SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V11I8P124

Volume 11 Issue 8, 244-259, August 2024


*Original Article*

# A Machine Learning Based Approach for the Fraud Detection in Imbalanced Credit Card Transaction Dataset

Rinku[1], Ashutosh Kumar Dubey[1*], Sushil Kumar Narang[2], Neha Kishore[3]

[1]*Chitkara University School of Engineering and Technology, Chitkara University, Himachal Pradesh, India.*
[2]*Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India.*
[3]*University Institute of Engineering and Technology, Maharaja Agrasen University, Himachal Pradesh, India.*

**Corresponding author: ashutosh.dubey@chitkara.edu.in*

Received: 18 June 2024 Revised: 30 July 2024 Accepted: 15 August 2024 Published: 31 August 2024

***Abstract -*** *In this study, a comprehensive evaluation of machine learning models was conducted to detect fraudulent transactions in a highly imbalanced credit card dataset. An ensemble of algorithms was utilized, including Logistic Regression (LR), k-Nearest Neighbors (kNN), Support Vector Machines (SVM), Decision Tree (DT), Random Forest (RF), AdaBoost, Gradient Boosting (GB), Multi-Layer Perceptron (MLP), and Gaussian Naïve Bayes (GNB), each chosen to address the distinct challenges posed by the dataset's skew. Preprocessing techniques, such as Synthetic Minority Over-Sampling Technique (SMOTE) and Adaptive Synthetic (ADASYN) sampling methods, were implemented to correct class imbalances, followed by feature selection through Linear Discriminant Analysis (LDA) to enhance model training efficacy. The experimental results showcased that the ensemble methods, particularly RF, outperform, offering high accuracy and specificity, evidenced by an accuracy rate of 0.9995 using ADASYN in an 80:20 training-test split. These methods effectively handled the imbalanced nature of the dataset while maintaining high levels of predictive reliability. This study demonstrates the efficacy of ensemble machine learning approaches in detecting fraud in datasets characterized by class imbalance. The strategic application of oversampling techniques, coupled with ensemble models, provides a robust framework for identifying fraudulent activities, thereby significantly reducing the risk associated with such transactions.*

***Keywords -*** *Fraud detection, Class imbalance, Ensemble learning, Oversampling techniques, Machine learning algorithms.*

## 1. Introduction

In the rapidly evolving domain of financial transactions, the detection and classification of fraudulent activities have become paramount due to the worldwide surge in credit card fraud [1-4]. This surge has necessitated the development of advanced analytical methodologies capable of accurately identifying fraudulent transactions within vast datasets [5-9]. However, a major challenge in this endeavor arises from the imbalanced nature of transactional data, where instances of fraud are significant [10-13]. This imbalance complicates the task of fraud detection, leading to high rates of false negatives for fraudulent transactions [14-18].

The motivation behind this work is to address these challenges by exploring and enhancing the ability of machine learning algorithms to detect and classify fraudulent activities within highly imbalanced credit card transaction datasets. The primary objectives of this study are threefold: (1) to assess the impact of dataset imbalance on the performance of various machine learning algorithms, (2) to explore and implement advanced preprocessing and resampling techniques aimed at mitigating the effects of data imbalance, and (3) to evaluate the effectiveness of a range of machine learning models, from traditional algorithms to ensemble approaches, in accurately detecting fraudulent transactions.

The contributions of this work are multifaceted. Firstly, it provides a comprehensive evaluation of the performance of different machine intelligent approaches in the context of imbalanced fraud detection datasets, including Logistic Regression (LR), k-Nearest Neighbors (kNN), Support Vector Machines (SVM), Decision Tree (DT), ensemble methods such as Random Forest (RF), AdaBoost, Gradient Boosting (GB), neural networks (Multi-Layer Perceptron (MLP)) and Gaussian Naive Bayes (GNB). Each algorithm is assessed for its robustness, accuracy, and efficiency in handling imbalanced data, offering valuable insights into its suitability for fraud detection tasks. Secondly, preprocessing and resampling strategies were considered to improve the performance of imbalanced datasets. These strategies include the application of oversampling methods such as the Adaptive Synthetic (ADASYN) sampling and Synthetic Minority Over-Sampling Technique (SMOTE), as well as neural network designs tailored to address class imbalance. By implementing

these methodologies, the research aims to enhance the models' sensitivity to fraudulent transactions without compromising their ability to classify legitimate activities [19-22] correctly. Finally, the work presented a framework for fraud detection by combining traditional machine learning models with ensemble learning techniques. This hybrid approach leverages the strengths of each model type, from the interpretability of LR and DTs to the predictive power of ensemble methods and the pattern recognition capabilities of neural networks, through a series of experiments conducted on a widely used credit card dataset. This study demonstrated the effectiveness of this integrated approach in detecting fraudulent activities within highly imbalanced datasets.

The remainder of this paper is organized as follows: Section 2 investigates and explores related work. Materials and methods are discussed and expanded in Section 3. Results are explored in Section 4. Section 5 discusses and analyzes these results and concludes in Section 6.

## 2. Literature Review

Related work based on machine intelligence algorithms in relation to credit card fraud classification has been discussed in this section. Recently, the application of machine intelligence algorithms in detecting credit card fraud has seen a marked rise in popularity. This increase can be attributed to the algorithms' remarkable ability to detect complex patterns within large datasets.

In 2022, Chao et al. [23] identified two main challenges in imbalanced data categorization: algorithms' performance is significantly affected by the unique characteristics of unbalanced data, and their robustness varies with imbalance ratios. Cost-sensitive algorithms notably dropped in effectiveness from 94% to 74% as the imbalance increased. They proposed using stochastic data envelopment analysis (DEA), combining statistical modeling and sampling, to assess classifier efficiency and performance accurately. In 2022, Thejas et al. [24] conducted an extensive analysis using a broad spectrum of real-time data across different domains. They employed several evaluation methods to compare models with diverse oversampling strategies. Their findings showed a significant improvement in accuracy over other methods, with a particular emphasis on the area under the curve (AUC) score. It is important to mention that their use of the Kalman filter approach, despite its cubic computational complexity, led to reduced performance on larger datasets due to increased processing time. Hilal et al. [25], in 2022, aimed to provide researchers with an understanding of the model's objectives, benefits, and limitations. Recent research has shown a trend towards unsupervised and semi-supervised models, such as Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and clustering algorithms, which have been successful in identifying fraudulent credit card transactions. During preprocessing, categorical variables were collected from the raw data and coded by Domashova and Kripak [26]. An autocoder, a specialized neural network design, and anomaly detection techniques were then utilized to construct a training sample. This sample underwent additional processing with specific techniques to correct the class imbalance. As a result, six datasets were produced, and seven different classifiers were trained on each dataset. Upon comparison of the models, the Tomek links method and the XGBoost technique were identified as having the highest classification quality for bank transactions. In 2022, Kou et al. [27] introduced three advanced imbalanced learning techniques. These contributions include resampling based on the model based on cluster size and the Hybrid Imbalanced Learning Framework (HILF), which integrates several resampling techniques to enhance performance. The proposed HILF and cluster size-distance based models have been shown to outperform significantly. Lee and Seo [28] explored the use of active learning to improve binary classification performance on imbalanced datasets. They developed a pre-selective method for faster processing and implemented active downsampling to reduce generalization errors. For datasets with severe skewness, adjusting the logistic regression's tuning parameter after each iteration proved successful. Their experiments on real-world and simulated datasets demonstrated superior performance over traditional resampling techniques. In 2022, Temraz et al. [29] introduced Counterfactual Augmentation (CFA) to tackle the issue of class imbalance in binary classification problems. CFA generates synthetic counterfactuals for the minority class using a case-based reasoning approach. This approach sets CFA apart from conventional methods, which often depend on extrapolation or interpolation techniques. A neural network-based model was suggested by Li et al. [30]. It has been used to generate data for managing credit risk based on distribution suitability. For this, they have arranged the data points from the credit class using a distance-based metric process. It is useful in catering to classes based on risk or not risk. A major drawback of the proposed model, however, was that the samples collected for the Nyström method's economic significance could not be adequately explained. Different research papers were reviewed by Cherif et al. [31] based on intelligent technologies, including big data and deep learning in terms of data security. Based on the detailed analysis, they discussed the major factors that influence methodological adoptions, as well as their advantages and disadvantages. In 2023, Karunachandra et al. [32] utilized various machine learning algorithms to identify fraudulent compensation activities among online merchants. Among the techniques tested, the kNN approach demonstrated superior performance, achieving an accuracy rate of 83.82%. In comparison, CNN and LSTM networks yielded lower accuracies of 49.39% and 51.13%, respectively. In 2023, Gupta et al. [33] identified XGBoost as yielding the highest precision, F1-Score, and accuracy compared to other classifiers evaluated. By applying three different data balancing techniques to the chosen models—DT, CNN, and LR—they aimed to enhance the overall

performance of classifiers. Among the techniques, random over sampling was found to be the most effective for the selected algorithm, outperforming the SMOTE method and random under-sampling. The XGBoost Classifier, when combined with random over-sampling, achieved the highest scores for accuracy and other performance metrics. Afriyie et al. [34] used LR, DT, and RF to detect fraudulent online credit card transactions. They balanced the dataset using an under-sampling strategy to prevent bias towards the majority class and reduce the risk of overfitting. The RF model emerged as the most effective, with an AUC value of 98.9% and an accuracy value of 96.0%, proving to be the best-suited model for predicting fraudulent transactions. In 2023, Noviandy et al. [35] utilized the XGBoost algorithm and data augmentation for credit card fraud detection, demonstrating improved accuracy and addressing imbalanced datasets. Their method incorporates SMOTEENN and historical data, enhanced precision, and recall. This approach benefits financial management by boosting integrity and customer trust. Alraddadi [36] studied online payment preferences, highlighting credit/debit card fraud risks. A DT Algorithm-based model for fraud detection and prevention was proposed. Surveying 102 international students revealed that 95.9% understood fraud mechanics, and 81.6% would use the model to combat fraud. In 2023, Prabhakaran and Nedunchelian [37] introduced a model for fraud detection based on deep learning and cat swarm optimization. Optimization has been used for feature selection. For the fraud classification, they have used a recurrent unit based on a chaotic krill herd algorithm. This approach, validated by extensive simulations, demonstrates superior performance over existing methods. Ileberi et al. [38] proposed a credit card fraud detection engine leveraging machine learning and genetic algorithms for feature selection. They employed DT, RF, LR, ANN, and NB classifiers. Tested on a European cardholder dataset, it outperformed existing systems. Leevy et al. [39] applied the CatBoost algorithm for fraud detection and classification, considering different performance metrics, including AUC. Ahmad et al. [40] proposed a framework for the grouping of fraud and normal instances based on a fuzzy C-means algorithm. Their algorithm is found to be efficient in grouping these instances. In 2023, Abd et al. [41] proposed a framework to address credit card fraud detection, focusing on resolving the imbalanced dataset issue through hybrid sampling and oversampling techniques. This approach significantly improved fraud detection, achieving 99.9% accuracy compared to existing algorithms.

