Heliyon 10 (2024) e25466

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

A soft voting ensemble learning approach for credit card fraud detection

Mimusa Azim Mim, Nazia Majadi *, Peal Mazumder

Department of Computer Science and Telecommunication Engineering, Noakhali Science and Technology University, Noakhali-3814, Bangladesh

ARTICLE INFO

Keywords:
Credit card fraud
Over-sampling
Under-sampling
Hybrid sampling
Ensemble learning techniques
Soft voting approach

ABSTRACT

With the advancement of e-commerce and modern technological development, credit cards are widely used for both online and offline purchases, which has increased the number of daily fraudulent transactions. Many organizations and financial institutions worldwide lose billions of dollars annually because of credit card fraud. Due to the global distribution of both legitimate and fraudulent transactions, it is difficult to discern between the two. Furthermore, because only a small proportion of transactions are fraudulent, there is a problem of class imbalance. Hence, an effective fraud-detection methodology is required to sustain the reliability of the payment system. Machine learning has recently emerged as a viable substitute for identifying this type of fraud. However, ML approaches have difficulty identifying fraud with high prediction accuracy, while also decreasing misclassification costs due to the size of the imbalanced data. In this research, a soft voting ensemble learning approach for detecting credit card fraud on imbalanced data is proposed. To do this, the proposed approach is evaluated and compared with numerous sophisticated sampling techniques (i.e., oversampling, undersampling, and hybrid sampling) to overcome the class imbalance problem. We develop several credit card fraud classifiers, including ensemble classifiers, with and without sampling techniques. According to the experimental results, the proposed soft-voting approach outperforms individual classifiers. With a false negative rate (FNR) of 0.0306, it achieves a precision of 0.9870, recall of 0.9694, f1-score of 0.8764, and AUROC of 0.9936.

1. Introduction

Online payment is becoming more common in all types of transactions today. The major reasons for this are the availability of credit cards and the growth of the e-commerce platform. There are 2.8 billion credit card users globally and 1.06 billion in the United States [1]. Online sales alone in the USA already make up 10 % of all retail sales and are estimated to increase by 15 % annually [2]. According to the Unisys protection index, Americans are significantly more concerned about credit and debit card theft than terrorism [3]. Credit card-based transactions have grown to be a vulnerable target for criminals, hackers, and thieves. It is not necessary to physically present the credit card while using it online; only the information of the card should be provided. One-Time Passwords (OTPs) sent via email are occasionally taken into consideration as an additional authentication step. A USA-based company called *Fidelity National Information Services* reports that just in April 2020, the dollar volume of attempted illegitimate transactions jumped by 35.8 %. Credit card fraud costs around $28.6 billion worldwide in 2020, with the United States having the highest number of incidents

* Corresponding author.
E-mail address: nazia_majadi@nstu.edu.bd (N. Majadi).

https://doi.org/10.1016/j.heliyon.2024.e25466
Received 12 August 2023; Received in revised form 27 December 2023; Accepted 27 January 2024
Available online 1 February 2024

and expected to reach $408 billion in transactions over the next ten years [4]. Therefore, it is essential to secure online transactions by effectively differentiating between fraudulent and authorized credit card transactions.

This type of real-time fraud detection issue can be effectively solved using machine learning. Big data, computational intelligence, machine learning, and deep learning technologies have been heavily invested in by numerous research and commercial groups to provide effective strategies for the problem [5]. However, a few common problems affect performance in general regardless of the technique used. For example, the training data typically has an imbalanced distribution. It characterizes prior transactions and causes various overfitting issues, resulting in poor performance of the adopted classifiers. This is due to the fact that the number of counterfeit samples available is frequently far less than the number of legitimate samples. In cases where the legitimate class is much larger than the fraudulent class, classifiers may have high accuracy in detecting legitimate transactions but struggle to accurately identify fraudulent transactions [6–8]. This results in a significant degree of imbalance, making it impossible to define a reliable evaluation model. As a result, analyzing and learning from unbalanced data is a difficult task.

The novelty of this study is to propose a soft voting ensemble learning-based approach for detecting credit card fraud in an unbalanced dataset. The proposed approach also assesses and compares several advanced under-sampling (i.e., random under-sampling), oversampling (i.e., SMOTE and ADASYN), and hybrid sampling (i.e., SMOTE-Tomek and SMOTE-ENN) techniques to deal with the imbalanced data. In total, sixty-six (66) credit card fraud classifiers (including fifty-five (55) using sampling and eleven (11) without sampling techniques) and eighteen (18) soft voting ensemble classifiers (including fifteen (15) using sampling, and three (3) without sampling techniques) have been developed. The optimal classifier demonstrates extremely high testing performance for identifying and classifying credit card fraud. Experimental results show that a soft voting ensemble learning approach with a combination of XGBoost, MLP, and KNN outperforms individual ML classifiers in terms of recall and FNR on the under-sampled dataset. The motivation behind this study lies in the fact that the performance of different machine learning classifiers and ensemble classifiers has not been previously investigated in the past for the credit card fraud detection challenge.The rest of the manuscript is organized as follows: Section 2 provides a succinct overview of the key studies on credit fraud detection that have been published in the past. Section 3 discusses the problem motivation. Section 4 emphasizes the proposed methodology for detecting credit card fraud. Section 5 presents an experimental setup and shows how ML classifiers, including ensemble learning techniques, perform on both imbalanced (i.e., without sampling) and balanced (i.e., with sampling) datasets. Finally, Section 6 provides concluding remarks with study strengths and limitations of the study and the possibilities for future research.

2. Literature review

Numerous methods have been proposed to detect fraudulent credit card transactions; one recent approach employed machine learning algorithms as a solution. These algorithms demonstrate their ability [9] and effectiveness in differentiating between genuine and fraudulent transactions [10]. There are two main methods for detecting fraudulent transactions. The first one employs supervised learning algorithms [11], which require labeling the dataset in order to identify patterns that distinguish fraudulent from non-fraudulent transactions. We identified Neural Networks [12], Fuzzy Logic [13], Particle Swarm Optimization [14], Regression Model [15], Genetic Algorithm [16], Naive Bayes [17], and Decision Trees [18] as a few frequently used supervised learning techniques. Ghosh and Reilly [19] suggested that K-Nearest Neighborhood (KNN) shows promising results on credit card fraud detection. The authors measured the performance of their proposed model based on factors such as specificity and sensitivity. However, the accuracy of the model is not satisfactory. This indicates the authors dealt with a significantly skewed dataset. Furthermore, Prodromidis and Stolfo [20] proposed a risk-based ensemble model to classify fraudulent and legitimate credit card users. Later, Stolfo et al. [21] used data mining techniques to counterfeit transactions on massive real-time data. The second technique detects fraudulent transactions using unsupervised learning algorithms [22]; this method utilizes unlabeled data to classify samples as fraudulent or authentic. We observed K-Means [23] and Self-Organizing Maps (SOMs) [24] as examples of this technique.

Moreover, Yu & Wang [25] implemented logistic regression to solve the classification problem. The authors discretized fraudulent cases using Gaussian mixture models (GMM) and used the synthetic minority over-sampling technique (i.e., SMOTE) for balancing a credit card fraud dataset. In another work, Ozcelik et al. [26] applied ML models using genetic algorithms for fraud detection. Next, Soltani et al. [27] provided a method for detecting fraudulent activities on both benchmarks and real-world data. In addition, Zareapoor et al. [28] focused on supervised methods for detecting credit card fraud. The authors compared numerous ML algorithms and illustrated how these algorithms behave differently under diverse conditions.

Besides, Vats et al. [29] proposed ML methods for identifying fraudulent credit card transactions. The authors discussed about how dealing with categorical data might be challenging when a valid transaction appears fraudulent or when a valid transaction appears fraudulent. However, a number of ML methods (including logistic regression, support vector machines, neural networks, and linear regression) would not operate with categorical data. Likewise, Patel and Singh [30] proposed a genetic algorithm-based approach to address the challenge of imbalanced classes in credit card fraud detection. Additionally, the use of the Random Forest (RF) algorithm to identify credit card fraud was investigated by Xuan et al. [31]. In the context of detecting credit card fraud, the authors most likely discussed their methodology, experimental findings, and insights into the potential advantages and limits of this strategy. A fundamental investigation employing RF to identify fraudulent transactions was also carried out by Kumar et al. [32].

Above and beyond this, some academics suggested artificial intelligence approach for classifying legitimate and fraudulent credit card users. For instance, Jain et al. [33] applied several AI techniques to counterfeit fraudulent transactions on a credit card. Furthermore, Carta et al. [34] applied a Prudential Multiple Consensus (PMC), Gaussian Naive Bayes (GNB), Random Forest (RF), Gradient Boosting (GB), and Adaptive Boosting (AB) algorithm to detect credit card fraud. The authors compared the PMC model with different performance metrics such as specificity, miss rate, sensitivity, fallout, and AUC. Moreover, Varmedja et al. [35] used logistic

regression (LR), RF, Naïve Bayes (NB), multilayer perceptrons, and an artificial neural network (ANN). Then, Puh and Brkić [36] studied the performance of different algorithms, namely RF, Support Vector Machine (SVM), and LR, in detecting credit card fraud. The authors (Varmedja et al., [35]; Puh & Brkić, [36]) solved the class imbalance problem in the data set using the composite minority oversampling (SMOTE) technique. Furthermore, John and Naaz [37] used both the local outlier factor (LOF) and isolation forests to detect fraudulent transactions. However, the authors did not address the unbalanced class problem in the dataset. As well, Najadat et al. [38] employed a bidirectional long short-term memory (BiLSTM) and bidirectional gated recurrent unit (BiGRU)-based model to identify fraudulent cases in the credit card fraud dataset.

Also, some researchers applied deep learning techniques to detect fraud countermeasures in credit card transactions. For instance, Van et al. [39] applied a technique for automatically identifying credit card theft in online stores based on client spending patterns. Moreover, Kumar et al. [40] developed a method for predicting fraud and legitimate transactions in terms of time and money based on ML algorithms and statistics. The authors also applied calculus and linear algebra in the construction of advanced machine learning models. In addition, Khatri et al. [41] analyzed various supervised learning models to identify fraudulent credit card transactions. In another work, Taha and Malebary [42] proposed a method for detecting credit card fraud using light gradient boosting machine (LightGBM). Additionally, Vengatesan et al. [43] tested the performance of LR and KNN on an unbalanced credit card fraud dataset, and Hema [44] used RF, LR, and category boosting (CatBoost) to identify credit card fraud. Additionally, Asha and KR [45] proposed an approach utilizing SVM, KNN, and ANN models to identify credit card fraud. However, none of the authors (Khatri et al., [41]; Taha and Malebary, [42]; Vengatesan et al., [43]; Hema, [44]; Asha and KR, [45]) mentioned the issue of class imbalance. Some researchers used the differential evolution hyperparameter optimization approach to identify fraudulent credit card transactions, differential evolution (DE) algorithm to address the issue of data imbalance, and optimized XGBoost algorithm to categorize fraudulent transactions [46]. Kafhali and Tayebi [47] developed an effective credit card fraud detection solution by integrating Differential Evolution for hyperparameter selection in XGBoost, addressing imbalanced data with SMOTE and ENN. Their optimized XGBoost algorithm demonstrated superior performance, achieving 99.94% accuracy, 80.68% precision, 86.02% recall, 83.27% F-measure, and a 99.21% AUC score, surpassing other machine learning models in this study. Kafhali and Tayebi [48] also proposed a novel oversampling technique leveraging generative adversarial neural networks to address imbalanced datasets, outperforming established methods like SMOTE, Random Oversampling (ROS), and ADASYN. Their approach demonstrated superior performance in handling imbalanced issues in a real-world European credit card dataset when evaluated against three machine learning algorithms. Ranjit Panigrahi et al. [49] suggested a host-based intrusion detection method to solve the problem of imbalanced intrusion detection dataset. In the preprocessing phase, they initially employed an enhanced random sampling technique to generate balanced samples from highly unbalanced data sets. Subsequently, an improved multiclass feature selection process was used to filter the datasets. In the final stage, a merged tree construction technique built on a detector based on C4.5 was developed. According to the experimental findings, their suggested approach obtains high detection accuracy.

