking sector-A survey,” *Future Computing and Informatics Journal,* vol. 8, no. 1, pp. 32-39, 2023.

[6] N. Bayomy, A. Khedr, and L. Abd-Elmegid, “Adaptive model to support business process reengineering,” *PeerJ Computer Science*, vol. 7, no. e505, pp. 1-25, 2020.

[7] A. Khedr, A. Idrees, and R. Salem, “Enhancing the e-learning system based on a novel tasks’ classification load-balancing algorithm,” *PeerJ Computer Science*, vol. 7, no. e669, pp. 1-28, 2021.

[8] A. Nadim, I. Sayem, A. Mutsuddy, and M. Chowdhury, “Analysis of machine learning techniques for credit card fraud detection,” *2019 International Conference on Machine Learning and Data Engineering (iCMLDE).* pp. 42-47, 2019.

[9] N. Uchhana, R. Ranjan, S. Sharma, D. Agrawal, and A. Punde, “Literature review of different machine learning algorithms for credit card fraud detection,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 10, no. 6, pp. 101-108, 2021.

[10] V. Chakshu, and S. Chand, “Credit card fraud detection and analysis using machine learning algorithms,” *International journal of innovations in engineering research and technology*, vol. 8, no. 5, pp. 121-127, 2021.

[11] P. Sadineni, “Detection of Fraudulent Transactions in Credit Card using Machine Learning Algorithms,” *in 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC),* IEEE, pp. 659 – 660, 2020.

[12] J. Ignatius, Y. Kulkarni, S. Bari, and D. Naglot, “Comparative Analysis of Machine Learning Algorithms for Credit Card Fraud Detection,” *International Journal of Computer Sciences and Engineering*, vol. 8, no. 6, pp. 6-9, 2020.

[13] V. Dornadula, S Geetha, “Credit Card Fraud Detection Using Machine Learning Algorithms,” *Procedia Computer Science,* vol. 165, pp. 631-641, 2019.

[14] M. Rout, “Analysis and comparison of credit card fraud detection using machine learning,” *Advances in electronics, communication, and computing,* Springer, pp. 33-40, 2021.

[15] V. Kumar, A. Shankar, and K. Pratibha, “Credit card fraud detection using machine learning Algorithms,” *International Journal of Engineering Research & Technology*, vol. 9, no. 7, pp. 1526-1530, 2020.

[16] S. Manohar, A. Bedi, S. Kumar, and K. Singh, “Fraud detection in credit card using machine learning techniques,” *International Research Journal of Engineering and Technology*, vol. 7, no. 4, pp. 1786-1791, 2020.

[17] I. Sadgali, S. Nawal, and F. Benabbou. “Fraud detection in credit card transactions using machine learning techniques” *In 2019 1st International Conference on Smart Systems and Data Science (ICSSD)*, IEEE, pp. 1-4, 2019.

[18] A. Husejinovic, “Credit card fraud detection using naive Bayesian and c4. 5 decision tree classifiers”, *Periodicals of Engineering and Natural Sciences*, vol. 4, no. 1, pp. 1-5, 2020.

[19] N. Trivedi, S. Simaiya, U. Lilhore, and S. Sharma, “An efficient credit card fraud detection model based on machine learning methods,” *International Journal of Advanced Science and Technology*, vol. 29, no. 5, pp. 3414-3424, 2020.

[20] A. Joshi, S. Soni, and V. Jain, “An Experimental Study using Unsupervised Machine Learning Techniques for Credit Card Fraud Detection,” *GIS Science Journal,* vol. 8, no. 5, pp. 1187-1206, 2021.

[21] H. Ahmad, B. Kasasbeh, B. Aldabaybah, and E. Rawashdeh, “Class balancing framework for credit card fraud detection based on clustering and similarity-based selection (SBS),” *International Journal of Information Technology,* vol. 15, no. 1, pp. 325-333, 2023.

[22] R. Mohammed, J. Rawashdeh, and M. Abdullah, “Machine learning with oversampling and under-sampling techniques: overview study and experimental results,” *In 2020 11th International Conference on Information and Communication Systems (ICICS),* IEEE, pp. 243-248, 2020.

[23] L. Chao, T. Fong, H. Han, and J. Shang, “Clustering-based under-sampling in class-imbalanced data,” *Information Sciences*, vol. 409-410, pp. 17-26, 2017.

[24] S. Castillo, A. Bernal, and J. Rodríguez, “Object Detection in Digital Documents based on Machine Learning Algorithms,” *IAENG International Journal of Computer Science*, vol. 50, no. 2, pp. 688-699, 2023.

[25] S. Rajora, D. Li, C. Jha, N. Bharill, O. Patel, S. Joshi, and M. Prasad, “A comparative study of machine learning techniques for credit card fraud detection based on time variance,” In *2018 IEEE symposium series on computational intelligence (SSCI)*, IEEE, pp. 1958-1963, 2018.

[26] A. Sharaff, and H. Gupta, “Extra-tree classifier with metaheuristics approach for email classification,” In *Advances in Computer Communication and Computational Sciences: Proceedings of IC4S 2018*, Springer Singapore, pp. 189-197, 2018.

[27] J. Singh, and R. Banerjee, "A Study on Single and Multi-layer Perceptron Neural Network," 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, pp. 35-40, 2019.

[28] G. Kumari1, M. Rani, “A Study of AdaBoost and Bagging Approaches on Student Dataset”, *International Research Journal of Advanced Engineering and Science*, Vol. 2, no. 2, pp. 375-380, 2017.