Recent studies address imbalanced data classification and credit card fraud detection, utilizing a variety of methods, including advanced sampling techniques, machine learning algorithms, and models like cluster-based resampling and active learning. Key findings highlight the effectiveness of hybrid sampling, stochastic DEA, and algorithms like XGBoost and RF in enhancing accuracy and model performance. Innovations like counterfactual augmentation and oppositional cat swarm optimization further push the boundaries of fraud detection. Overall, these approaches show significant promise in improving fraud detection accuracy and handling imbalanced datasets.

## 3. Materials and Methods

The experiments in this study were conducted using a widely recognized credit card dataset that includes transactions made by European cardholders in September 2013 [19, 20]. This dataset contains 492 fraudulent transactions out of a total of 284,807 transactions recorded over two days. Since fraudulent transactions constitute only 0.172% of the total, the dataset is highly imbalanced [19, 20]. It is available on the Kaggle repository. In this dataset, a target class value of 1 indicates fraud, while 0 indicates non-fraud.

The selection of algorithms for fraud detection and classification using the credit card dataset employs a diverse array of methods to tackle the challenge posed by its significant imbalance. The selected algorithms comprise four machine learning models: LR, kNN, SVM, and DT. Additionally, ensemble methods such as RF, AdaBoost, and GB have been chosen for their robustness and accuracy. The study also incorporates MLP and a probabilistic approach, GNB, to provide a comprehensive analysis of the dataset. Given the context provided, the choice of algorithms for fraud detection in the credit card dataset employs a strategic approach to address the significant imbalance within the dataset. Machine learning models like LR, kNN, SVM, and DT provide a solid foundation, each with unique strengths in handling classification problems. Ensemble methods, including RF, AdaBoost, and GB, are chosen for their enhanced accuracy and ability to reduce overfitting, making them particularly effective against the dataset's imbalance. Neural networks, specifically MLP, offer advanced pattern recognition capabilities essential for detecting complex fraudulent behaviors. GNB adds a probabilistic approach, which is beneficial for its efficiency with high-dimensional data. This multifaceted selection ensures a thorough analysis, maximizing the chances of accurately identifying fraud amidst the dataset's challenges.

Initially, the process starts with data collection, where the credit card dataset is considered to serve as the foundation for analysis. Following the dataset selection, preprocessing is undertaken to ensure the data is in an optimal state for analysis. This stage involves handling any missing values present within the dataset, encoding categorical variables to numerical ones if they exist, and addressing the issue of class imbalance, which is prevalent in fraud detection datasets due to the rarity of fraudulent transactions compared to legitimate ones. SMOTE and ADASYN were used to address the class imbalance problem. These methods are designed to balance class distribution. SMOTE works by creating synthetic examples rather than simply duplicating minority class instances. ADASYN builds on the concept of SMOTE with an

added strategy to adaptively generate minority data samples. Once the data is pre-processed, the next step is feature selection or reduction. Here, Linear Discriminant Analysis (LDA) was applied for dimensionality reduction, aiming to simplify the dataset while retaining the most relevant information for detecting fraud. With a refined set of features, the algorithm proceeds to the model training phase. This phase is extensive and involves training multiple models to explore various methodologies for fraud detection. The models include LR, kNN, SVM, DT, RF, AdaBoost, GB, MLP, and GNB. Each of these models offers a unique approach to classification and is evaluated to determine its effectiveness in detecting fraudulent transactions. The final step in the algorithmic approach is the prediction phase, where the trained models, possibly enhanced through ensemble learning, are used to predict and identify fraudulent transactions within the credit card dataset. This comprehensive approach, from data collection to prediction, aims to effectively detect fraudulent activities, thereby minimizing the risks associated with credit card fraud (Figure 1).

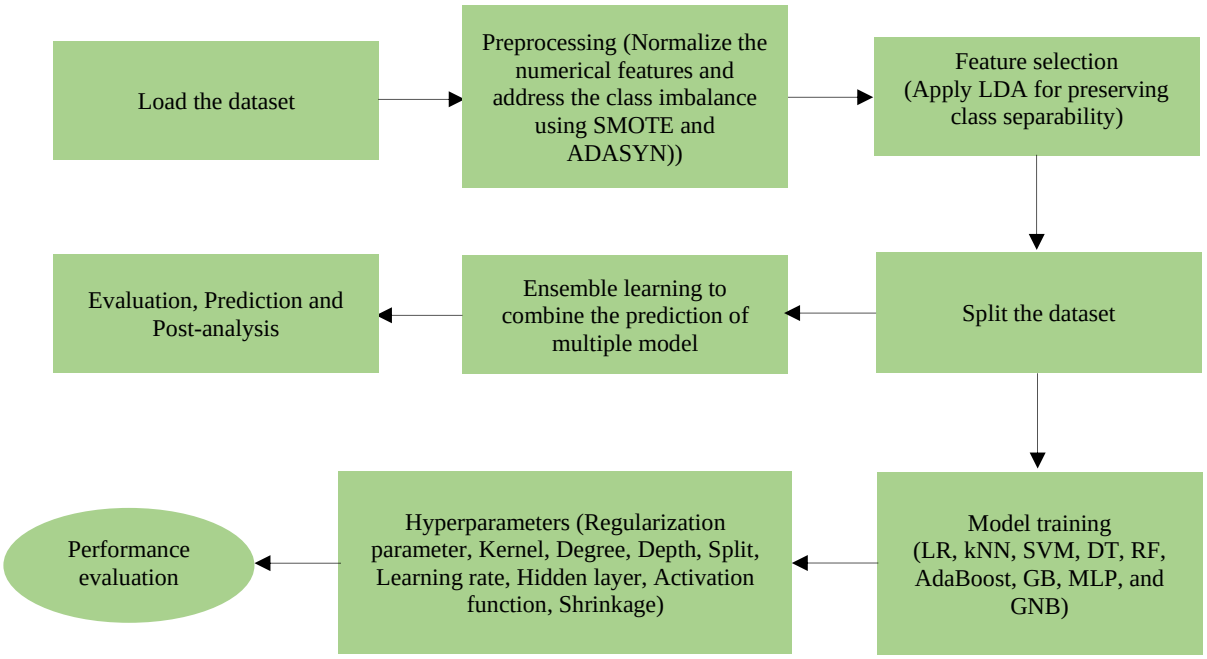



**Fig. 1 Working mechanism of the complete work**

### *3.1. Logistic Regression*

LR is utilized for binary classification tasks. It is a classification algorithm. It models the relationship between independent variables and the likelihood of an event occurring. This is achieved through the logistic function, as depicted in Equations 1 and 2:

$$S(z)=\frac{1}{1+e^{-z}} \quad (1)$$

Here, z shows the following.

$$z=\beta_0+\beta_1x_1+\beta_2x_2+\beta_3x_3+\cdots+\beta_nx_n \quad (2)$$

Where

- S(z) is the output of the logistic function
- e is the base of the natural algorithm
- $\beta_0$ is bias or intercept term
- $\beta_1,\beta_2,\beta_3,\ldots\ \beta_n$ are the coefficient or associated weights with the features $x_1, x_2, x_3 \ldots, x_n$

### *3.2. k-Nearest Neighbors*

kNN predicts data using nearest neighbors for classification and regression. In such a scenario, each data point in the dataset is characterized by a set of features (attributes) that describe it, along with a corresponding class label that denotes its category or group. These features are used to represent the data point in a multidimensional feature space, where the dimensions correspond to the attributes. Each data point is defined by its features (attributes) and an associated class label (Equation 3).

$$D=\{(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\} \quad (3)$$

Where $x_i$ represents the feature of i-th data point, and $y_i$ is its corresponding label.

kNN determines the similarity between data points using a distance metric (Equation 4).

$$Distance\ (x_i,x_j)=\sqrt{\sum_{k=1}^{p}(x_{ik}-x_{jk})^2} \quad (4)$$

Where p is the features count.

Subsequently, a value for *k* is chosen, representing the number of neighbors to consider. This value is a hyperparameter that must be tuned according to the dataset and the specific problem. Given a new data point $x_{new}$ for which you want to predict the class label, find the k nearest neighbors from the training dataset based on the chosen distance metric.

To predict the class label for a new data point $x_{new}$, identify the *k* nearest neighbors from the training dataset using the selected distance metric (Equation 5).

$$N_{new} = argmin_{N^D}(\sum_{i \in N} Distance(x_{new}, x_i)) \quad (5)$$

$N_{new}$ is the set of indices. $argmin_{N^D}$ finds the set N that minimizes the expression for the selection of the best N. D represents a distance metric that quantifies the total distance or dissimilarity.

Finally, Count the occurrences of each class in the k nearest neighbors and assign the class label that has the majority (Equation 6).

$$Prediction = argmax_c\,(\sum_{i \in N_{new}} \delta(y_i, c)) \quad (6)$$

Where $\delta(y_i, c)$ is 1 if $y_i = c$ otherwise 0. $argmax_c$ selects the best class c.

### *3.3. Support Vector Machine*

SVM is a supervised machine learning algorithm used for classification and regression tasks. This technique constructs hyperplanes in high-dimensional space. SVM focuses on maximizing the margin, the distance between the hyperplane and nearest class points, to improve generalizability and robustness in binary classification tasks.