In addition, there had been a number of research studies conducted on ensemble learning techniques. For example, Wang and Han [50] presented a model to anticipate credit card fraud based on integrated SVM and cluster analysis. Moreover, Bhanusri et al. [51] and Sellam et al. [52] executed different ML algorithms such as LR, NB, and RF with ensemble classifiers on an imbalanced dataset. Furthermore, Alfaiz and Fati [53] proposed a two-step credit card fraud detection method. The first step names the top three machine learning algorithms out of the nine. The second stage integrates the three best algorithms with nineteen resampling techniques. Each model in both phases was evaluated based on the AUROC curve, accuracy, recovery, precision, and F1 score. Padhi et al. [54] implemented six boosting techniques, that is, XGBoost, AdaBoost, Gradient Boosting, LightGBM, CatBoost, and Histogram-based Gradient Boosting, which were hybridized using a stacking framework to predict stock market direction across various datasets from different countries. Employing overfitting protection and evaluating with multiple metrics, the study suggests Meta-LightGBM as a promising predictive model with minimal training and testing accuracy differences, potentially offering investors a tool for risk control and short-term, sustainable profits. Last but not least, Nandi et al. [55] designed an ensemble multi-classifier system (MCS) model incorporating the behavior-knowledge space (BKS) for identifying credit card fraud. The authors measured the performance of their proposed approach with majority voting on publicly available real-world financial data sets. However, the issue of class imbalance in the dataset was not addressed by the authors.

3. Problem motivation

Considering the limitations mentioned in Section 2, the main contribution of this study is to propose a soft voting ensemble learning approach for the efficient detection of fraudulent transactions. Furthermore, this study investigates the challenges of dealing with an imbalanced dataset of credit card fraud countermeasures. The research objectives are as follows.

- To investigate and analyze various machine learning algorithms to identify credit card fraud as accurately as possible.
- To evaluate the proposed model by grouping the selected ML models into triples, known as the "*soft voting ensemble learning approach*".
- To achieve high prediction accuracy in fraud detection while concurrently minimizing misclassification costs.
- To perform a comprehensive comparative analysis to determine the most effective classifier for credit card fraud detection.

4. Materials and methods

In this study, a soft voting ensemble learning approach is presented for identifying fraudulent cases in credit card transactions. The

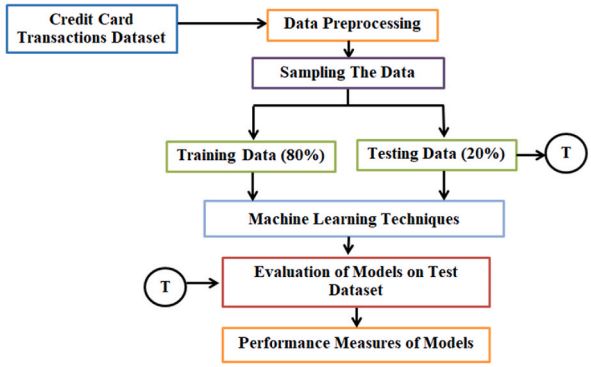


Fig. 1. System architecture for credit card fraud detection using ML techniques.

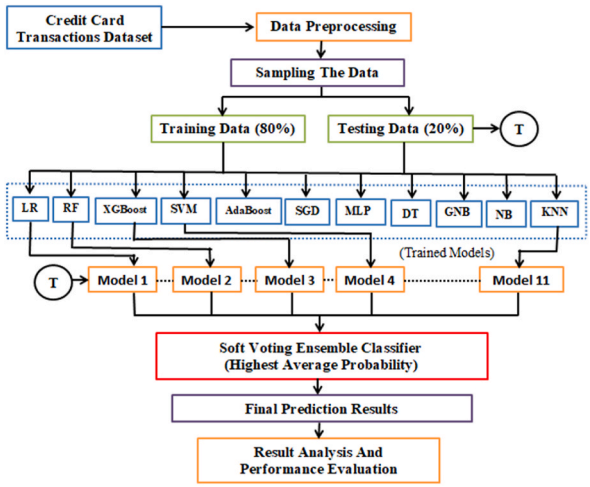


Fig. 2. System Architecture of the proposed soft voting ensemble learning approach.

approach is anticipated to be able to evaluate credit card transactions and determine whether they are legitimate or fraudulent. This section outlines the structure of the proposed system for detecting credit card fraud, including system architecture, feature scaling, sampling and scaling data, as well as ML algorithms.

4.1. System architecture

Fig. 1 presents the proposed system architecture for detecting credit card fraud using the ML models. First, the credit card dataset is preprocessed (i.e., class sampling and scaling data using standardization and normalization) in the data preprocessing phase. To detect fraudulent transactions, a set of experiments is performed to observe which one is best. Several ML classification algorithms have been applied to the original credit card fraud dataset. Furthermore, random undersampling, oversampling (e.g., SMOTE, ADASYN), and hybrid sampling (e.g., SMOTE-Tomek, SMOTE-ENN) techniques are used to deal with an imbalanced dataset. The steps involving the sampling methods are described in Section 3.3.

Moreover, the proposed model is tested by combining the predictions of multiple classifiers, which is known as the 'soft voting ensemble learning' approach. In order to achieve the highest classification accuracy possible, it enables greater flexibility in the combination of strategies. The three independent models with the highest performance levels are combined using various methodologies to find the most viable combination. Combining more than three of the best-performing individual models adds time complexity while producing the same results [56]. The soft voting strategy initially assigns weights to each base classifier. It generates prediction probabilities for each test sample belonging to various classes during the testing phase. Later, these probabilities are multiplied with the weights assigned to each class label, and then they are averaged. Ultimately, test samples are grouped into the class with the highest average probability. Data samples are mathematically categorized by the soft voting technique as the *argmax* (argument of maxima) of the sum of assigned probabilities. The system architecture of the soft voting ensemble approach used in this study to detect credit card fraud is shown in Fig. 2.

The pseudo-code of the proposed soft voting ensemble learning approach for identifying credit card fraud is presented in the following algorithm.

Algorithm 1
A soft voting ensemble learning approach for credit card fraud detection

```
Input: The credit card fraud dataset
    Output: Prediction result (Legitimate/Fraud)
Let's the whole dataset consists of i instances and X features.
    The class variable is Y so labels_Y_i = [2]
    Function F: X→ labels_Y_i
    Procedure Split_data (dataset):
    #Split the dataset into training and testing data
      Training_data, Testing_data = split (dataset)
    Procedure datasampling (dataset):
    # Apply sampling technique to the dataset
        return sampled_dataset
    C1 = LR (Training_dataset, Testing_data)
    C2 = RF (Training_dataset, Testing_data)
    C3 = XGB (Training_dataset, Testing_data)
    C4 = SVM (Training_dataset, Testing_data)
    C5 = AdaBoost (Training_dataset, Testing_data)
    C6 = SGD (Training_dataset, Testing_data)
    C7 = MLP (Training_dataset, Testing_data)
    C8 = DT (Training_dataset, Testing_data)
    C9 = GNB (Training_dataset, Testing_data)
    C10 = GB (Training_dataset, Testing_data)
    C11 = KNN (Training_dataset, Testing_data)
    Procedure ensemble_model (Training_dataset, Testing_data):
    # Create instances of the three top-performing individual models
    model1 = C2 # For example, Random Forest
    model2 = C3 # For example, XGBoost
    model3 = C7 # For example, MLP
    # Fit each individual model on the training data
      model1.fit(Training_dataset, Testing_data)
      model2.fit(Training_dataset, Testing_data)
      model3.fit(Training_dataset, Testing_data)
    # Assign weights to models based on their performance with validation data
    weight_model1 = 0.4
    weight_model2 = 0.4
    weight_model3 = 0.2
    # Create a soft voting ensemble with the three models and weights
    ensemble = VotingClassifier(estimators=[
        ('model1', model1),
        ('model2', model2),
        ('model3', model3)
      ], voting='soft', weights=[weight_model1, weight_model2, weight_model3])
    # Fit the ensemble on the training data (no need to fit individual models again)
    Ensemble.fit (Training_dataset, Testing_data)
    # Make predictions using the ensemble
    Predictions = ensemble.predict (Testing_data)
    return Predictions
    # Usage of the ensemble model
    Sampled_data = datasampling(dataset)
    Training_data, Testing_data = Split_data(Sampled_data)
    predictions = ensemble_model(Training_data, Testing_data)
```

4.2. Feature scaling

Feature scaling is one of the most essential stages in the preprocessing of data before building a machine learning model [57]. The performance of a machine learning model heavily relies on proper scaling, as it impacts the separation between data points. For instance, in distance-based measurements, unscaled features with a larger value range can dominate the analysis [58]. Algorithms that demand quick convergence, such as neural networks, often necessitate feature scaling [59]. In this study, the *StandardScaler* method [60] was employed to rescale the credit card transaction dataset. This process standardizes each column to have a mean of 0 and a standard deviation of 1 by subtracting the mean and dividing by the standard deviation. Typically, the training dataset encompasses 80% of the data, with the remaining 20% allocated to the testing dataset. These datasets are constructed using random sampling, where the dataset is shuffled, and 80% of the instances are randomly assigned to the training dataset, while the remaining 20% form the test/validation dataset. This randomization mitigates potential biases stemming from data ordering or patterns in the original dataset, ensuring the representativeness of both datasets. Subsequently, the model's performance is assessed on the testing dataset, which

contains unseen data, providing an estimate of how well the model will perform on new, unseen instances.

4.3. Sampling of credit card fraud data

Imbalanced data typically alludes to situations when the classification of classes is not equally divided. When learning from skewed data, predictive performance suffers [61]. Furthermore, due to baseline classifiers' preference for the majority class, the minority class (typically the class of interest) in the imbalanced dataset would be incorrectly categorized. A serious concern in the domain of fraud detection is a screwed-up class distribution. And the possible explanation is that the fraudulent class seems to be misclassified as "normal". Furthermore, classifiers that learn to always estimate the majority class could acquire 99 % accuracy. However, such classifiers are inadequate for identifying the fraud class. Nevertheless, while the fraud class has the highest cost of misclassification, classifiers often have poor accuracy when detecting fraud classes.

The credit card transaction dataset is highly imbalanced, with legitimate cases at 284,315 and fraud cases at 492 (refer to Section 5.1). There are numerous approaches available to overcome imbalanced learning problems, which are classified into two groups [62]: (i) data-level approach (data sampling), and (ii) algorithm-level approach (cost-sensitive learning). The data-level technique rebalances the distribution of classes by adjusting the class ratio. On the contrary, the algorithm-level method provides the classes with different weights (i.e., a higher cost to the class of interest). We emphasize data sampling techniques that increase data for the minority class (over-sampling), decrease data from the majority class (under-sampling), or do both (hybrid sampling). Hybrid methods combine the advantages of the two previous methods [63].