Given a dataset of m training samples $\{(x_1, y_1), (x_2, y_2), \ldots\ldots\ldots, (x_n, y_n)\}$, where each $x_1 \varepsilon R_n$ is a feature vector, and $y_i \varepsilon \{-1,1\}$ is the class label of the $i^{th}$ set, the goal of SVM is to find the optimal separating hyperplane that maximizes the margin between the two classes. The hyperplane can be defined by the Equation 7:

$$w.x+b=0 \quad (7)$$

Where w is the weight vector, and b is the bias term.

The main objective to minimize Equation 8:

$$\text{Min: } \frac{1}{2}\|w\|^2 \quad (8)$$

The condition for the minimization is $y_i(w.\ x_i + b) \geq 1,\ \forall\ i=1,2,\ldots..,m$

This constraint ensures that all data points are classified accurately.

### *3.4. Decision Tree*

DT is a supervised technique that recursively divides data into subsets based on input feature values. The decisions made by each internal node of the tree are based on certain features, and each leaf node represents the anticipated output (class label or regression value). The objective is to identify the feature and threshold value that most effectively divides the data into groups that are more like each other with respect to the target variable. The main aim is to identify the decision rules that optimize the target variable's homogeneity within every subgroup. For this, Gini impurity and entropy measurements have been considered (Equations 9 and 10).

$$\text{Gini(p)} = 1 - \sum_{i=1}^{k} p_i^2 \quad (9)$$

$$\text{Entropy(p)} = -\sum_{i=1}^{k} \log_2(p_i) \quad (10)$$

Where $p_i$ is the proportion of the samples belonging to class *i* in the node

### *3.5. Random Forest*

RF is an ensemble learning method that combines the predictions of multiple decision trees to improve the overall accuracy and robustness of the model.

N: Total number of data points in the training set.
M: Number of DTs in the RF.
m: Feature count.

The aggregation of the DTs based on the new data point (X) is shown in (Equation 11).

$$\bar{y}\ (\text{prediction}) = \frac{1}{M}\sum_{i=1}^{M} f_i(X) \quad (11)$$

Where $f_i(X)$ is the prediction of the i-th decision tree.

### *3.6. AdaBoost*

AdaBoost combines weak learners into a stronger model by leveraging strengths. It focuses on iteratively enhancing the performance of these weak learners. The steps of AdaBoost are as follows.

Step 1: Assign equal weights to all training examples. If you have N training instances, each weight is initially set to $\frac{1}{N}$.

Step 2: For each iteration (t = 1 to T, where T is the total number of iterations or weak learners):

2.1 Train a weak learner on the training data, where the data is weighted based on the previous iteration's results. The weak learner is usually a model that is only slightly better than random chance.

2.2 Calculate the error of the weak learner, which is the sum of the weights of the misclassified instances (Equation 12)

$$\epsilon_t = \frac{\sum_{i=1}^{N} w_i . l(h_t(x_i) \neq y_i)}{\sum_{i=1}^{N} w_i} \quad (12)$$

Where

- $\epsilon_t$ is the error of the weak learner at iteration t.

- $h_t(x_i)$ is the prediction of the week learner, for example $x_i$.
- $y_i$ is the true label of the example $x_i$.
- $w_i$ is the weight of the example $x_i$.
- $l()$ is an indicator function that outputs 1 if the specified condition is true and 0 otherwise.

Step 3: Calculate the weight ($\alpha_t$) of the weak learner based on its error (Equation 13).

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \quad (13)$$

The weight ($\alpha_t$) is used to give more importance to the predictions of the weak learner in the final combined model.

Step 4: Update the weights of the training instances. Increase the weights of the misclassified cases and decrease the weights of the correctly classified (Equation 14).

$$w_{t+1,i} = w_i.\exp\left(-\alpha_t.y_i.h_t(x_i)\right) \quad (14)$$

Step 5: Normalize the updated weights to ensure that they sum to 1 (Equation 15).

$$w_{t+1,i} = \frac{w_{t+1,i}}{\sum_{i=1}^{N} w_{t+1,i}} \quad (15)$$

Step 6: Combine the weak learners into a strong learner by assigning a weight to each weak learner's prediction (Equation 16).

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t.h_t(x)\right) \quad (16)$$

Where H(x) is the final prediction.

### *3.7. Gradient Boosting*

GB improves predictions by sequentially adding models, typically decision trees, to correct previous errors. The process begins by initializing a base model and calculating its residuals—the differences between the predicted and actual values. For each subsequent tree it fits to these residuals, effectively reducing the error.

If the current model at stage t-1 is $F_{t-1}(x)$, the next tree, $h_t(x)$, is trained to predict the negative gradient. The model is updated as shown in Equation 17. This loss function has been evaluated at $F_{t-1}(x)$.

$$F_t(x) = F_{t-1}(x) + \eta \times h_t(x) \quad (17)$$

Where $\eta$ is the learning rate, controlling how fast the model learns. This process is repeated, gradually improving the model's accuracy by focusing training on hard-to-predict instances.

### *3.8. Multi-Layer Perceptron*

MLP is a type of artificial neural network that consists of multiple layers of nodes, each connected to the nodes in the adjacent layers. The MLP is a feedforward neural network that processes information from the input layer to the output layer, representing input data features. If it consists of 'n' features, it determines 'n' nodes in the input layer. These nodes are often denoted as $x_1, x_2, x_3 \ldots, x_n$. Between the input and output layers, there can be one or more hidden layers. Each node in a hidden layer is connected to every node in the previous layer (input or hidden layer), and each connection has an associated weight. Let $z_i^{(l)}$ represent the weighted sum of inputs to node i in layer l, and let $a_i^{(l)}$ represent the activation of node I in layer l. The activation is generally a nonlinear function applied to the weighted sum. The most common activation functions are the sigmoid function, hyperbolic tangent (tanh), or Rectified Linear Unit (ReLU).

For node i in layer l (Equation 18 and Equation 19):

$$a_i^{(l)} = \sum_{j=1}^{m^{(l-1)}} w_{ij}^{(l)} a_j^{(l-1)} + b_i^{(l)} \quad (18)$$

$$a_i^{(l)} = activation(z_i^{(l)}) \quad (19)$$

Here

- $w_{ij}^{(l)}$ is the weight of ij representing layer l.
- $b_i^{(l)}$ is the i[th] bias term for layer l.
- $m^{(l-1)}$ is the number of nodes in the previous layer.

The output layer produces the result.

### *3.9. Gaussian Naïve Bayes*

GNB model is based on Bayes' theorem. "Naive" refers to the assumption that features in classification are conditionally independent given the class label unaffected by other features. Bayes' theorem is a fundamental concept in probability theory, and it relates the conditional and marginal probabilities of random events. It is shown in Equation 20:

$$P\left(\frac{y}{X}\right) = \frac{P\left(X/y\right).P(y)}{P(X)} \quad (20)$$

Where

- $P\left(\frac{y}{X}\right)$ depicts the probability of y (X shows the feature)
- $P\left(X/y\right)$ depicts the likelihood of class y
- $P(y)$ is the probability (prior) of class y.
- $P(X)$ is the considering probability for the features X

In the case of GNB, it is assumed that the likelihood P(X|y) follows a Gaussian (normal) distribution. This is appropriate when the features are continuous and can be modeled by a bell-shaped curve. The Probability Density Function (PDF) of the Gaussian distribution is given by (Equation 21):

$$f(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (21)$$

Where

- $\sigma$ denotes the distribution standard deviation
- $\mu$ denotes the distribution mean

For each feature $X_i$ and class y, estimate the mean $\mu_{y,i}$ and the standard deviation $\sigma_{y,i}$ from the training data.

To predict a new data point, calculate the posterior probability for each class using Bayes' theorem and choose the class with the highest probability (Equation 22).

$$\text{Prediction} = \text{argmax}_c \, P(\frac{y}{x}) \tag{22}$$

### *3.10. Linear Discriminant Analysis*

LDA is used for dimensionality reduction and classification. LDA operates under the assumption that the data for each class is normally distributed and has the same covariance matrix. The goal of LDA is to find a linear combination of features that characterizes or separates two or more classes. This linear combination is chosen in such a way that the distance between the means of different classes is maximized, and the variance within each class is minimized.

Step 1: For each class, calculate the mean vector, which is the average of all data points belonging to that class (Equation 23).

$$m_i = \frac{1}{n_i} \sum_{k=1}^{n_i} x_{ik} \tag{23}$$

Where $m_i$ is the mean vector for the $i^{th}$ class, the count of data points is represented by $n_i$, and $x_{ik}$ is the $k^{th}$ data point in $i^{th}$ class.

Step 2: Calculate the scatter matrices for between classes ($S_B$) and within class ($S_w$) (Equations 24 and 25).

$$S_B = \sum_{i=1}^{c} n_i (m_i - m)(m_i - m)^T \tag{24}$$

$$S_w = \sum_{i=1}^{c} \sum_{k=1}^{n_i} (x_{ik} - m_i)(x_{ik} - m_i)^T \tag{25}$$

Where c signifies the classes count, m is the overall mean vector, and $n_i$ is the number of data points in class i.

Step 3: Solve the generalized eigenvalue problem for $S_w^{-1} S_B$.

$S_w^{-1} S_B v = \Lambda v$ where v is eigenvector and $\Lambda$ is eigenvalue.

Step 4: Sort the eigenvalues in descending order and choose the top k eigenvectors as discriminants, where k is the number of classes minus one (to avoid overfitting).

Step 5: Form a matrix W with the selected eigenvectors as columns. Project the data onto the new subspace using $= W^T x$ , where y is the transformed data.

LDA aims to maximize the distance between class means while minimizing the spread (variance) within each class. This makes it a useful technique for dimensionality reduction and classification, especially when the assumption of normal distribution holds for the data.

The algorithm of the complete approach is shown below.

Step 1: Load the dataset.

Step 2: Preprocessing:

2.1 Normalize/standardize the numerical features.

2.2 Address the class imbalance using SMOTE and ADASYN.

Step 3: Feature Selection:

3.1 Apply LDA to reduce dimensionality while preserving class separability.

Step 4: Split the dataset into training and testing sets.

Step 5: Model Training:

5.1 Train the models listed (LR, kNN, SVM, DT, RF, AdaBoost, GB, MLP, GNB) using the training set.

5.2 Perform hyperparameter tuning to find the best settings for each model.

Step 6: Apply ensemble learning techniques to combine the predictions of multiple models.