4.3.1. Under-sampling techniques

The under-sampling technique reduces the quantity of observations from the majority class to have less of an impact on ML algorithms while creating a balanced dataset [64]. When the dataset is massive, the approach performs best. Additionally, reducing the amount of training data eliminates the storage issue while also improving runtime. However, it may remove some significant instances from the dataset while decreasing the number of instances.

The random under-sampling strategy identifies random instances from the majority class and eliminates them from the training dataset in such a way that the majority and minority class ratios become 1:1 [65]. For instance, there are 492 minority instances and 284,315 majority instances in the credit card transaction dataset. We reduce it to 394 to maintain a 1:1 ratio by randomly deleting the majority class.

4.3.2. Over-sampling techniques

The over-sampling strategy increases the quantity of data available to the minority class, which has a greater impact on ML algorithms [66]. To balance the data, it duplicates the observations from the minority class. It is sometimes referred to as "up-sampling." Oversampling has the advantage of causing no information or data loss. However, it could duplicate observations in the dataset, which eventually results in overfitting the dataset [67]. There are several sophisticated over-sampling methods, such as SMOTE and ADASYN.

(a) **Synthetic Minority Oversampling Technique (SMOTE):** SMOTE is an oversampling approach that uses the K-Nearest Neighbor algorithm to produce synthetic data [68]. The overfitting problem due to random oversampling is reduced with the help of this technique. Through the use of interpolation between the positive instances that are close together, it concentrates on the feature space to produce new instances [69].

(b) **Adaptive Synthetic Sampling Approach (ADASYN):** ADASYN generates synthetic data based on the density of the data [70]. The density of the minority class has an inverse relationship with the development of synthetic data [71]. In low-density minority-class areas, a disproportionately larger amount of synthetic data is generated than in higher-density areas. In other words, more synthetic data is created in the less dense areas of the minority class [72].

4.3.3. Hybrid sampling techniques

The classification techniques could not be directly implemented on the skewed dataset [73]. Hence, the unbalanced dataset must be converted into a balanced dataset. It is possible to achieve this through the use of hybrid sampling. If classification models are performed precisely on an unbalanced dataset in which samples from one class are much larger than samples from another class, then the predictions may be biased more towards the majority class [74]. In this scenario, the predicted outcome could be inaccurate.

An imbalanced dataset can be converted into a balanced dataset using a variety of techniques. Hybrid data sampling is one of the most widely used techniques. There are two techniques available combining over and under-sampling, which makes them hybrid methods [75], such as SMOTE-Tomek and SMOTE-ENN.

(a) **Synthetic Minority Oversampling Technique and Tomek (SMOTE-Tomek):** SMOTE method generates random samples by inserting additional points within marginal outliers and inliers [76]. This problem can be fixed by clearing the space left behind after oversampling. Regarding this, Tomeklink is used to clean the space [77].

(b) **Synthetic Minority Oversampling Technique and Edited Nearest Neighbor (SMOTE-ENN):** SMOTE is a well-known oversampling technique that could be combined with some under-sampling approaches. The ENN approach is a prominent under-sampling strategy. The concept of SMOTE-ENN is over-sampling using SMOTE and cleaning using ENN.

To understand the concepts of Tomek and ENN, let us consider an example. Assume that two data points a and b are from separate classes, and m (a, b) is the distance between a and b. If there is no data c, such that m (a, c) < m (a, b) or m (b, c) < m (b, a), the pair (a, b) is a Tomek. If two data samples form a Tomek, then one of them is either noise or borderline. In this approach, instead of merely eliminating data from Tomek's majority class, data from both classes is eliminated. On the contrary, ENN eliminates data whose class differs from the majority of its k-nearest neighbors and continues to remove data until the remaining dataset is at a minimum [78]. ENN removes more occurrences than Tomek. Consequently, it is thought to provide more comprehensive data cleaning in depth.

4.4. Machine learning algorithms

There are several machine learning algorithms available to detect counterfeit fraud in a particular domain. For our proposed credit card fraud detection model, we choose the following algorithms.

4.4.1. Logistic regression (LR)

Logistic regression (LR), a supervised classification technique, returns the probability of a binary dependent variable that is estimated from an independent variable in the dataset. It is a regression model that explores the interaction between several independent variables and has a categorical dependent variable. The probability of an outcome with the two possible values of *zero* or *one*, *yes* or *no*, and *false* or *true* is predicted using LR. There are several LR models available, including binary logistic, multiple logistic, and binomial logistic.

LR is different from linear regression. For instance, linear regression yields a straight line, whereas LR displays a curve. Furthermore, LR generates logistic curves that depict values between *zero* and *one* depending on the number of predictors or independent variables used.

4.4.2. Random forest (RF)

Random Forest (RF) is one of the most frequently used ML algorithms in both developed models and real-world instances. It randomly selects features that are independent variables. Additionally, the rows are chosen at random using row sampling, and hyperparameter optimization is used to calculate the size of the decision tree. The outcome of a classification problem is the maximum occurrence output from each Decision Tree (DT) model inside the RF. The root node is generated randomly in RF. This is the fundamental difference between random forest and the traditional DT algorithm.

4.4.3. eXtreme gradient boosting (XGBoost)

XGBoost has been extensively utilized in many domains to obtain state-of-the-art outcomes on various data challenges. It is a very efficient and scalable ML algorithm for tree boosting. The basic idea of "boosting" is to merge a sequence of weak classifiers with low accuracy to develop a strong classifier with improved classification performance.

4.4.4. Support vector machine (SVM)

SVM is a prominent supervised learning technique for analyzing data used for classification and regression. SVM modeling consists of two steps: first, training a dataset to produce a model; and second, using this model to predict information from a testing dataset. SVM is a discriminative classifier that may be expressed theoretically by a separating hyperplane. The model represents the training data points as points in space, and the mapping is then performed to partition the points that belong to distinct classes by as large a distance as possible. New data points are mapped onto the same space, and their location inside the gap is predicted.

4.4.5. Adaptive boosting (AdaBoost)

Boosting is an ensemble modeling strategy that aims to generate a strong classifier from a collection of weak ones. In this method, models are added repeatedly until either the whole training data set is correctly predicted or the maximum number of models is reached. AdaBoost was the first effective boosting technique designed for binary classification. It is an abbreviation for "adaptive boosting," which is a prominent boosting strategy that combines numerous poor classifiers into a single effective classifier.

4.4.6. Stochastic gradient descent (SGD)

SGD is a popular optimization technique in ML applications for determining model parameters that correspond to the best fit between expected and actual outputs. The term "stochastic" refers to a system or process connected with a random probability. SGD is an iterative approach to improving the smoothness of an objective function. For each iteration, SGD takes a few samples at random rather than the whole data set.

4.4.7. Multilayer perceptron (MLP)

MLP is a type of feedforward artificial neural network that generates a set of results based on a set of inputs. MLP is used to refer to any transitional ANN, but it is also used explicitly to refer to networks consisting of multiple layers of perceptrons. MLP is characterized by many layers of input nodes connected as directed graphs between the input and output layers.

4.4.8. Decision tree (DT)

DT is applied to the applications of classification and regression. The working method is the same for both, although certain formulas differ. Entropy and information gain are used to build the DT model for classification issues. Entropy indicates how randomly

Table 1
Dataset details.

Transactions	Legitimate	Fraudulent	Features	Classes
284,807	284,315	492	30	2

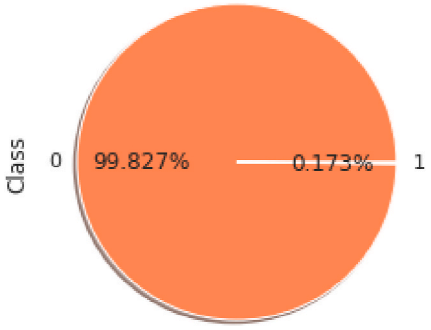


Fig. 3. Analysis of valid and fraud transactions (0: valid and 1: fraud).

data is distributed, whereas information gain reveals how much knowledge we might get from a given attribute or feature. The DT model for regression is built using the *Gini* and *Gini indexes*. When solving classification problems, the root node is selected based on information gain, which prioritizes nodes with high information gain and low entropy. Furthermore, the feature with the least *Gini* is preferred as the root when solving regression problems. The hyperparameter optimization approach is used to compute the depth of the tree using a grid search cross-validation procedure.

4.4.9. Gaussian Naive Bayes (GNB)

GNB is a variant of Naive Bayes (NB) that applies the Gaussian normal distribution and works with continuous data. NB is a group of supervised ML classification algorithms based on the Bayes theorem. It is a simple classification method with high functionality. GNB accepts continuous-valued features and models each as a Gaussian (normal) distribution.

4.4.10. Gradient boosting (GB)

GB is a greedy approach that can quickly overfit a training dataset. GB is a machine learning boosting technique that represents a decision tree for vast and complicated data. As we all know, ML algorithm errors are widely categorized into two types: bias errors and variance errors. As one of the boosting strategies, GB is used to decrease the model's bias error. GB can predict both continuous target variables (as a regressor) and categorical target variables (as a classifier). The cost function is Mean Square Error (MSE) when it is used as a regressor, but Log loss when it is used as a classifier.

4.4.11. K-nearest neighbor (KNN)

KNN is a straightforward technique that reserves all existing instances and identifies new instances based on the majority vote of its K-neighbors. The instance is assigned to the class because it has the highest frequency of occurrence among its KNN, as determined by a distance function.

4.4.12. Ensemble techniques

The ensemble technique is a general meta-approach in machine learning that creates numerous models and then combines them to achieve the best outcomes [79]. Ensemble approaches often generate more accurate results than a single model. Boosting, bagging, and stacking are the most prevalent ensemble techniques. Ensemble techniques are effective for regression and classification because they decrease bias and variance while improving model accuracy [80].

5. Experimental findings, comparative analysis, and discussion

This section describes how the proposed model performs on the credit card dataset to identify fraudulent transactions.

5.1. Dataset description

We apply the proposed credit card fraud detection approach to a publicly available, processed real-world dataset [81]. The dataset consists of a collection of credit card transactions made by European cardholders on two separate days in September 2013. Andrea Dal Pozzolo and his colleagues collected and evaluated the data as part of research cooperation between World Line and the Machine Learning group at ULB (University Libre de Bruxelles) on big data mining and fraud detection [82].