Step 7: Use the selected model(s) to predict fraudulent transactions in unseen data and perform evaluation, prediction and post-analysis.

Step 8: End

## 4. Results

For the experimentation, Python 3.9, an Intel(R) Core(TM) i5-10210U CPU with a base clock speed of 1.60 GHz, the Windows 10 operating system, and 16 GB of RAM were utilized. The performance measured and considered for the experimentation are as follows.

Accuracy: It represents the ratio of correct results (both true positives and true negatives) to the total number of cases examined.

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

Where:

TP = True Positives

TN = True Negatives

FP = False Positives

FN = False Negatives

Precision (Positive predictive value): Precision measures the proportion of positive identifications that were correct.

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

Recall (Sensitivity or True Positive Rate): It quantifies the proportion of actual positives that were correctly identified.

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

Specificity (True Negative Rate): It measures the correctly identified segment that is correctly identified as actual negatives.

$$\text{Specificity} = \frac{TN}{TN+FP}$$

F1-Score: It is the harmonic mean of precision and recall.

$$\text{F1-Score} = 2\times \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right)$$

The split ratio considered for the experimentation are 70:30, 75:25 and 80:20. The results presented in Table 1 offer a detailed comparison of various machine learning models using multiple metrics, including precision, recall, specificity, F1-Score, accuracy, AUC-ROC, and training and testing times. These models, when applied with ADASYN and SMOTE sampling techniques and different training-test split ratios (80:20, 75:25, 70:30), aim to address the challenges of imbalanced datasets.

The evaluated models encompass LR, kNN, DT, SVM, RF, GNB, MLP, Adaboost, and GB. Among the various methods, the RF model demonstrated superior performance in terms of accuracy, specificity, and F1 score. Notably, with an 80:20 split using ADASYN, it achieved an accuracy of 0.9995, a specificity of 0.9997, an F1-score of 0.8528, and a precision of 0.8485. The high accuracy and specificity make it highly reliable for both positive and negative class predictions. The GB model, with an 80:20 split using ADASYN, achieved an accuracy of 0.9937 and a specificity of 0.9938. The Adaboost model, particularly with an 80:20 split and ADASYN, reached an accuracy of 0.992 and a specificity of 0.9922. The training and testing time for the SVM is notably high at 42,160.34 seconds, which is the highest among all listed models. This long duration reflects the computationally intensive nature of SVM with large datasets and complex feature spaces. The GB model also has a significant training and testing time of 870.4889 seconds. While not as extreme as SVM, the time is still considerable, hinting at the iterative nature of boosting algorithms, which build multiple trees sequentially, each one correcting errors made by the previous ones. The RF model, with a training and testing time of 711.1515 seconds, demands moderate computational resources. This ensemble method builds numerous DTs and aggregates their predictions, resulting in robust model performance despite its complexity. These findings indicate that the RF model as a robust solution for imbalanced datasets, delivering an exceptional balance between accuracy and discriminative power. GB and Adaboost also demonstrated promising outcomes, suggesting their suitability in situations demanding predictive accuracy and model reliability. Conversely, SVM, KNN, and LR, despite their wide application in various machine learning endeavors, exhibited limitations in this specific context. The variability observed across different split ratios appears to be minor; therefore, any of the split ratios can be considered viable for model training and evaluation.

Figure 2 shows the confusion matrices for different machine learning models that have been trained using either SMOTE or ADASYN to address class imbalance. Each confusion matrix corresponds to a specific model and data balancing technique, showcasing the TP, TN, FP, and FN rates achieved by each model.

AUC-ROC is used on various threshold settings to analyze classification performance. The AUC represents a degree of separability, telling how much a model is capable of distinguishing between classes. Higher values are indicative of better model performance (Figure 3). RF shows high AUC-ROC scores across all splits and sampling methods, with a peak score of 0.991757 using SMOTE for the 80:20 split. This model excels in distinguishing between classes due to its ability to handle complex interrelations in large datasets and its ensemble method, which significantly enhances its performance. Adaboost also demonstrates excellent AUC-ROC scores, which are among the highest across the different splits and samplers, peaking at 0.987788 with SMOTE for the 70:30 split. Its performance indicates a strong adaptability to varying data distributions and the efficacy of its boosting strategy. GB maintains robust AUC-ROC scores above 0.983 in all scenarios, highlighting its effectiveness in classification tasks and confirming the strength of ensemble learning techniques, especially when dealing with imbalanced datasets. These evaluations reflect the models' abilities to distinguish between class labels effectively. Ensemble methods like RF, Adaboost, and GB are found to be strong in handling the challenges presented by imbalanced data, as indicated by their AUC-ROC scores.

Figure 4 presents a pair plot, also known as a scatterplot matrix, utilized to visualize the distribution of a dataset across several quantitative variables. The x-axis delineates the range of possible values for the variables, while the y-axis quantifies the frequency of data points within each bin. Notably, the off-diagonal plots are omitted, indicating that bivariate conditions are not included in this matrix. Each histogram provides insights into the distribution of an individual variable, highlighting characteristics such as skewness, normality, presence of gaps, or outliers. Several histograms exhibit a bimodal distribution, signaling two prevalent groupings or values within the data. A few histograms appear nearly uniform, suggesting a similar frequency of values across their range. Predominantly, the histograms are left-skewed, indicating a concentration of values toward the right and fewer lower-value occurrences. The description suggests that these histograms may be relevant in distinguishing between adjacent variables characterized as fraudulent or non-fraudulent. The skewness and distribution patterns can be

particularly informative in such contexts, as they may reflect underlying trends or behaviors associated with fraudulent activities. Figure 5 depicts a heatmap, which is a data visualization technique that shows the magnitude of a phenomenon as color in two dimensions. The color bar on the right functions as a legend, displaying a gradient that transitions from dark to light shades.

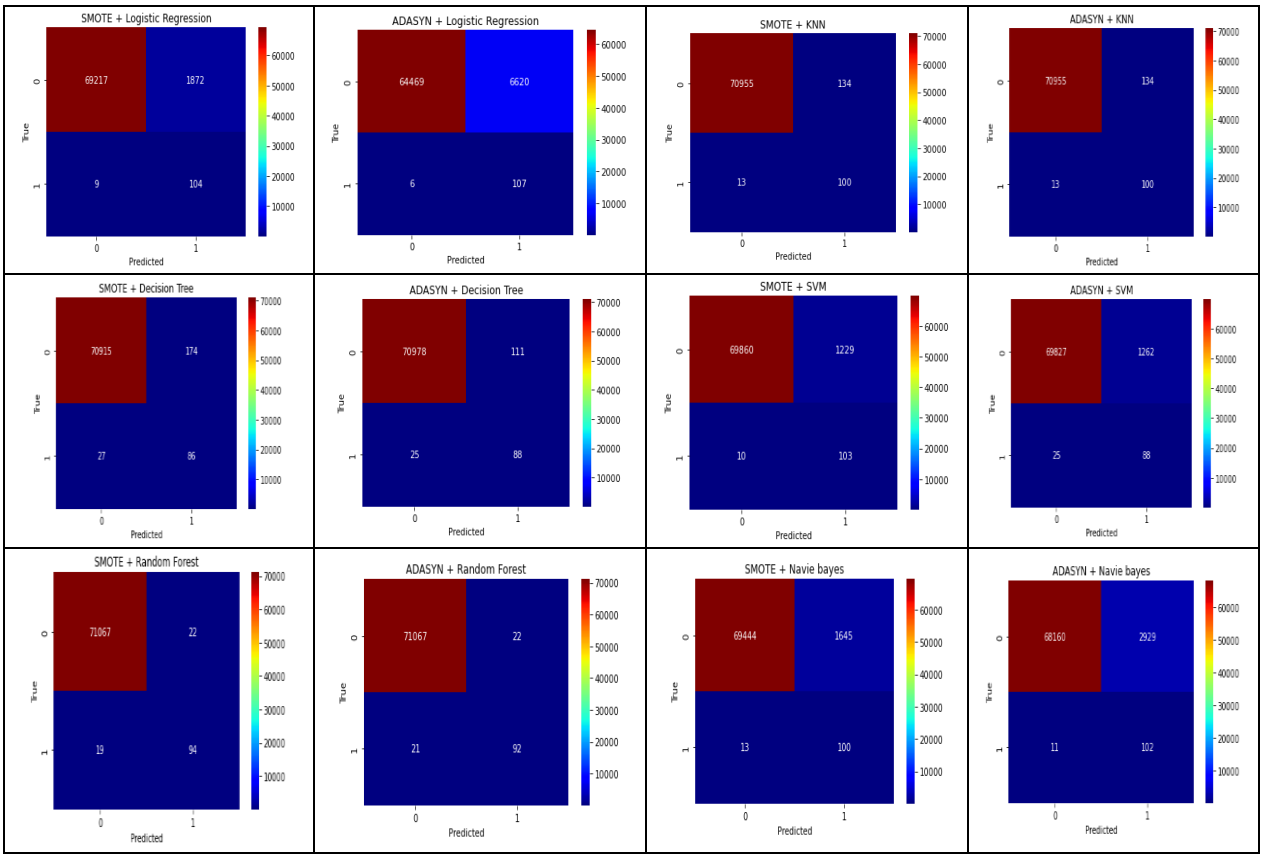
The darker end of the gradient represents higher values, while the lighter end corresponds to lower values. The heatmap includes a correlation coefficient of 1 along the diagonal, which indicates the maximum value and signifies that each variable is perfectly correlated with itself. Blocks of similar colors represent clusters of variables that are closely related to each other. This pattern of clustering can identify groups of variables with similar properties or behaviors. This heatmap is used to visualize the correlations among various variables, compare measurements across different conditions, and highlight similarities within a dataset.

RF was found to be a robust solution for handling imbalanced datasets, delivering a balance between accuracy and discriminative power. GB and Adaboost also demonstrated promising outcomes, making them suitable for situations requiring predictive accuracy and model reliability. The long training time of SVM highlights its limitations in this context, while RF's moderate time requirement and good performance make it a preferred choice for large and complex datasets.

**Table 1. Model performance comparison based on precision, recall, specificity, F1-score, accuracy, and time, categorized by the sampler used in different models**

| S.No. | Model | Precision | Recall | Specificity | F1-Score | Accuracy | Time (Training + Test) | Ratio | Sampler |
|---|---|---|---|---|---|---|---|---|---|
| 1 | LR | 0.0464 | 0.8878 | 0.9685 | 0.0881 | 0.9684 | 2.2031 | 80:20 | ADASYN |
| 2 | kNN | 0.014 | 0.6327 | 0.9233 | 0.0274 | 0.9228 | 64.4363 | 80:20 | ADASYN |
| 3 | DT | 0.3553 | 0.8265 | 0.9974 | 0.4969 | 0.9971 | 53.1088 | 80:20 | ADASYN |
| 4 | SVM | 0.0024 | 0.2245 | 0.8396 | 0.0048 | 0.8385 | 35977.95 | 80:20 | ADASYN |
| 5 | RF | 0.8485 | 0.8571 | 0.9997 | 0.8528 | 0.9995 | 711.1515 | 80:20 | ADASYN |
| 6 | GNB | 0.1461 | 0.7245 | 0.9927 | 0.2432 | 0.9922 | 0.3438 | 80:20 | ADASYN |
| 7 | MLP | 0.0494 | 0.898 | 0.9702 | 0.0937 | 0.9701 | 686.403 | 80:20 | ADASYN |
| 8 | Adaboost | 0.1657 | 0.898 | 0.9922 | 0.2798 | 0.992 | 472.3728 | 80:20 | ADASYN |
| 9 | GB | 0.2027 | 0.9082 | 0.9938 | 0.3315 | 0.9937 | 1218.037 | 80:20 | ADASYN |
| 10 | LR | 0.095 | 0.9204 | 0.9861 | 0.1722 | 0.986 | 6.1717 | 75:25 | ADASYN |
| 11 | kNN | 0.0129 | 0.646 | 0.9214 | 0.0253 | 0.9209 | 74.1705 | 75:25 | ADASYN |
| 12 | DT | 0.2984 | 0.8053 | 0.997 | 0.4354 | 0.9967 | 48.5152 | 75:25 | ADASYN |
| 13 | SVM | 0.0023 | 0.2301 | 0.8388 | 0.0045 | 0.8378 | 33432.96 | 75:25 | ADASYN |
| 14 | RF | 0.7869 | 0.8496 | 0.9996 | 0.817 | 0.9994 | 535.4685 | 75:25 | ADASYN |
| 15 | GNB | 0.1386 | 0.7345 | 0.9927 | 0.2331 | 0.9923 | 0.3906 | 75:25 | ADASYN |
| 16 | MLP | 0.068 | 0.9115 | 0.9801 | 0.1265 | 0.98 | 1038.889 | 75:25 | ADASYN |
| 17 | Adaboost | 0.1347 | 0.9027 | 0.9908 | 0.2345 | 0.9906 | 374.8289 | 75:25 | ADASYN |
| 18 | GB | 0.1835 | 0.9027 | 0.9936 | 0.3049 | 0.9935 | 955.4467 | 75:25 | ADASYN |
| 19 | LR | 0.0689 | 0.9118 | 0.9804 | 0.1281 | 0.9802 | 2.9218 | 70:30 | ADASYN |
| 20 | kNN | 0.0118 | 0.6324 | 0.9153 | 0.0231 | 0.9148 | 84.1175 | 70:30 | ADASYN |
| 21 | DT | 0.2935 | 0.8309 | 0.9968 | 0.4338 | 0.9965 | 41.9518 | 70:30 | ADASYN |
| 22 | SVM | 0.0022 | 0.2353 | 0.8288 | 0.0043 | 0.8278 | 39358.08 | 70:30 | ADASYN |
| 23 | RF | 0.7933 | 0.875 | 0.9996 | 0.8322 | 0.9994 | 497.3387 | 70:30 | ADASYN |
| 24 | GNB | 0.144 | 0.7721 | 0.9927 | 0.2428 | 0.9923 | 0.3594 | 70:30 | ADASYN |
| 25 | MLP | 0.0297 | 0.9338 | 0.9513 | 0.0575 | 0.9513 | 639.1534 | 70:30 | ADASYN |
| 26 | Adaboost | 0.149 | 0.9191 | 0.9916 | 0.2564 | 0.9915 | 343.6525 | 70:30 | ADASYN |
| 27 | GB | 0.1938 | 0.9265 | 0.9939 | 0.3206 | 0.9938 | 870.4889 | 70:30 | ADASYN |
| 28 | LR | 0.1045 | 0.9286 | 0.9863 | 0.1878 | 0.9862 | 7.5466 | 80:20 | SMOTE |
| 29 | kNN | 0.019 | 0.5612 | 0.9502 | 0.0368 | 0.9495 | 63.9386 | 80:20 | SMOTE |
| 30 | DT | 0.4235 | 0.7347 | 0.9983 | 0.5373 | 0.9978 | 55.1401 | 80:20 | SMOTE |
| 31 | SVM | 0.0027 | 0.2347 | 0.8515 | 0.0054 | 0.8504 | 42160.34 | 80:20 | SMOTE |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 32 | RF | 0.8737 | 0.8469 | 0.9998 | 0.8601 | 0.9995 | 578.3685 | 80:20 | SMOTE |
| 33 | GNB | 0.1449 | 0.7245 | 0.9926 | 0.2415 | 0.9922 | 0.3282 | 80:20 | SMOTE |
| 34 | MLP | 0.1778 | 0.898 | 0.9928 | 0.2968 | 0.9927 | 895.4473 | 80:20 | SMOTE |
| 35 | Adaboost | 0.154 | 0.8878 | 0.9916 | 0.2624 | 0.9914 | 406.636 | 80:20 | SMOTE |
| 36 | GB | 0.2159 | 0.8878 | 0.9944 | 0.3473 | 0.9943 | 1033.081 | 80:20 | SMOTE |
| 37 | LR | 0.0419 | 0.885 | 0.9678 | 0.08 | 0.9677 | 2.203 | 75:25 | SMOTE |
| 38 | kNN | 0.0162 | 0.531 | 0.9488 | 0.0315 | 0.9481 | 74.561 | 75:25 | SMOTE |
| 39 | DT | 0.3563 | 0.7788 | 0.9978 | 0.4889 | 0.9974 | 44.5935 | 75:25 | SMOTE |
| 40 | SVM | 0.0017 | 0.1504 | 0.8573 | 0.0033 | 0.8562 | 37636.58 | 75:25 | SMOTE |
| 41 | RF | 0.8291 | 0.8584 | 0.9997 | 0.8435 | 0.9995 | 523.0098 | 75:25 | SMOTE |
| 42 | GNB | 0.1381 | 0.7345 | 0.9927 | 0.2325 | 0.9923 | 0.3281 | 75:25 | SMOTE |
| 43 | MLP | 0.1039 | 0.9204 | 0.9874 | 0.1867 | 0.9873 | 693.4781 | 75:25 | SMOTE |
| 44 | Adaboost | 0.1497 | 0.9115 | 0.9918 | 0.2572 | 0.9916 | 379.3088 | 75:25 | SMOTE |
| 45 | GB | 0.1892 | 0.9027 | 0.9939 | 0.3129 | 0.9937 | 977.457 | 75:25 | SMOTE |
| 46 | LR | 0.0417 | 0.8971 | 0.9671 | 0.0796 | 0.967 | 1.8905 | 70:30 | SMOTE |
| 47 | kNN | 0.0152 | 0.5147 | 0.9467 | 0.0295 | 0.946 | 84.717 | 70:30 | SMOTE |
| 48 | DT | 0.3737 | 0.8162 | 0.9978 | 0.5127 | 0.9975 | 41.7655 | 70:30 | SMOTE |
| 49 | SVM | 0.0023 | 0.2279 | 0.8433 | 0.0046 | 0.8424 | 33144.06 | 70:30 | SMOTE |
| 50 | RF | 0.8322 | 0.875 | 0.9997 | 0.853 | 0.9995 | 485.6037 | 70:30 | SMOTE |
| 51 | GNB | 0.1429 | 0.7647 | 0.9927 | 0.2407 | 0.9923 | 0.3281 | 70:30 | SMOTE |
| 52 | MLP | 0.1468 | 0.9044 | 0.9916 | 0.2526 | 0.9915 | 681.9175 | 70:30 | SMOTE |
| 53 | Adaboost | 0.1353 | 0.9191 | 0.9906 | 0.2358 | 0.9905 | 352.2616 | 70:30 | SMOTE |
| 54 | GB | 0.1856 | 0.9118 | 0.9936 | 0.3085 | 0.9935 | 891.6927 | 70:30 | SMOTE |