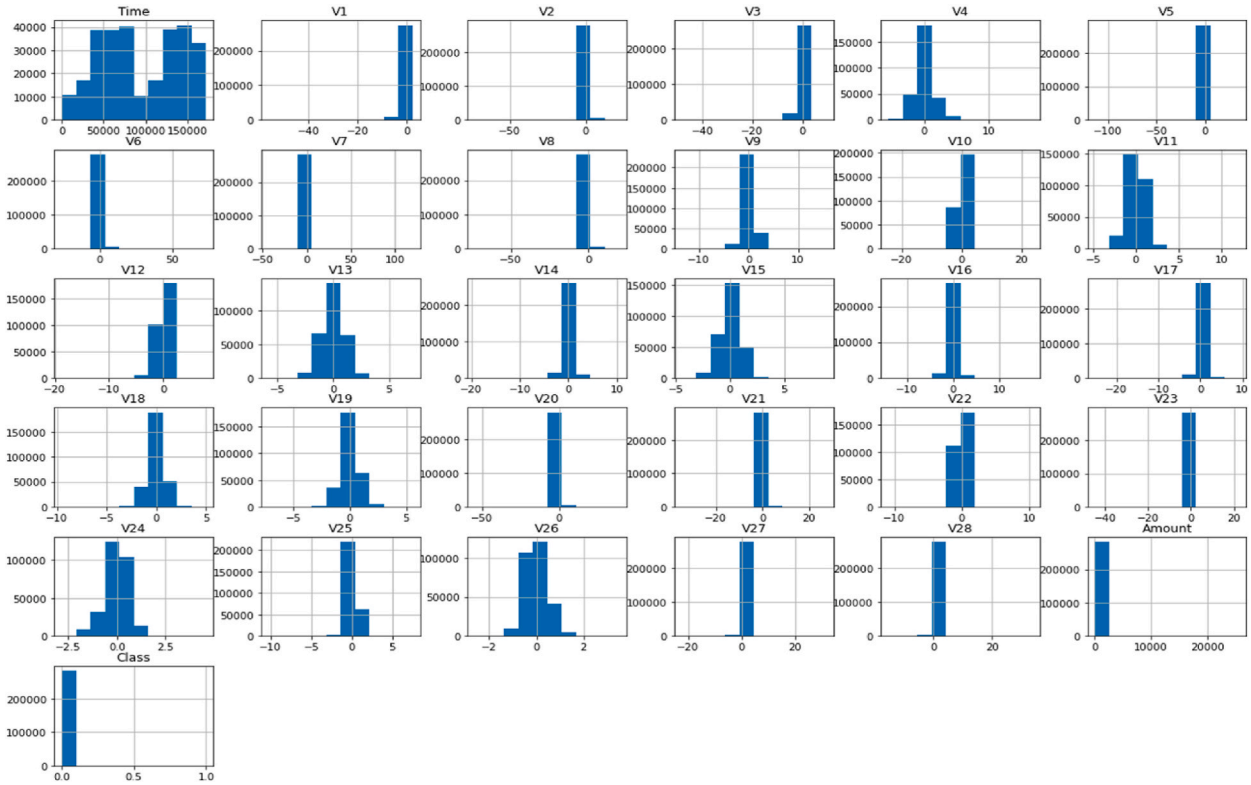


Fig. 4. Histogram for feature time and other features (e.g., *v*1 to *v*28).

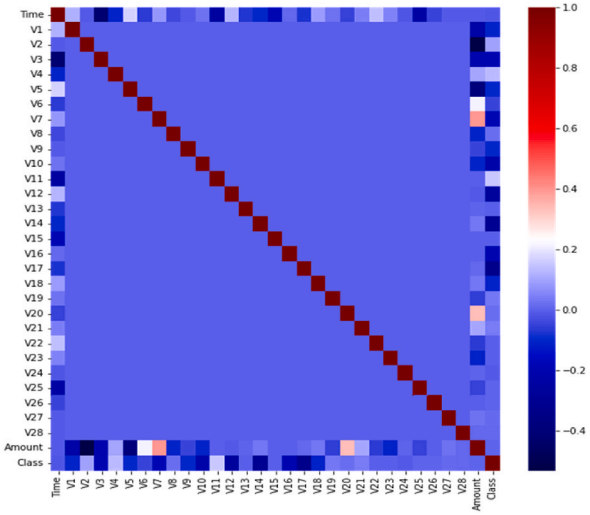


Fig. 5. Heatmap.

In the dataset, there are 492 frauds out of 284,807 transactions that occurred in the last two days of September 2013, as shown in Table 1. The dataset is highly imbalanced, with the positive class (fraud) accounting for just 0.173 % of all transactions (see Fig. 3). There are in total 31 variables that have been analyzed, with three variables unaltered (i.e., *Time*, *Amount*, and *Class*) and the remaining 28 variables provided as *V*1, *V*2, …, and *V*28 with their altered values using Principal Component Analysis (PCA) due to confidentiality concerns. *Time* presents the distance between the initial transaction and the following transaction, as well as any time limits or variations between the two. *Amount* is the amount of money used during the transaction. *Class* has two values: 0 and 1; 0 signifies a genuine transaction, whereas 1 denotes a fraudulent one.

Fig. 4 shows the histogram of each column in the dataset, and Fig. 5 represents a heatmap indicating a strong correlation matrix between different variables in the dataset. Furthermore, it shows the characteristics that are essential for the overall categorization. We observe that some of the feature values are nearly zero. This means there is no strong relationship found between different *V* parameters. Moreover, the important thing is to focus on the *Class* attribute of the dataset. In Fig. 5, the lighter blue indicates a positive correlation, whereas the deep blue represents a strong negative correlation. For example, *V*11 would be a stronger positive correlation, whereas *V*17 would be a stronger negative correlation.

5.2. Performance metrics

Since it has a considerably higher misclassification cost, the present study places emphasis on the suspicious class (i.e., fraud) in the fraud detection domain. Many performance metrics are employed to evaluate the performance of the proposed model. It is inappropriate to evaluate the performance of the model only by accuracy, as the dataset utilized in this study is highly skewed. Hence, the following quality measurements (including accuracy) are considered in the present study to obtain further insights into the results given by each classifier.

- **Accuracy:** The accuracy of a classifier determines how correctly the classifier categorizes the training tuple. Predicting the class label of tuples is the purpose of this measurement, and the testing sets of the classifier estimate its accuracy. An accuracy value is generated by dividing the number of observations that were successfully predicted by the total number of observations. It is the most intuitive performance indicator. Accuracy is measured by using the following formula, Eq. (1):

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

- **Precision:** Precision is the number of True Positives divided by the number of True Positives (TP) and False Positives (FP). The percentage of tuples classified as positive depends on precision, which is a measure of exactness. In fraud detection, precision is important because it indicates how many of the predicted fraud cases are actually fraud. Precision is measured by the following formula, Eq. (2):

$$\text{Precision (P)} = \frac{TP}{TP + FP} \ (0 < P < 1) \tag{2}$$

- **Recall:** Recall is the number of True Positives (TP) divided by the total number of True Positives (TP) and False Negatives (FN). Recall is also known as the "True Positive Rate" or "Sensitivity". Recall is a measure of thoroughness that determines how many positive tuples were identified. In fraud detection, recall is crucial because it indicates the proportion of actual fraud cases that are correctly identified. Recall is expressed as follows in Eq. (3):

$$\text{Recall (R)} = \frac{TP}{TP + FN} \ (0 < R < 1) \tag{3}$$

- **F1 Score:** F1-score is the weighted average of precision and recall. It provides a balance between precision and recall. F1-Score is especially useful when there is a significant class imbalance, as it considers both false positives and false negatives. It can be measured using Eq. (4):

$$\text{F1} - \text{Score} = \frac{2 * \ Precision \ * \ Recall}{Precision + Recall} \tag{4}$$

In this study, recall is given more emphasis as it shows how sensitive the classifier is to detecting fraudulent transactions. Additionally, the area under the receiver operating characteristic curve (AUROC) measures the model's accuracy in distinguishing between the two classes of data observations. AUROC scores for credit card records range from 0.50 to 1.00, where a score of 0.50 represents a random classifier and a score of 1.00 indicates a perfect classifier. AUROC scores above 0.80 are generally considered indicative of a competent classifier. This means that the classifier has a good ability to distinguish between fraudulent and non-fraudulent transactions, demonstrating a high level of predictive accuracy.

In addition, we examine the misclassification rate of fraudulent classes by FNR and FPR. As a key performance indicator, the misclassification rate indicates the rate of incorrect predictions without distinguishing between positive and negative outcomes. Accurately predicting fraudulent transactions is critical for detecting credit card fraud. Therefore, the best model with the lowest FNR

Table 2
Sampling of credit card dataset.

Method	Type	Legitimate Instances	Fraudulent Instances	New Total Instances
Random Under-sampling	Under-sampling	394	394	788
SMOTE	Over-sampling	227,451	227,451	454,902
ADASYN	Over-sampling	227,451	227,373	454,824
SMOTE-Tomek	Hybrid-sampling	226,808	226,808	453,616
SMOTE-ENN	Hybrid-sampling	227,451	218,649	446,100

(number of fraudulent transactions considered legitimate transactions) and highest recall is selected for this study. Furthermore, receiver operating characteristics (ROC) representing the value of TPR as a function of FPR for different cut-off points are analyzed. A point on the ROC curve indicates sensitivity/specificity pairs corresponding to decision thresholds.

5.3. Comparative analysis and discussion

In the present study, different ML algorithms are initially applied to the original (highly imbalanced) credit card dataset. Then we consider random under-sampling, SMOTE, ADASYN, SMOTE-Tomek, and SMOTE-ENN, respectively, as under-sampling, over-sampling, and hybrid sampling techniques for the credit card dataset. Five sampling techniques listed in Table 2 are applied, and as a consequence, five separate credit card training datasets of different sizes are produced.

A comparative analysis of eleven ML models with and without the sampling technique is presented in Table 3. Our study has revealed that RF excels in precision, achieving the highest value of 0.9868 and simultaneously minimizing the FPR to 0.0000 on the original dataset. Additionally, RF achieves the highest F1-score of 0.8677 when utilizing the SMOTE-Tomek hybrid sampled dataset. Furthermore, MLP exhibits superior discrimination capabilities, with the highest AUROC value of 0.9881 when utilizing the SMOTE oversampled dataset. This emphasizes MLP's ability to accurately discriminate between legitimate and fraudulent transactions. For the under-sampled dataset, XGBoost stands out with the highest recall of 0.9388, indicating its exceptional ability to capture a significant proportion of actual fraud cases while keeping the FNR at a low value of 0.0612. This underscores the effectiveness of XGBoost in reducing the risk of missing fraudulent transactions in scenarios with imbalanced classes. In this study, recall is prioritized as a crucial metric, emphasizing the sensitivity of classifiers in detecting fraudulent transactions. Our findings highlight the exceptional performance of XGBoost, securing the highest recall values and minimizing the FNR, as detailed in Table 3.

In our continuous pursuit of optimizing model performance, the ensemble-based soft-voting approach is applied to the same datasets using three distinct voting classifier (VC) combinations, incorporating the top-performing classifiers, namely XGBoost, RF, MLP, and KNN. Table 4 presents a comparative analysis of the soft-voting ensemble learning approach for both original imbalanced and sampled datasets. In the soft voting strategy, the voting classifier (VC1), a combination of RF, XGBoost, and MLP, achieves the highest AUROC of 0.9936 for the ADASYN oversampled dataset. VC2 (RF, XGBoost, KNN) provides the highest precision and F1-score of 0.9870 and 0.8764 respectively, and also coupled with FPR of 0.0000 for the original dataset. VC3 (XGBoost, MLP, and KNN) excels in recall with a value of 0.9694 and maintains the lowest FNR of 0.0306 for the under-sampled dataset. The results of soft voting classifiers (refer to Table 4) yield substantial improvements, surpassing the individual classifiers (refer to Table 3) in terms of all the performance matrices. The performance analysis bar charts of different classifiers on both the imbalanced and sampled datasets are shown in Fig. 6, while the ROC curve graphs are depicted in Fig. 7.

To demonstrate the significance of our work, the proposed soft-voting approach is compared with previous studies considering the same dataset (see Table 5). It is observed that our proposed model outperforms existing methods in terms of precision, F1 score, AUROC, and recall. These metrics are of paramount importance in fraud detection, ensuring an effective ability to identify and prevent fraudulent activities. It is worth noting that our current study does not outperform other models in terms of accuracy, despite its remarkable performance. The accuracy metric, when dealing with a highly imbalanced dataset such as the credit card dataset, can be misleading. Therefore, it is essential to consider a broader range of performance metrics, including FNR, FPR, precision, F1 score, recall, and AUROC.