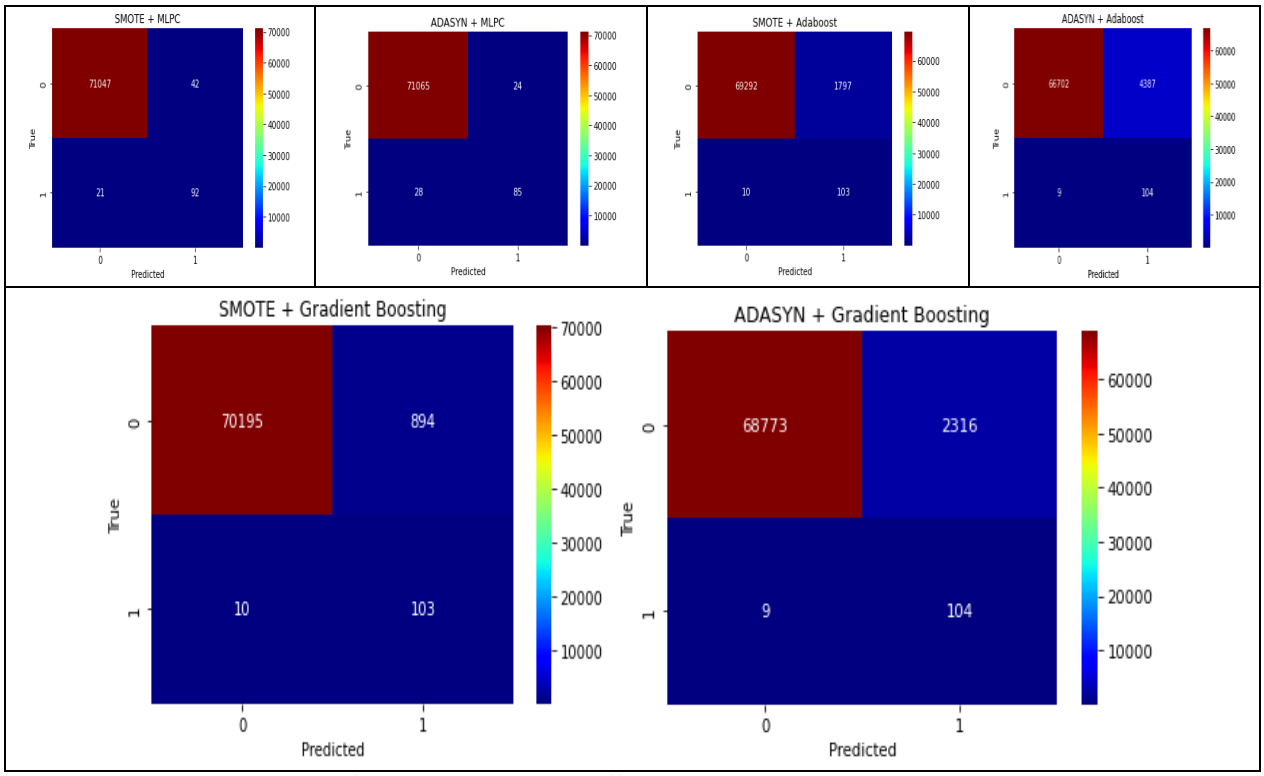
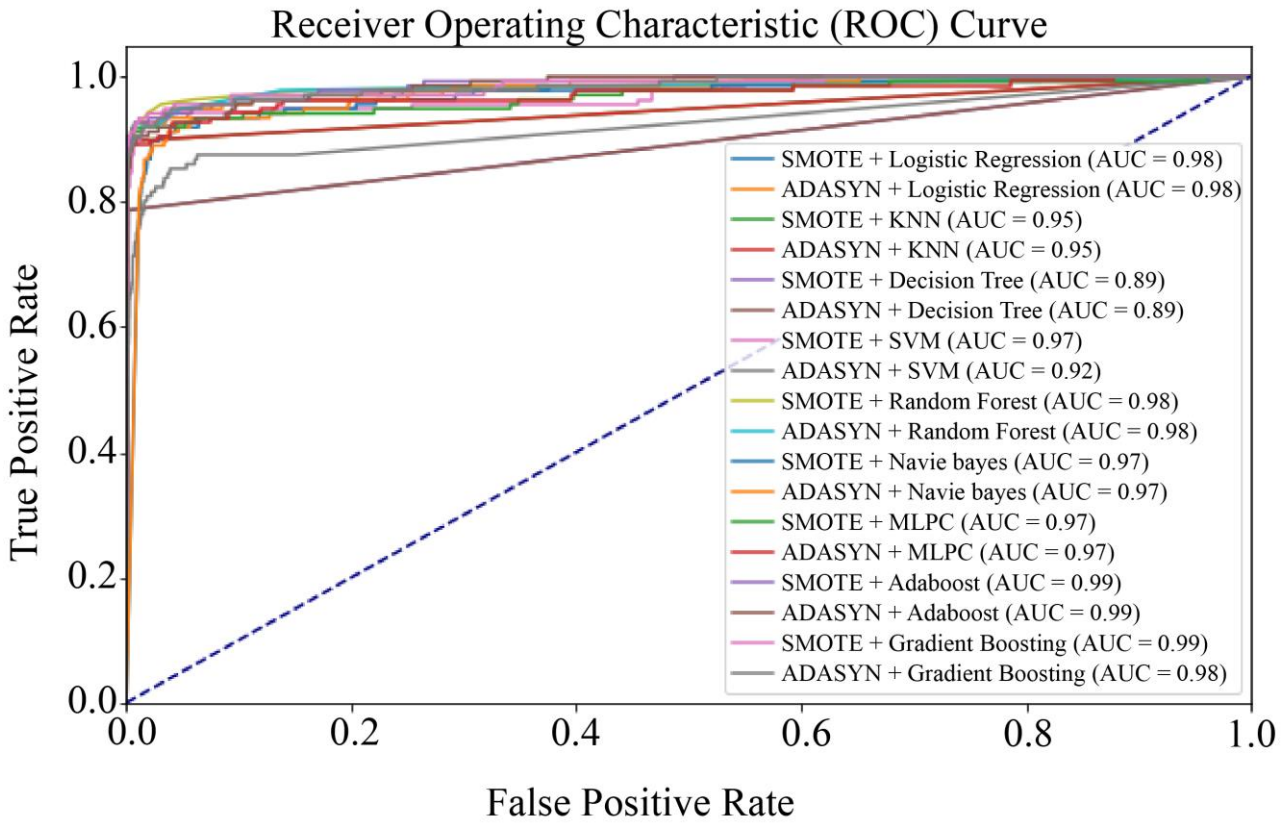
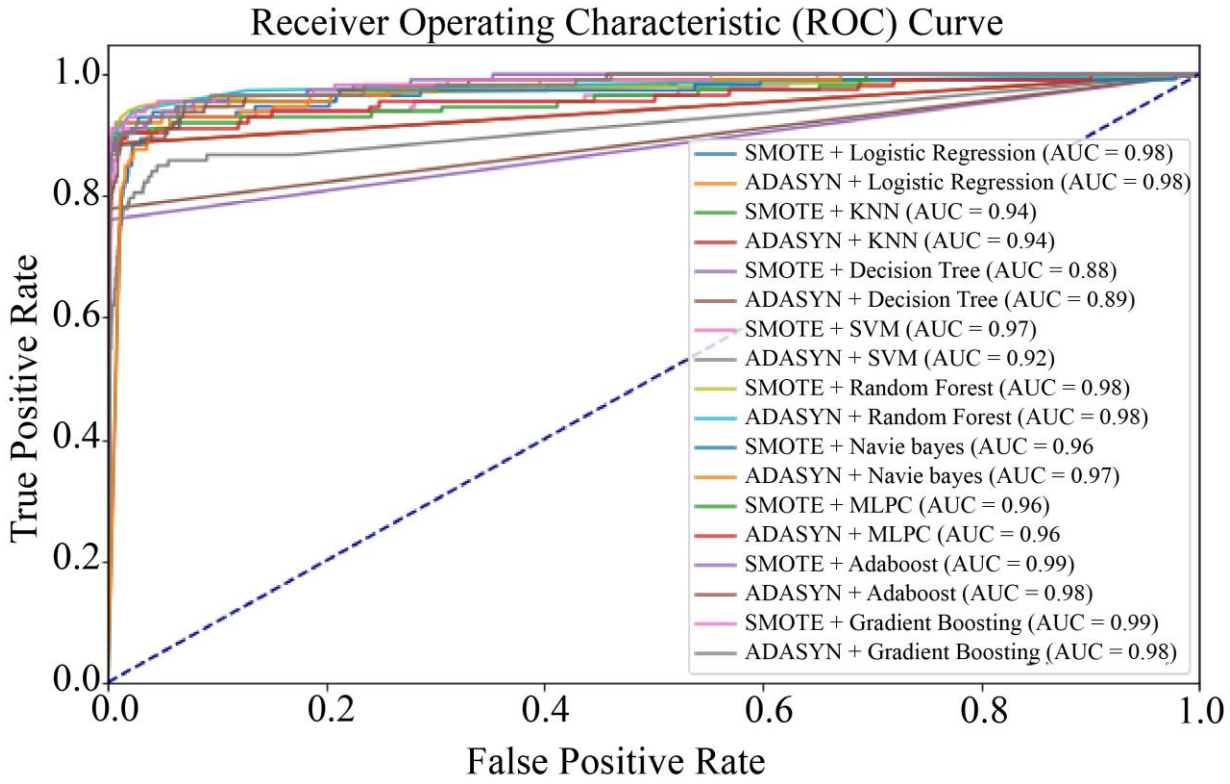


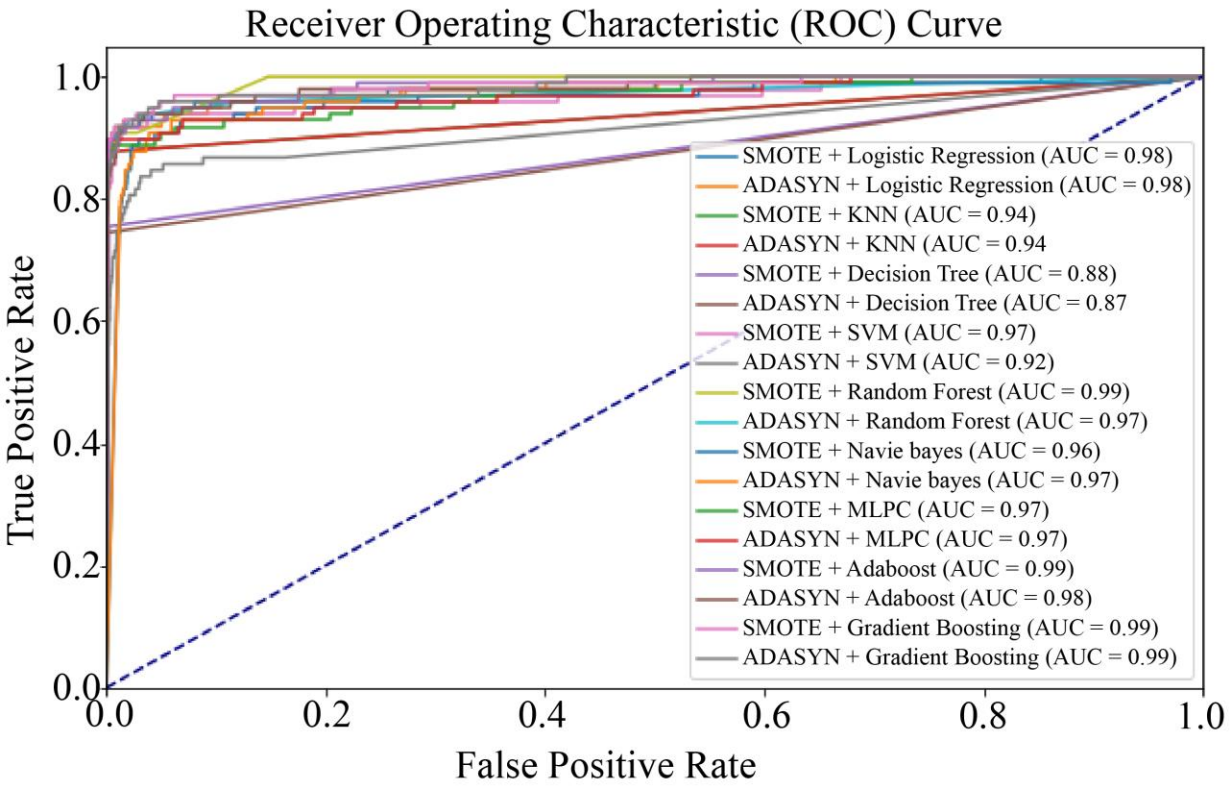

**Fig. 2 Confusion matrix considering different models using SMOTE and ADASYN**




**(a) Split ratio: 70:30**




**(b) Split ratio: 75:25**




**(c) Split ratio: 80:20**

**Fig. 3 AUC-ROC score based on the different split ratio**

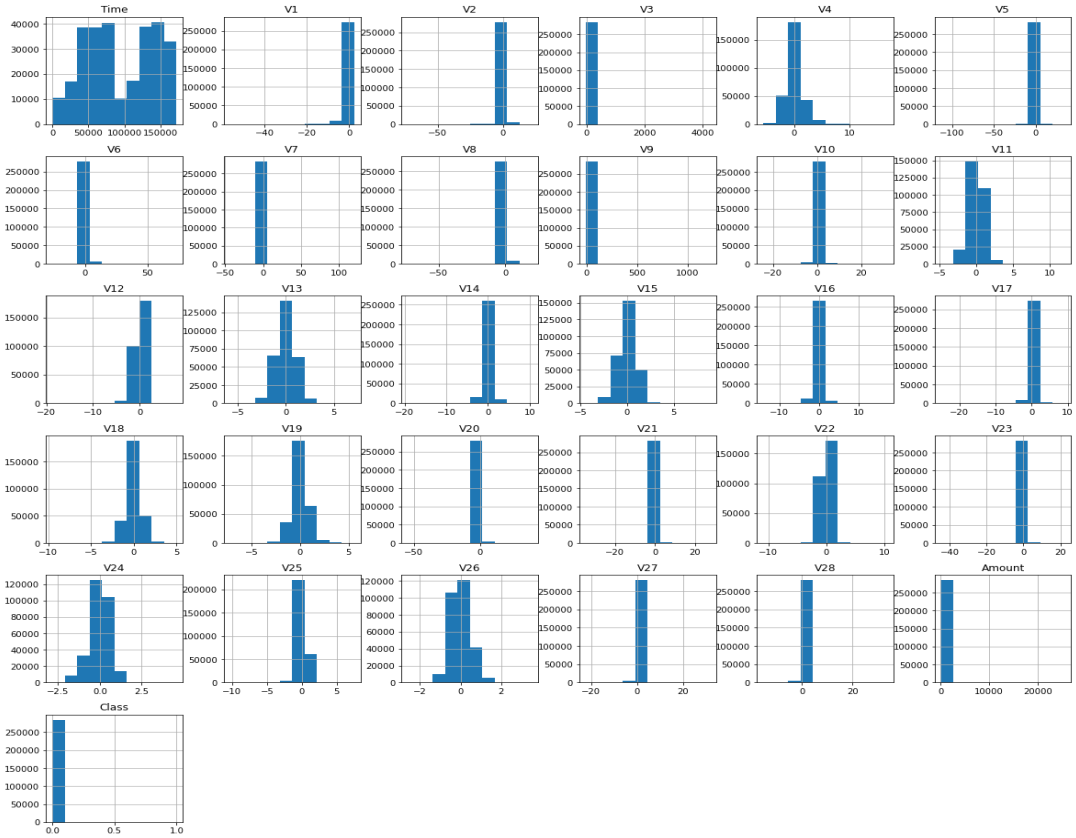



**Fig. 4 Matrix of histograms displaying variable distributions and pairwise relationships in a multivariate dataset**

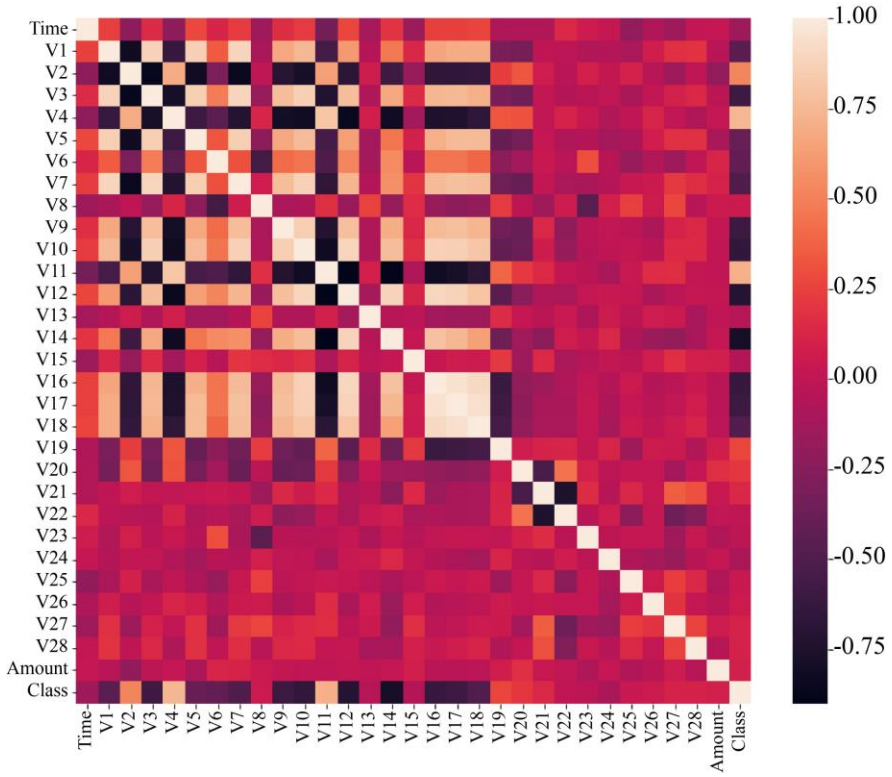



**Fig. 5 Correlation heatmap with color gradient legend indicating variable relationships in the dataset**

## 5. Discussion

The selected algorithms for this study—LR, kNN, SVM, DT, RF, AdaBoost, GB, MLP, and GNB—were strategically chosen to address the class imbalance problem and to cover a range of modeling approaches, from simple LR to complex ensemble and neural network models. The initial phase of data preprocessing, including the handling of missing values, encoding of categorical variables, and application of techniques such as SMOTE and ADASYN for class imbalance, was crucial to prepare the dataset for effective modelling.

Feature selection and dimensionality reduction were performed using LDA to retain the most pertinent features for fraud detection. The training of multiple models allowed the exploration of a spectrum of methodologies, culminating in the prediction phase, where the best-performing models were used to identify fraudulent transactions.

The evaluated models, including LR, kNN, DT, SVM, RF, GNB, MLP, Adaboost, and GB, were assessed based on various performance metrics. Among these, the RF model demonstrated superior performance, particularly in accuracy, specificity, and F1-score. With an 80:20 split using ADASYN, RF achieved outstanding results: an accuracy of 0.9995, a specificity of 0.9997, and an F1-score of 0.8528. The high accuracy and specificity suggest that RF is highly reliable in predicting both positive and negative classes, making it an ideal choice for imbalanced datasets.

The reason for RF's superior performance lies in its ensemble approach, which combines the predictions of multiple decision trees. Along with the use of SMOTE and ADASYN for addressing class imbalance and effective feature selection by LDA, it enhances its capability to deliver accurate and reliable predictions. This method effectively reduces overfitting, handles complex interrelations within large datasets, and provides more stable and accurate predictions.

Additionally, RF's ability to manage imbalanced data, where some classes are underrepresented, makes it particularly effective for tasks such as fraud detection, where fraudulent transactions are rare. The hybrid RF-based approach presented in this paper was found to be more efficient compared to the related work [42, 43]. With higher accuracy and precision, this hybrid approach outperforms previous methods [42, 43].

### *5.1. Limitations and Future Scope*

1. The dataset used in this study is limited to transactions made by European cardholders within a specific two-day period, which may not generalize to different regions or periods. Moreover, it contains a highly imbalanced class distribution, which, despite the utilization of SMOTE and ADASYN to mitigate this issue, still influences the performance and generalization of the models.
2. Some of the employed models, such as SVM and MLP, are inherently complex and require substantial computational resources for training and testing.
3. Fraudulent behavior evolves over time, and the models trained on historical data might not capture these changes. This study does not account for the temporal dynamics of fraud, which can result in decreased performance when the models are applied to more recent data.
4. Different attack types are not considered along with the different network environments like the Internet of Things [44, 45].

## 6. Conclusion

The study's findings confirm the effectiveness of machine learning ensembles in the domain of fraud detection within highly imbalanced datasets. By integrating a diverse set of algorithms, a robust approach was tailored to the unique distribution challenges posed by the data, reflecting real-world scenarios where fraudulent activities are rare but significantly impactful. Ensemble methods like RF emerged as superior, with high accuracy and specificity indicating their capacity to discern between classes effectively. The application of SMOTE and ADASYN oversampling methods proved critical, enhancing the predictive power of the models and counteracting the imbalance present in the dataset. Moreover, the employment of LDA for feature selection highlighted the importance of preprocessing in the machine learning pipeline, ensuring that the most relevant features were utilized for model training. While models like SVM showed precision, they also highlighted the computational expense associated with complex models, emphasizing the need for efficient yet powerful solutions in practical applications. This research advances the field of fraud detection by illustrating the substantial benefits of ensemble machine learning techniques, combined with appropriate preprocessing methods, in tackling class imbalance—a common and challenging issue in financial datasets. It offers a scalable and effective framework that can be adapted for similar problems in various domains, thereby contributing to the more reliable and efficient detection of fraudulent transactions.