5.4. Statistical test

In contemporary computer science practice, researchers must determine whether or not the obtained improvements are statistically significant, since the experimental outcomes are usually insufficient to state that one algorithm performs better when compared to other competitors. In this paper, the Friedman test [83] is used to determine the statistical significance of the proposed soft voting approach (VC1, VC2, and VC3) over other individual ML algorithms (i.e., XGBoost, because it outperforms other ML algorithms in terms of recall and FNR). The evaluation metrics considered for each classifier include accuracy, precision, F1-score, recall, AUROC, FNR, and FPR. The Friedman test results on various datasets employing different sampling techniques are reported in Table 6. The original dataset yields a significant Chi-squared statistic of 24.0000 with a corresponding P-value of 0.0005. The under-sampled dataset exhibits a Chi-squared statistic of 20.0357 and a p-value of 0.0027, indicating a statistically significant difference. The over-sampled datasets, utilizing SMOTE and ADASYN techniques, both display a Chi-squared statistic of 22.7143 with P-values of 0.0009. Similarly, the hybrid sampled datasets (SMOTE-Tomek and SMOTE-ENN) show a Chi-squared statistics of 22.1786 and 24.0000, respectively, with corresponding P-values of 0.0011 and 0.0005.

Table 3
Comparative analysis of ML classifiers.

Classifier	Datasets	Sampling Techniques Used?	Sampling Techniques	Accuracy	Precision	F1-score	Detection Rate		Misclassification Rate	
							Recall	AUROC	FNR	FPR
LR	Original	No	No	0.9991	0.8636	0.6951	0.5816	0.9747	0.4184	0.0002
	Under-sampled	Yes	Random-Sampling	0.9502	0.0308	0.0596	0.9184	0.9712	0.0816	0.0498
	Over-sampled	Yes	SMOTE	0.9826	0.0814	0.1491	0.8878	0.9770	0.1122	0.0173
			ADASYN	0.9799	0.0718	0.1330	0.8980	0.9299	0.1020	0.0200
	Hybrid-sampled	Yes	SMOTE-Tomek	0.9835	0.0857	0.1563	0.8878	0.9071	0.1122	0.0163
			SMOTE-ENN	0.9675	0.0452	0.0860	0.8878	0.9620	0.1122	0.0323
RF	Original	No	No	**0.9996**	**0.9868**	0.8621	0.7653	0.9525	0.2347	**0.0000**
	Under-sampled	Yes	Random-Sampling	0.9759	0.0620	0.1161	0.9184	0.9795	0.0816	0.0240
	Over-sampled	Yes	SMOTE	0.9995	0.8723	0.8542	0.8367	0.9650	0.1633	0.0002
			ADASYN	0.9995	0.8646	0.8557	0.8469	0.9700	0.1531	0.0002
	Hybrid-sampled	Yes	SMOTE-Tomek	**0.9996**	0.9011	**0.8677**	0.8367	0.9791	0.1633	0.0002
			SMOTE-ENN	0.9995	0.8542	0.8454	0.8367	0.9857	0.1633	0.0002
XGBoost	Original	No	No	0.9996	0.9620	0.8588	0.7755	0.9811	0.2245	0.0001
	Under-sampled	Yes	Random-Sampling	0.9610	0.0399	0.0766	**0.9388**	0.9825	**0.0612**	0.0389
	Over-sampled	Yes	SMOTE	0.9994	0.8000	0.8276	0.8571	0.9860	0.1429	0.0004
			ADASYN	0.9994	0.8137	0.8300	0.8469	0.9827	0.1531	0.0003
	Hybrid-sampled	Yes	SMOTE-Tomek	0.9994	0.8283	0.8325	0.8367	0.9835	0.1633	0.0003
			SMOTE-ENN	0.9993	0.7830	0.8137	0.8469	0.9847	0.1531	0.0004
SVM	Original	No	No	0.9993	0.9683	0.7578	0.6224	0.5000	0.3776	0.0000
	Under-sampled	Yes	Random-Sampling	0.4841	0.0019	0.0039	0.5816	0.5494	0.4184	0.5161
	Over-sampled	Yes	SMOTE	0.4854	0.0013	0.0027	0.3980	0.4723	0.6020	0.5144
			ADASYN	0.4622	0.0015	0.0029	0.4592	0.4701	0.5408	0.5378
	Hybrid-sampled	Yes	SMOTE-Tomek	0.4851	0.0013	0.0027	0.3980	0.4723	0.6020	0.5147
		Yes	SMOTE-ENN	0.4634	0.0014	0.0028	0.4388	0.5279	0.5612	0.5366
AdaBoost	Original	No	No	0.9993	0.8554	0.7845	0.7245	0.9779	0.2755	0.0002
	Under-sampled	Yes	Random-Sampling	0.9451	0.0283	0.0550	0.9286	0.9798	0.0714	0.0549
	Over-sampled	Yes	SMOTE	0.9866	0.1074	0.1926	0.9286	0.9644	0.0714	0.0133
			ADASYN	0.9987	0.5950	0.6575	0.7347	0.9611	0.2653	0.0009
	Hybrid-sampled	Yes	SMOTE-Tomek	0.9871	0.1104	0.1972	0.9184	0.9822	0.0816	0.0127
			SMOTE-ENN	0.9874	0.1135	0.2022	0.9286	0.9827	0.0714	0.0125
SGD	Original	No	No	0.9989	0.8333	0.5921	0.4592	0.5000	0.5408	0.0002
	Under-sampled	Yes	Random-Sampling	0.9816	0.0021	0.0038	0.0204	0.5000	0.9796	0.0168
	Over-sampled	Yes	SMOTE	0.9845	0.0811	0.1469	0.7755	0.8500	0.2245	0.0151
			ADASYN	0.9982	0.1250	0.0189	0.0102	0.5100	0.9898	0.0001
	Hybrid-sampled	Yes	SMOTE-Tomek	0.9950	0.1655	0.2447	0.4694	0.9200	0.5306	0.0041
			SMOTE-ENN	0.0026	0.0017	0.0034	1.0000	0.8750	0.0000	0.9991
MLP	Original	No	No	0.9995	0.8864	0.8387	0.7959	0.9694	0.2041	0.0002
	Under-sampled	Yes	Random-Sampling	0.9443	0.0089	0.0173	0.2857	0.6835	0.7143	0.0546
	Over-sampled	Yes	SMOTE	0.9852	0.0984	0.1779	0.9286	**0.9881**	0.0714	0.0147
			ADASYN	0.9935	0.1955	0.3204	0.8878	0.9747	0.1122	0.0063
	Hybrid-sampled	Yes	SMOTE-Tomek	0.9952	0.2507	0.3910	0.8878	0.9825	0.1122	0.0046
			SMOTE-ENN	0.9890	0.1255	0.2206	0.9082	0.9855	0.0918	0.0109
DT	Original	No	No	0.9992	0.7477	0.7805	0.8163	0.9079	0.1837	0.0005
	Under-sampled	Yes	Random-Sampling	0.8905	0.0142	0.0281	0.9184	0.9044	0.0816	0.1095
	Over-sampled	Yes	SMOTE	0.9979	0.4405	0.5564	0.7551	0.8167	0.2449	0.0017
			ADASYN	0.9978	0.4167	0.5396	0.7653	0.8417	0.2347	0.0018
	Hybrid-sampled	Yes	SMOTE-Tomek	0.9977	0.4134	0.5343	0.7551	0.8766	0.2449	0.0018
			SMOTE-ENN	0.9977	0.4130	0.5390	0.7755	0.8868	0.2245	0.0019
GNB	Original	No	No	0.9778	0.0604	0.1124	0.8163	0.9671	0.1837	0.0219
	Under-sampled	Yes	Random-Sampling	0.9840	0.0744	0.1350	0.7245	0.9675	0.2755	0.0155
	Over-sampled	Yes	SMOTE	0.9922	0.1429	0.2375	0.7041	0.9199	0.2959	0.0073
			ADASYN	0.9918	0.1409	0.2365	0.7347	0.9110	0.2653	0.0077
	Hybrid-sampled	Yes	SMOTE-Tomek	0.9923	0.1437	0.2388	0.7041	0.9200	0.2959	0.0072
			SMOTE-ENN	0.9924	0.1459	0.2417	0.7041	0.9600	0.2959	0.0071
GB	Original	No	No	0.9989	0.7375	0.6629	0.6020	0.7855	0.3980	0.0004
	Under-sampled	Yes	Random-Sampling	0.9616	0.0404	0.0775	0.9387	0.9808	0.0611	0.0384

(*continued on next page*)

Table 3 (*continued*)

Classifier	Datasets	Sampling Techniques Used?	Sampling Techniques	Accuracy	Precision	F1-score	Detection Rate		Misclassification Rate	
							Recall	AUROC	FNR	FPR
	Over-sampled	Yes	SMOTE	0.9941	0.2131	0.3444	0.8980	0.9744	0.1020	0.0057
			ADASYN	0.9938	0.2051	0.3340	0.8980	0.9847	0.1020	0.0060
	Hybrid-sampled	Yes	SMOTE-Tomek	0.9939	0.2052	0.3333	0.8878	0.9646	0.1122	0.0059
			SMOTE-ENN	0.9941	0.2112	0.3412	0.8878	0.9863	0.1122	0.0057
KNN	Original	No	No	0.9995	0.9383	0.8492	0.7755	0.9336	0.2245	0.0001
	Under-sampled	Yes	Random-Sampling	0.6723	0.0032	0.0063	0.6020	0.6656	0.3980	0.3276
	Over-sampled	Yes	SMOTE	0.9495	0.0190	0.0368	0.5612	0.7681	0.4388	0.0498
			ADASYN	0.9470	0.0178	0.0345	0.5510	0.7625	0.4490	0.0523
	Hybrid-sampled	Yes	SMOTE-Tomek	0.9496	0.0191	0.0369	0.5612	0.7687	0.4388	0.0498
			SMOTE-ENN	0.9288	0.0140	0.0273	0.5816	0.7546	0.4184	0.0706

This finding indicates that there is a statistically significant difference in performance across at least one performance metric among the classifiers. The Friedman p-value, which is below the conventional significance threshold of 0.05, provides evidence to reject the null hypothesis (H_0), suggesting that the performance of the proposed soft voting method is statistically significant compared with other individual algorithms. In summary, the Friedman Test results reveal statistically significant differences between XGBoost and VC (VC1, VC2, and VC3) for the five sampled datasets. These findings provide valuable insights into the performances of XGBoost and VC across diverse sampling strategies. Additionally, the significance of these results emphasizes the importance of considering different sampling methods in the evaluation and deployment of machine-learning models for imbalanced classification tasks.

6. Conclusion

Undoubtedly, credit card fraud is a significant concern for all financial firms. The European credit card holder dataset is an excellent example of a skewed dataset because the majority of entries represent valid transactions and only a small portion mark fraudulent actions. Therefore, it is not accurate to evaluate the performance of classifiers based only on accuracy. Our findings highlight the effectiveness of precision, recall, F1-score, AUROC, and FNR as critical evaluation metrics. We apply eleven machine learning algorithms and three soft-voting classifiers on both the original imbalanced and sampled datasets, as outlined in our methodology. Therefore, by prioritizing precision, recall, F1-score, AUROC while minimizing the FNR, the proposed ensemble-based soft-voting method outperforms individual ML classifiers for each evaluation metric. Finally, a statistical test is performed to confirm the significance of outcomes. These results emphasize the pivotal role of model selection and ensemble methods in enhancing fraud detection in real-world scenarios. The approach we develop enables authorities to get notified of credit card fraud and take the appropriate steps to investigate the transaction, and classify it as either fraudulent or legitimate. The implications of this study will contribute significantly to advancing security measures and risk mitigation in the financial industry.

Limitations and future work

Despite the outstanding outcomes of our proposed methodology, certain drawbacks should be addressed in the future. This study is limited to the European credit card dataset (2013), but it could be more comprehensive if it includes recent credit card datasets such as the European dataset (2022), UCI dataset, and IEEE-CIS dataset. In further research, it is suggested that hyperparameters be tuned using metaheuristic techniques in combination with a deep neural network model to identify fraudulent transactions. Furthermore, it would be pertinent to compare the trade-off to other methods, particularly the computational complexity of the models. The findings of this study should be further validated in real-world financial settings to ensure practical effectiveness. Additionally, researchers could explore post-hoc tests to identify which specific sampling methods differ from each other.

Data availability

https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud, last accessed on December 24, 2023.

CRediT authorship contribution statement

Mimusa Azim Mim: Conceptualization, Data curation, Formal analysis, Methodology, Visualization, Writing – original draft. **Nazia Majadi:** Conceptualization, Data curation, Investigation, Methodology, Supervision, Writing – review & editing. **Peal Mazumder:** Data curation, Formal analysis, Methodology, Visualization.

Table 4
Comparative analysis of soft-voting classifiers.

Strategy	Voting Classifier (VC)	Dataset	Sampling Techniques	Accuracy	Precision	F1-score	Detection Rate		Misclassification Rate	
							Recall	AUROC	FNR	FPR
Soft-Voting	VC1 (RF, XGBoost, MLP)	Original	No	0.9996	0.9747	0.8701	0.7857	0.9807	0.2143	0.0000
		Under-sampled	Random-Sampling	0.9902	0.1395	0.2418	0.9082	0.9794	0.0918	0.0097
		Over-sampled	SMOTE	0.9993	0.7727	0.8173	0.8673	0.9893	0.1327	0.0004
			ADASYN	0.9994	0.8000	0.8276	0.8571	**0.9936**	0.1429	0.0004
		Hybrid-sampled	SMOTE-Tomek	0.9994	0.8058	0.8259	0.8469	0.9848	0.1531	0.0004
			SMOTE-ENN	0.9993	0.7679	0.8190	0.8776	0.9862	0.1224	0.0005
	VC2 (RF, XGBoost, KNN)	Original	No	**0.9996**	**0.9870**	**0.8764**	0.7959	0.9817	0.2041	**0.0000**
		Under-sampled	Random-Sampling	0.9696	0.0507	0.0962	0.9388	0.9763	0.0612	0.0303
		Over-sampled	SMOTE	0.9996	0.9121	0.8783	0.8469	0.9782	0.1531	0.0001
			ADASYN	0.9996	0.9121	0.8783	0.8469	0.9724	0.1531	0.0001
		Hybrid-sampled	SMOTE-Tomek	0.9996	0.9022	0.8737	0.8469	0.9706	0.1531	0.0002
			SMOTE-ENN	0.9995	0.8485	0.8528	0.8571	0.9770	0.1429	0.0003
	VC3 (XGBoost, MLP, KNN)	Original	No	0.9996	0.9625	0.8652	0.7857	0.9820	0.2143	0.0001
		Under-sampled	Random-Sampling	0.6261	0.0044	0.0088	**0.9694**	0.9661	**0.0306**	0.3745
		Over-sampled	SMOTE	0.9988	0.6115	0.7173	0.8673	0.9755	0.1327	0.0009
			ADASYN	0.9994	0.7850	0.8195	0.8571	0.9743	0.1429	0.0004
		Hybrid-Sampled	SMOTE-Tomek	0.9890	0.1213	0.2128	0.8673	0.9738	0.1327	0.0108
			SMOTE-ENN	0.9990	0.6667	0.7500	0.8571	0.9797	0.1429	0.0007

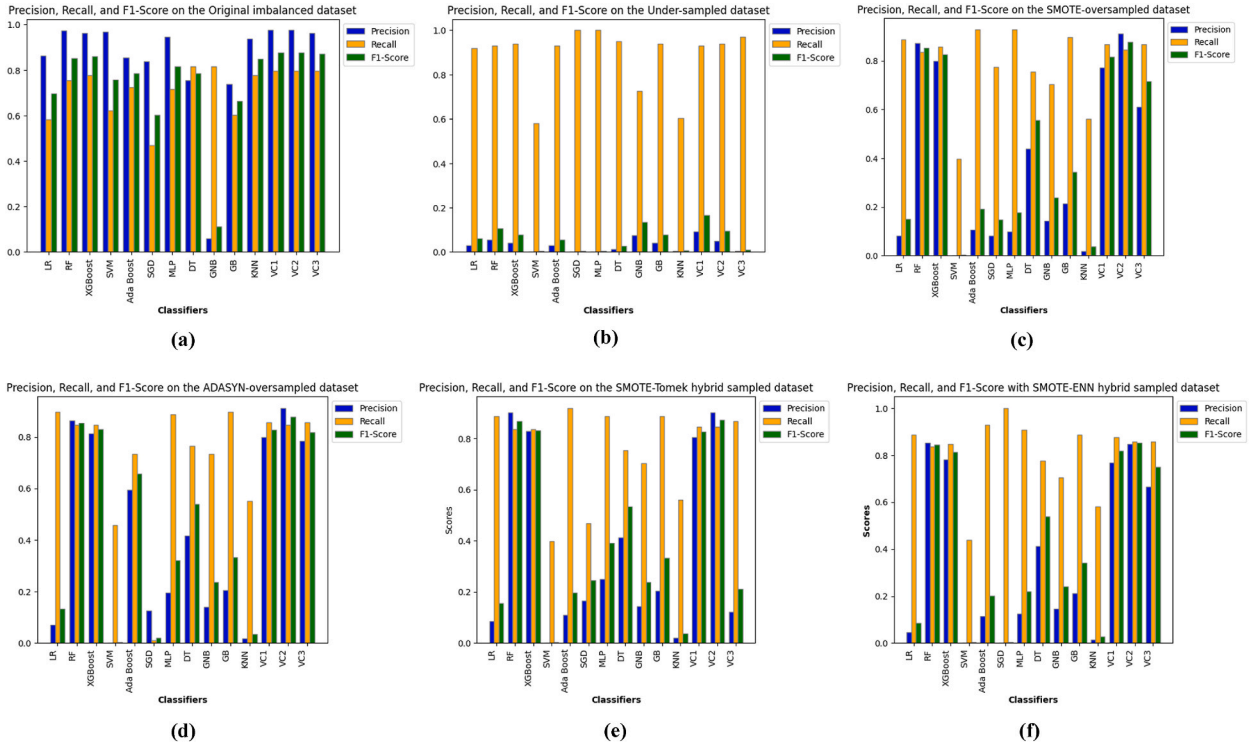
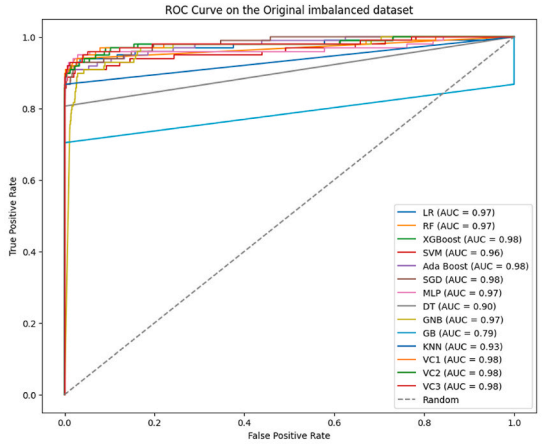
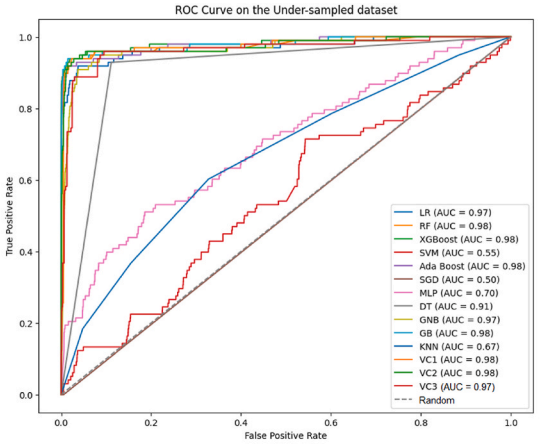


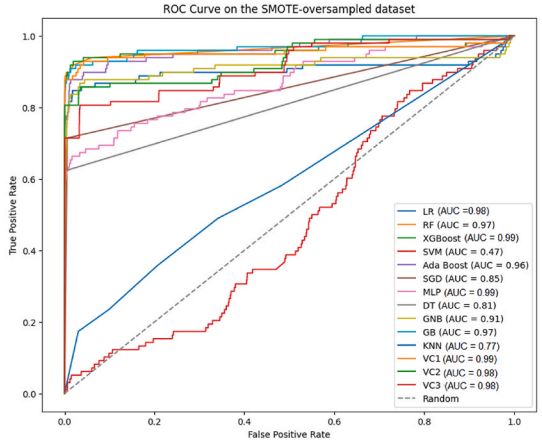
Fig. 6. Performance analysis bar chart of eleven ML and three soft-voting classifiers on the (a) Original imbalanced dataset, (b) Under-sampled dataset, (c) SMOTE-oversampled dataset, (d) ADASYN-oversampled dataset, (e) SMOTE-Tomek hybrid sampled dataset, (f) SMOTE-ENN hybrid sampled dataset.



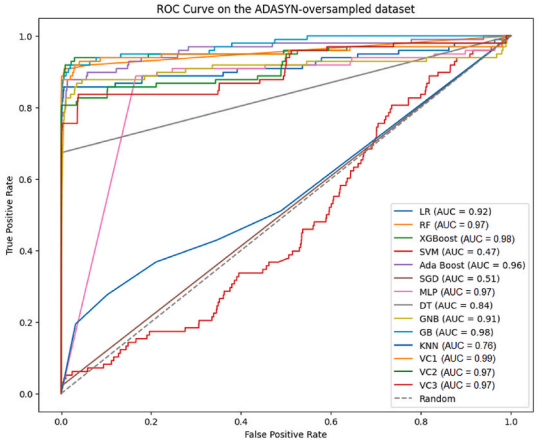
(a)



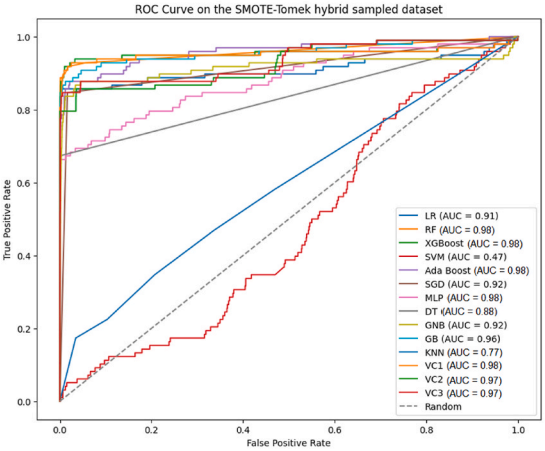
(b)



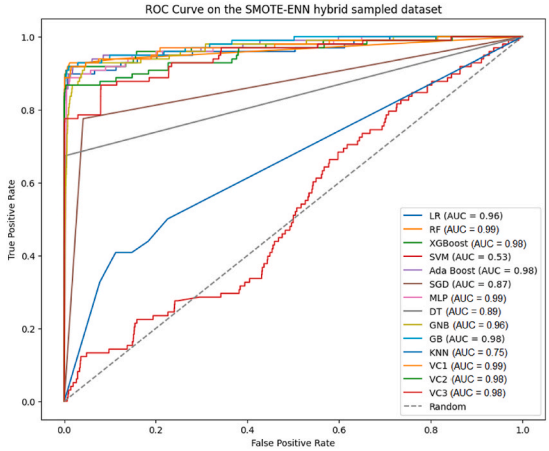
(c)



(d)



(e)



(f)

Fig. 7. ROC curve for comparing the performance of ML and soft-voting classifiers on the (a) Original imbalanced dataset, (b) Under-sampled dataset, (c) SMOTE-oversampled dataset, (d) ADASYN-oversampled dataset, (e) SMOTE-Tomek hybrid sampled dataset, (f) SMOTE-ENN hybrid sampled dataset.