## References

[1] Ayoub Mniai, Mouna Tarik, and Khalid Jebari, "A Novel Framework for Credit Card Fraud Detection," *IEEE Access*, vol. 11, pp. 112776-112786, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] Rafaël Van Belle, Bart Baesens, and Jochen De Weerdt, "CATCHM: A Novel Network-Based Credit Card Fraud Detection Method Using Node Representation Learning," *Decision Support Systems*, vol. 164, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[3] Yusuf Yusuf Dayyabu, Dhamayanthi Arumugam, and Suresh Balasingam, "The Application of Artificial Intelligence Techniques in Credit Card Fraud Detection: A Quantitative Study," *E3S Web of Conferences*, vol. 389, pp. 1-19, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] Zahra Salekshahrezaee, Joffrey L. Leevy, and Taghi M. Khoshgoftaar, "The Effect of Feature Extraction and Data Sampling on Credit Card Fraud Detection," *Journal of Big Data*, vol. 10, no. 1, pp. 1-17, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] Yuanming Ding et al., "Credit Card Fraud Detection Based on Improved Variational Autoencoder Generative Adversarial Network," *IEEE Access*, vol. 11, pp. 83680-83691, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] Sunil Gupta et al., "Authentication for Online Fraud Detection through Hidden Markov Model," *2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Noida, India, pp. 1-6, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Shivam Priyadarshi, and M. Adil Hashmi, "Cybersecurity Data Science and Threats: An Overview from Machine Learning Perspective," *ACCENTS Transactions on Information Security*, vol. 7, no. 25, pp. 1-8, 2022. [CrossRef] [Publisher Link]

[8] Daniele Lunghi et al., "An Adversary Model of Fraudsters' Behavior to Improve Oversampling in Credit Card Fraud Detection," *IEEE Access*, vol. 11, pp. 136666-136679, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[9] Jiajian Zheng et al., "The Credit Card Anti-Fraud Detection Model in the Context of Dynamic Integration Selection Algorithm," *Frontiers in Computing and Intelligent Systems*, vol. 6, no. 3, pp. 119-122, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[10] Maryam Habibpour et al., "Uncertainty-Aware Credit Card Fraud Detection Using Deep Learning," *Engineering Applications of Artificial Intelligence*, vol. 123, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Emilija Strelcenia, and Simant Prakoonwit, "A Survey on GAN Techniques for Data Augmentation to Address the Imbalanced Data Issues in Credit Card Fraud Detection," *Machine Learning and Knowledge Extraction*, vol. 5, no. 1, pp. 304-329, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] Honghao Zhu et al., "NUS: Noisy-Sample-Removed Undersampling Scheme for Imbalanced Classification and Application to Credit Card Fraud Detection," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 2, pp. 1793-1804, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[13] Huanjing Wang et al., "Enhancing Credit Card Fraud Detection through a Novel Ensemble Feature Selection Technique," *2023 IEEE 24th International Conference on Information Reuse and Integration for Data Science (IRI)*, Bellevue, WA, USA, pp. 121-126, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] B. Lebichot et al., "Assessment of Catastrophic Forgetting in Continual Credit Card Fraud Detection," *Expert Systems with Applications*, vol. 249, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] C. Victoria Priscilla, and D. Padma Prabha, "A Two-Phase Feature Selection Technique Using Mutual Information and XGB-RFE for Credit Card Fraud Detection," *International Journal of Advanced Technology and Engineering Exploration*, vol. 8, no. 85, pp. 1656-1668, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[16] Mimusa Azim Mim, Nazia Majadi, and Peal Mazumder, "A Soft Voting Ensemble Learning Approach for Credit Card Fraud Detection," *Heliyon*, vol. 10, no. 3, pp. 1-19, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[17] Kun Zhu et al., "An Adaptive Heterogeneous Credit Card Fraud Detection Model Based on Deep Reinforcement Training Subset Selection," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 8, pp. 4026-4041, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[18] Fatima Zohra El Hlouli et al., "Credit Card Fraud Detection: Addressing Imbalanced Datasets with a Multi-phase Approach," *SN Computer Science*, vol. 5, no. 1, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[19] Menglin Kong et al., "CFTNet: A Robust Credit Card Fraud Detection Model Enhanced by Counterfactual Data Augmentation," *Neural Computing and Applications*, vol. 36, no. 15, pp. 8607-8623, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[20] Vaman Ashqi Saeed, and Adnan Mohsin Abdulazeez, "Credit Card Fraud Detection using KNN, Random Forest and Logistic Regression Algorithms: A Comparative Analysis," *The Indonesian Journal of Computer Science*, vol. 13, no. 1, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[21] K.P. Bindu Madavi, and K. Krishna Sowjanya, *Credit Card Fraud Detection Using Big Data Analytics and Machine Learning*, 1st ed., Big Data Computing, CRC Press, pp. 1-15, 2024. [Google Scholar] [Publisher Link]

[22] Seema Garg, and Ritu Sharma, *Fraud Detection with Machine Learning and Artificial Intelligence*, 1st ed., Handbook of Artificial Intelligence Applications for Industrial Sustainability, CRC Press, pp. 1-10, 2024. [Google Scholar] [Publisher Link]

[23] Xiangrui Chao et al., "An Efficiency Curve for Evaluating Imbalanced Classifiers Considering Intrinsic Data Characteristics: Experimental Analysis," *Information Sciences*, vol. 608, pp. 1131-1156, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[24] G.S. Thejas et al., "An Extension of Synthetic Minority Oversampling Technique Based on Kalman Filter for Imbalanced Datasets," *Machine Learning with Applications*, vol. 8, pp. 1-12, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[25] Waleed Hilal, S. Andrew Gadsden, and John Yawney, "Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances," *Expert Systems with Applications*, vol. 193, pp. 1-34, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[26] Jenny Domashova, and Elena Kripak, "Development of a Generalized Algorithm for Identifying Atypical Bank Transactions Using Machine Learning Methods," *Procedia Computer Science*, vol. 213, pp. 101-109, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[27] Gang Kou, Hao Chen, and Mohammed A. Hefni, "Improved Hybrid Resampling and Ensemble Model for Imbalance Learning and Credit Evaluation," *Journal of Management Science and Engineering*, vol. 7, no. 4, pp. 511-529, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[28] Wonjae Lee, and Kangwon Seo, "Downsampling for Binary Classification with a Highly Imbalanced Dataset Using Active Learning," *Big Data Research*, vol. 28, pp. 1-19, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[29] Mohammed Temraz, and Mark T. Keane, "Solving the Class Imbalance Problem Using a Counterfactual Method for Data Augmentation," *Machine Learning with* Applications, vol. 9, pp. 1-16, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[30] Tie Li, Gang Kou, and Yi Peng, "A New Representation Learning Approach for Credit Data Analysis," *Information Sciences*, vol. 627, pp. 115-131, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[31] Asma Cherif et al., "Credit Card Fraud Detection in the Era of Disruptive Technologies: A Systematic Review," *Journal of King Saud University-Computer and Information Sciences*, vol. 35, no. 1, pp. 145-174, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[32] Bryan Karunachandra et al., "On the Benefits of Machine Learning Classification in Cashback Fraud Detection," *Procedia Computer Science*, vol. 216, pp. 364-369, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[33] Palak Gupta et al., "Unbalanced Credit Card Fraud Detection Data: A Machine Learning-Oriented Comparative Study of Balancing Techniques," *Procedia Computer Science*, vol. 218, pp. 2575-2584, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[34] Jonathan Kwaku Afriyie et al., "A Supervised Machine Learning Algorithm for Detecting and Predicting Fraud in Credit Card Transactions," *Decision Analytics Journal*, vol. 6, pp. 1-12, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[35] Teuku Rizky Noviandy et al., "Credit Card Fraud Detection for Contemporary Financial Management Using XGBoost-Driven Machine Learning and Data Augmentation Techniques," *Indatu Journal of Management and Accounting*, vol. 1, no. 1, pp. 29-35, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[36] Abdulaziz Saleh Alraddadi, "A Survey and a Credit Card Fraud Detection and Prevention Model Using the Decision Tree Algorithm," *Engineering, Technology & Applied Science Research*, vol. 13, no. 4, pp. 11505-11510, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[37] N. Prabhakaran, and R. Nedunchelian, "Oppositional Cat Swarm Optimization-Based Feature Selection Approach for Credit Card Fraud Detection," *Computational Intelligence and Neuroscience*, vol. 2023, no. 1, pp. 1-13, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[38] Emmanuel Ileberi, Yanxia Sun, and Zenghui Wang, "A Machine Learning Based Credit Card Fraud Detection Using the GA Algorithm for Feature Selection," *Journal of Big Data*, vol. 9, no. 1, pp. 1-17, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[39] Joffrey L. Leevy, John Hancock, and Taghi M. Khoshgoftaar, "Comparative Analysis of Binary and One-Class Classification Techniques for Credit Card Fraud Data," *Journal of Big Data*, vol. 10, no. 1, pp. 1-13, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[40] Hadeel Ahmad et al., "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. [CrossRef] [Google Scholar] [Publisher Link]

[41] Aya Abd El-Naby, Ezz El-Din Hemdan, and Ayman El-Sayed, "An Efficient Fraud Detection Framework with Credit Card Imbalanced Data in Financial Services," *Multimedia Tools and Applications*, vol. 82, no. 3, pp. 4139-4160, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[42] Indrani Vejalla et al., "Credit Card Fraud Detection Using Machine Learning Techniques," *2023 2nd International Conference on Paradigm Shifts in Communications Embedded Systems, Machine Learning and Signal Processing (PCEMS)*, Nagpur, India, pp. 1-4, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[43] Aditi Singh et al., "Design and Implementation of Different Machine Learning Algorithms for Credit Card Fraud Detection," *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, Maldives, Maldives, pp. 1-6, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[44] Anshika Sharma, and Himanshi Babbar, "Towards Resilient IoT Security: An Analysis and Classification of Attacks in MQTT-Based Networks" *2024 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT)*, Gharuan, India, pp. 122-125, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[45] Sonam Mittal et al., "Security of Internet of Things Based on Cryptographic Algorithm," *International Journal of Electronic Security and Digital Forensics*, vol. 16, no. 1, pp. 28-39, 2024. [CrossRef] [Google Scholar] [Publisher Link]