Table 5
A comparative analysis of our proposed approach with the previous methods.

Paper	Year	Model	Accuracy	Precision	F1-score	Detection Rate		Misclassification Rate	
						Recall	AUROC	FNR	FPR
Kumar et al. [32]	2019	RF	0.9000	-	-	-	-	-	-
Vermedja et al. [35]	2019	RF + SMOTE	**0.9996**	0.9638	–	0.8163		-	-
Puh and Brkić [36]	2019	LR + SMOTE+ StaticLearning	-	–	–	–	0.9114	-	-
John and Naaz [37]	2019	LOF + iForest	0.9700	-	-	-	-	-	-
Khatri et al. [41]	2020	KNN	–	0.9111	–	0.8119	–	–	–
Taha and Malebary [42]	2020	LightGMB + Hyper-parameter	0.9840	0.9734	0.5695	0.4059	0.9094	–	–
Vengatesan et al. [43]	2020	KNN	-	0.9500	0.8200	0.7200	–	–	–
Hema [44]	2021	RF	0.9995	0.9195	0.8510	0.7920	0.8900	–	–
Asha and KR [45]	2021	ANN	0.9992	0.8115	–	0.7619	–	–	–
Alfiz and Fati [53]	2022	AllKNN-CatBoost	**0.9996**	0.8028	0.8740	0.9591	0.9794	-	-
Kafhali and Tayebi [47]	2022	DE + XGB + SMOTE + ENN	0.9994	0.8068	0.8327	0.8602	0.9921	-	-
Present Study	**2023**	**Ensemble-based Soft-voting**	**0.9996**	**0.9870**	**0.8764**	**0.9694**	**0.9936**	**0.0306**	**0.0000**

Table 6
Friedman test results between XGBoost and VC for the five sampled dataset.

	XGBoost - VC					
	Original Dataset	Under-sampled Dataset	Over-sampled Dataset (SMOTE)	Over-sampled Dataset (ADASYN)	Hybrid-sampled Dataset (SMOTE-Tomek)	Hybrid-sampled Dataset (SMOTE-ENN)
Chi-squared Statistic	24.0000	20.0357	22.7143	22.7143	22.1786	24.0000
P-Value	0.0005	0.0027	0.0009	0.0009	0.0011	0.0005

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the ICT Ministry of Bangladesh (The Tracking Number is 24FS16588).

References

[1] Shift Credit Card Processing, [Online] Available: https://shiftprocessing.com/credit-card/, Last accessed on 24 December 2023.
[2] Kinsta Ecommerce Statistics, [Online] Available: https://kinsta.com/blog/ecommerce-statistics/, Last accessed on 21 December 2023.
[3] Unisys Security Index, [Online] Available: https://www.app5.unisys.com/library/cmsmail/USI/Unisys%20Security%20Index_Global.pdf, Last accessed on 27 December 2023.
[4] Nilson Report, [Online] Available: https://nilsonreport.com/newsletters/1209/, Last accessed on 30 January 2024.
[5] S.K. Jagatheesaperumal, M. Rahouti, K. Ahmad, A. Al-Fuqaha, M. Guizani, The Duo of artificial intelligence and big data for industry 4.0: applications, techniques, challenges, and future research directions, IEEE Internet Things J. (2021), https://doi.org/10.1109/JIOT.2021.3139827.
[6] N. Japkowicz, S. Stephen, The class imbalance problem: a systematic study, Intell. Data Anal. 6 (5) (2002) 429–449, https://doi.org/10.3233/ida-2002-6504.
[7] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.
[8] V. Ganganwar, An overview of classification algorithms for imbalanced datasets, International Journal of Emerging Technology and Advanced Engineering 2 (4) (2012) 42–47.
[9] M. Tayebi, S. El Kafhali, Performance analysis of metaheuristics based hyperparameters optimization for fraud transactions detection, Evolutionary Intelligence (2022) 1–19.
[10] P. Caroline Cynthia, S. Thomas George, An outlier detection approach on credit card fraud detection using machine learning: a comparative analysis on supervised and unsupervised learning, in: Intelligence in Big Data Technologies—Beyond the Hype: Proceedings of ICBDCC 2019, Springer Singapore, 2021, pp. 125–135, https://doi.org/10.1007/978-981-15-5285-4_12.
[11] R. More, C. Awati, S. Shirgave, R. Deshmukh, S. Patil, Credit card fraud detection using supervised learning approach, International journal of scientific & technology research 9 (10) (2021) 216–219.
[12] E. Esenogho, I.D. Mienye, T.G. Swart, K. Aruleba, G. Obaido, A neural network ensemble with feature engineering for Improved Credit Card Fraud Detection, IEEE Access: Practical Innovations, Open Solutions 10 (2022) 16400–16407, https://doi.org/10.1109/ACCESS.2022.3148298.
[13] T. Razooqi, P. Khurana, K. Raahemifar, A. Abhari, Credit card fraud detection using fuzzy logic and neural network, in: Proceedings of the 19th Communications & Networking Symposium, Academic Press, 2016, April, pp. 1–5.
[14] M. Tayebi, S. El Kafhali, Deep neural networks hyperparameter optimization using Particle Swarm optimization for detecting frauds transactions, in: Advances on Smart and Soft Computing, Springer, 2022, pp. 507–516, https://doi.org/10.1007/978-981-16-5559-3_42.
[15] A.S. Hussein, R.S. Khairy, S.M.M. Najeeb, Credit card fraud detection using fuzzy rough nearest neighbor and sequential minimal optimization with logistic regression, International Journal of Interactive Mobile Technologies 15 (5) (2021) 24–42, https://doi.org/10.3991/ijim.v15i05.17173.

[16] M. Tayebi, S. El Kafhali, Hyperparameter optimization using genetic algorithmsto detect frauds transactions, in: The International Conference on Artificial Intelligence and Computer Vision, Springer, 2021,June, pp. 288–297, https://doi.org/10.1007/978-3-030-76346-6_27.

[17] A. Gupta, M.C. Lohani, M. Manchanda, Financial fraud detection using naive bayes algorithm in highly imbalance data set, J. Discrete Math. Sci. Cryptogr. 24 (5) (2021) 1559–1572, https://doi.org/10.1080/09720529.2021.1969733. Bahnsen, A. C., Villegas, S., Aouada, D., & Ottersten, B. (2019). Fraud detection by stacking cost-sensitive decision trees. In Data Science for Cyber-Security (pp. 251–266). Academic Press.

[18] S. Ghosh, D.L. Reilly, Credit card fraud detection with a neural-network, in: System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on, IEEE, 1994, January, pp. 621–630, https://doi.org/10.1109/hicss.1994.323314, vol. 3.

[19] A.L. Prodromidis, S. Stolfo, Agent-based Distributed Learning Applied to Fraud Detection, 1999, https://doi.org/10.7916/D86Q28GG.

[20] S.J. Stolfo, W. Fan, W. Lee, A. Prodromidis, P.K. Chan, Cost-based modelling for fraud and intrusion detection: results from the JAM project, in: Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00, IEEE, 2000, January, pp. 130–144, vol. 2.

[21] F. Carcillo, Y.A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, G. Bontempi, Combining unsupervised and supervised learning in credit card fraud detection, Inf. Sci. 557 (2021) 317–331, https://doi.org/10.1016/j.ins.2019.05.042.

[22] A.K. Rai, R.K. Dwivedi, Fraud detection in credit card data using unsupervised machine learning based scheme, in: 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020, July, pp. 421–426, https://doi.org/10.1109/ICESC48915.2020.9155615. IEEE.

[23] D. Olszewski, Fraud detection using self-organizing map visualizing the user profiles, Knowl. Base Syst. 70 (2014) 324–334, https://doi.org/10.1016/j.knosys.2014.07.008.

[24] W.F. Yu, N. Wang, Research on credit card fraud detection model based on distance sum, in: 2009 International Joint Conference on Artificial Intelligence, IEEE, 2009, April, pp. 353–356, https://doi.org/10.1109/jcai.2009.146.

[25] M.H. Özçelik, E. Duman, M. Işik, T. Çevik, Improving a credit card fraud detection system using genetic algorithm, in: 2010 International Conference on Networking and Information Technology, IEEE, 2010, June, pp. 436–440, https://doi.org/10.1109/icnit.2010.5508478.

[26] N. Soltani, M.K. Akbari, M.S. Javan, A new user-based model for credit card fraud detection based on artificial immune system, in: The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012), IEEE, 2012, May, pp. 29–33, https://doi.org/10.1109/AISP.2012.6313712.

[27] M. Zareapoor, K.R. Seeja, M.A. Alam, Analysis on credit card fraud detection techniques: based on certain design criteria, International journal of computer applications 52 (3) (2012), https://doi.org/10.5120/8184-1538.

[28] S. Vats, S.K. Dubey, N.K. Pandey, Genetic algorithms for credit card fraud detection, in: International Conference on Education and Educational Technologies, 2013, July, https://doi.org/10.47893/ijcns.2013.1062.

[29] R.D. Patel, D.K. Singh, Credit card fraud detection & prevention of fraud using genetic algorithm, Int. J. Soft Comput. Eng. 2 (6) (2013) 292–294, https://doi.org/10.17577/ijertv9is070649.

[30] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, C. Jiang, Random forest for credit card fraud detection, in: 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), IEEE, 2018, March, pp. 1–6, https://doi.org/10.1109/icnsc.2018.8361343.

[31] M.S. Kumar, V. Soundarya, S. Kavitha, E.S. Keerthika, E. Aswini, Credit card fraud detection using random forest algorithm, in: 2019 3rd International Conference on Computing and Communications Technologies (ICCCT), IEEE, 2019, February, pp. 149–153, https://doi.org/10.1109/ICCCT2.2019.8824930.

[32] R. Jain, B. Gour, S. Dubey, A hybrid approach for credit card fraud detection using rough set and decision tree technique, Int. J. Comput. Appl. 139 (10) (2016) 1–6, https://doi.org/10.5120/ijca2016909325.

[33] S. Carta, G. Fenu, D.R. Recupero, R. Saia, Fraud detection for E-commerce transactions by employing a prudential Multiple Consensus model, J. Inf. Secur. Appl. 46 (2019) 13–22, https://doi.org/10.1016/j.jisa.2019.02.007.

[34] D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic, A. Anderla, Credit card fraud detection-machine learning methods, in: 2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH), IEEE, 2019, March, pp. 1–5, https://doi.org/10.1109/INFOTEH.2019.8717766.

[35] M. Puh, L. Brkić, Detecting credit card fraud using selected machine learning algorithms, in: 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), IEEE, 2019, May, pp. 1250–1255, https://doi.org/10.23919/MIPRO.2019.8757212.

[36] H. John, S. Naaz, Credit card fraud detection using local outlier factor and isolation forest, Int. J. Comput. Sci. Eng. 7 (4) (2019) 1060–1064, https://doi.org/10.26438/ijcse/v7i4.10601064.

[37] H. Najadat, O. Altiti, A.A. Aqouleh, M. Younes, Credit card fraud detection based on machine and deep learning, in: 2020 11th International Conference on Information and Communication Systems (ICICS), IEEE, 2020, April, pp. 204–208, https://doi.org/10.1109/access.2022.3166891.

[38] V. Van Vlasselaer, C. Bravo, O. Caelen, T. Eliassi-Rad, L. Akoglu, M. Snoeck, B. Baesens, APATE: a novel approach for automated credit card transaction fraud detection using network-based extensions, Decis. Support Syst. 75 (2015) 38–48, https://doi.org/10.1016/j.dss.2015.04.013.

[39] K.S. Varun Kumar, V.G. Vijaya Kumar, A. Vijay Shankar, K. Pratibha, Credit card fraud detection using machine learning algorithms, Int. J. Eng. Res. Technol. 9 (2020), https://doi.org/10.17577/ijertv9is070649.

[40] S. Khatri, A. Arora, A.P. Agrawal, Supervised machine learning algorithms for credit card fraud detection: a comparison, in: 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), IEEE, 2020, January, pp. 680–683, https://doi.org/10.1109/Confluence47617.2020.9057851.

[41] A.A. Taha, S.J. Malebary, An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine, IEEE Access 8 (2020) 25579–25587, https://doi.org/10.1109/ACCESS.2020.2971354.

[42] K. Vengatesan, A. Kumar, S. Yuvraj, V. Kumar, S. Sabnis, Credit card fraud detection using data analytic techniques, Adv. Math.: Scientific Journal 9 (3) (2020) 1185–1196, https://doi.org/10.37418/amsj.9.3.43.

[43] A. Hema, Machine learning methods for discovering credit card fraud, IRJCS:: International Research Journal of Computer Science VIII (2020) 1–6, https://doi.org/10.26562/irjcs.2021.v0801.001.

[44] R.B. Asha, S.K. Kr, Credit card fraud detection using artificial neural network, Global Transitions Proceedings 2 (1) (2021) 35–41, https://doi.org/10.1016/j.gltp.2021.01.006.

[45] M. Tayebi, S. El Kafhali, Credit card fraud detection based on Hyperparameters Optimization using the differential evolution, International Journal of Information Security and Privacy (IJISP) 16 (1) (2022) 1–21, https://doi.org/10.4018/IJISP.314156.

[46] S. El Kafhali, M. Tayebi, XGBoost based solutions for detecting fraudulent credit card transactions, in: 2022 International Conference on Advanced Creative Networks and Intelligent Systems (ICACNIS), IEEE, 2022, November, pp. 1–6, https://doi.org/10.1109/ICACNIS57039.2022.10054965.

[47] S. El Kafhali, M. Tayebi, Generative adversarial neural networks based oversampling technique for imbalanced credit card dataset, in: 2022 6th SLAAI International Conference on Artificial Intelligence (SLAAI-ICAI), IEEE, 2022, December, pp. 1–5, https://doi.org/10.1109/SLAAI-ICAI56923.2022.10002630.

[48] R. Panigrahi, S. Borah, A.K. Bhoi, M.F. Ijaz, M. Pramanik, Y. Kumar, A consolidated decision tree-based intrusion detection system for binary and multiclass imbalanced datasets, Mathematics 9 (7) (2021) 751, https://doi.org/10.3390/math9070751.

[49] C. Wang, D. Han, Credit card fraud forecasting model based on clustering analysis and integrated support vector machine, Cluster Comput. 22 (6) (2019) 13861–13866. https://link.springer.com/article/10.1007/s10586-018-2118-y.

[50] A. Bhanusri, K.R.S. Valli, P. Jyothi, G.V. Sai, R. Rohith, Credit card fraud detection using Machine learning algorithms, Journal of Research in Humanities and Social Science 8 (2) (2020) 4–11.

[51] V. Sellam, P. Tushar, G. Rohit, S. Sanyam, Credit card fraud detection using machine learning, Indian Journal of Computer Graphics and Multimedia 1 (1) (2021).

[52] N.S. Alfaiz, S.M. Fati, Enhanced credit card fraud detection model using machine learning, Electronics 11 (4) (2022) 662, https://doi.org/10.3390/electronics11040662.

[53] D.K. Padhi, N. Padhy, A.K. Bhoi, J. Shafi, M.F. Ijaz, A fusion framework for forecasting financial market direction using enhanced ensemble models and technical indicators, Mathematics 9 (21) (2021) 2646, https://doi.org/10.3390/math9212646.

[54] A.K. Nandi, K.K. Randhawa, H.S. Chua, M. Seera, C.P. Lim, Credit card fraud detection using a hierarchical behaviour-knowledge space model, PLoS One 17 (1) (2022) e0260579, https://doi.org/10.1371/journal.pone.0260579. PMID: 35051184.

[55] B. Jlifi, C. Sakrani, C. Duvallet, Towards a soft three-level voting model (Soft T-LVM) for fake news detection, J. Intell. Inf. Syst. (2022) 1–21.

[56] Data Analytics, [Online] Available: https://vitalflux.com/python-improve-model-performance-using-feature-scaling/, Last accessed on 30 January 2024.
[57] reason.town, [Online] Available: https://reason.town/why-normalize-data-machine-learning/, Last accessed on 27 December 2023.
[58] Z. Huo, B. Gu, H. Huang, Training neural networks using features replay, Adv. Neural Inf. Process. Syst. 31 (2018), https://doi.org/10.48550/arXiv.1807.04511.
[59] scikit learn, [Online] Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html, Last accessed on 24 December 2023.
[60] G. Vandewiele, I. Dehaene, G. Kovács, L. Sterckx, O. Janssens, F. Ongenae, T. Demeester, Overly optimistic prediction results on imbalanced data: a case study of flaws and benefits when applying over-sampling, Artif. Intell. Med. 111 (2021) 101987, https://doi.org/10.1016/j.artmed.2020.101987.
[61] A. Mahani, A.R.B. Ali, Classification problem in imbalanced datasets, Recent Trends in Computational Intelligence 1–23 (2019), https://doi.org/10.5772/intechopen.89603.
[62] K. Wang, J. Wan, G. Li, H. Sun, A hybrid algorithm-level ensemble model for imbalanced credit default prediction in the energy industry, Energies 15 (14) (2022) 5206, https://doi.org/10.3390/en15145206.
[63] T. Hasanin, T. Khoshgoftaar, The effects of random under sampling with simulated class imbalance for big data, in: 2018 IEEE International Conference on Information Reuse and Integration (IRI), IEEE, 2018, July, pp. 70–79, https://doi.org/10.1109/IRI.2018.00018.
[64] S. Warghade, S. Desai, V. Patil, Credit card fraud detection from imbalanced dataset using machine learning algorithm, Int. J. Comput. Trends Technol. 68 (3) (2020) 22–28, https://doi.org/10.14445/22312803/IJCTT-V68I3P105.
[65] R.F. De Morais, G.C. Vasconcelos, Boosting the performance of over-sampling algorithms through under-sampling the minority class, Neurocomputing 343 (2019) 3–18, https://doi.org/10.1016/j.neucom.2018.04.088.
[66] P.J. Huang, Classification of Imbalanced Data Using Synthetic Over-sampling Techniques, University of California, Los Angeles, 2015.
[67] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357. https://dl.acm.org/doi/10.5555/1622407.1622416.
[68] K. Savetratanakaree, K. Sookhanaphibarn, S. Intakosum, R. Thawonmas, Borderline over-sampling in feature space for learning algorithms in imbalanced data environments, IAENG Int. J. Comput. Sci. 43 (3) (2016).
[69] H. He, Y. Bai, E.A. Garcia, S. Li, ADASYN: adaptive synthetic sampling approach for imbalanced learning, in: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), IEEE, 2008, June, pp. 1322–1328, https://doi.org/10.1109/IJCNN.2008.4633969.
[70] D. Elreedy, A.F. Atiya, A novel distribution analysis for smote oversampling method in handling class imbalance, in: International Conference on Computational Science, Springer, Cham, 2019, June, pp. 236–248, https://doi.org/10.1007/978-3-030-22744-9_18.
[71] A. Gosain, S. Sardana, Handling class imbalance problem using oversampling techniques: a review, in: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2017, September, pp. 79–85, https://doi.org/10.1109/ICACCI.2017.8125820.
[72] A. Mahani, A.R.B. Ali, Classification problem in imbalanced datasets, Recent Trends in Computational Intelligence (2019) 1–23, https://doi.org/10.5772/intechopen.89603.
[73] Kaggle, [Online] Available: https://www.linkedin.com/pulse/what-imbalanced-dataset-its-impacts-machine-learning-models-cheruku/, Last accessed on 27 December 2023.
[74] Z. Xu, D. Shen, T. Nie, Y. Kou, A hybrid sampling algorithm combining M-SMOTE and ENN based on random forest for medical imbalanced data, J. Biomed. Inf. 107 (2020) 103465, https://doi.org/10.1016/j.jbi.2020.103465.
[75] EvolutionIQ, [Online] Available: https://evolutioniq.com/the-journey-begins/, Last accessed on 27 December 2023.
[76] T. Elhassan, M. Aljurf, Classification of imbalance data using tomek link (t-link) combined with random under-sampling (rus) as a data reduction method, Global J. Technol. Optim. S 1 (2016), https://doi.org/10.4172/2229-8711.S1111.
[77] M. Beckmann, N.F. Ebecken, B.S.P. de Lima, A KNN undersampling approach for data balancing, J. Intell. Learn Syst. Appl. 7 (4) (2015) 104, https://doi.org/10.4236/jilsa.2015.74010.
[78] N. Guo, K. Di, H. Liu, Y. Wang, J. Qiao, A metric-based meta-learning approach combined attention mechanism and ensemble learning for few-shot learning, Displays 70 (2021) 102065, https://doi.org/10.1016/j.displa.2021.102065.
[79] B.H. Mevik, V.H. Segtnan, T. Næs, Ensemble methods and partial least squares regression, J. Chemometr.: A Journal of the Chemometrics Society 18 (11) (2004) 498–507, https://doi.org/10.1002/cem.895.
[80] Credit Card Fraud Detection, [Online] Available: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud, Last accessed on 24 December 2023.
[81] A. Dal Pozzolo, O. Caelen, R.A. Johnson, G. Bontempi, Calibrating probability with undersampling for unbalanced classification, in: 2015 IEEE Symposium Series on Computational Intelligence, IEEE, 2015, December, https://doi.org/10.1109/ssci.201, pp. 159–166.
[82] J. Xu, G. Shan, A. Amei, J. Zhao, D. Young, S. Clark, A modified Friedman test for randomized complete block designs, Commun. Stat. Simulat. Comput. 46 (2) (2017) 1508–1519, https://doi.org/10.1080/03610918.2015.1006777.
[83] R. Eisinga, T. Heskes, B. Pelzer, M. Te Grotenhuis, Exact p-values for pairwise comparison of Friedman rank sums, with application to comparing classifiers, BMC Bioinf. 18 (1) (2017) 1–18.