Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 7194728, 16 pages
https://doi.org/10.1155/2021/7194728

Research Article

Solving Misclassification of the Credit Card Imbalance Problem Using Near Miss

Nhlakanipho Michael Mqadi,[1] Nalindren Naicker,[1] and Timothy Adeliyi[2]

[1]*ICT and Society Research Group, Information Systems, Durban University of Technology, Durban 4001, South Africa*
[2]*ICT and Society Research Group, Information Technology, Durban University of Technology, Durban 4001, South Africa*

Correspondence should be addressed to Nalindren Naicker; nalindrenn@dut.ac.za

Received 30 May 2021; Accepted 2 July 2021; Published 20 July 2021

Academic Editor: Jude Hemanth

In ordinary credit card datasets, there are far fewer fraudulent transactions than ordinary transactions. In dealing with the credit card imbalance problem, the ideal solution must have low bias and low variance. The paper aims to provide an in-depth experimental investigation of the effect of using a hybrid data-point approach to resolve the class misclassification problem in imbalanced credit card datasets. The goal of the research was to use a novel technique to manage unbalanced datasets to improve the effectiveness of machine learning algorithms in detecting fraud or anomalous patterns in huge volumes of financial transaction records where the class distribution was imbalanced. The paper proposed using random forest and a hybrid data-point approach combining feature selection with Near Miss-based undersampling technique. We assessed the proposed method on two imbalanced credit card datasets, namely, the European Credit Card dataset and the UCI Credit Card dataset. The experimental results were reported using performance matrices. We compared the classification results of logistic regression, support vector machine, decision tree, and random forest before and after using our approach. The findings showed that the proposed approach improved the predictive accuracy of the logistic regression, support vector machine, decision tree, and random forest algorithms in credit card datasets. Furthermore, we found that, out of the four algorithms, the random forest produced the best results.

1. Introduction

The South African Banking Risk Information Centre (SABRIC) presented its annual crime data for 2019, which showed that online banking fraud incidences climbed by 20% between 2018 and 2019 [1]. These statistics revealed that card fraud occurs in different forms, namely, "without the card," "when the card is lost," "when the card is stolen," and "when the card is not received." "Without the card fraud" is when the fraudulent transactions occur without the consent of the owner and while the physical card is in the owner's possession [1, 2]. "When the card is lost" fraud is defined as a fraud committed when the valid cardholder is not in possession of the card and transactions are made on the card. Furthermore, "when the card is stolen" fraud is when the fraud is committed by the person who is not the rightful owner of the card. "When the card is not received" fraud occurs when legitimately issued cards are intercepted before they reach their intended recipients [2]. The cards are subsequently used fraudulently by impostors who have intercepted them. Card fraud transactions stored by financial issuers are very small compared to legitimate transactions, which results in a high imbalance credit card dataset [3]. The situation in which the dominant classes have a significant advantage over the minority classes is referred to as imbalanced data. An imbalance credit card dataset refers to a class distribution in which the bulk of valid transactions recorded outnumber the minority fraudulent transactions [4]. The imbalance problems cause the machine learning classification solutions to be partial towards the majority class and produce a prediction with a high misclassification rate. Failure to deal with imbalanced data jeopardizes the machine learning system's integrity and prediction ability, which can have a significant cost impact

[5]. Learning algorithms operate on the assumption that the data is evenly distributed; hence, imbalanced data is acknowledged as part of the fundamental issues in the field of data analytics and data science [6].

The science of building and implementing algorithms that can learn patterns from previous events is known as machine learning [7]. The machine learning classifiers can be trained by continually feeding input data and assessing their performances. A machine learning classification solution employs sophisticated algorithms that loop over big datasets and evaluate data patterns [8]. In machine learning, the ideal solution must have low bias and should accurately model the true relationship of positive and negative classes. Machine learning classifiers tend to perform well with a specific dataset that has been manipulated to suit the classifier [9]. The use of one dataset tends to have a bias as the data could be manipulated to support the classification solution. This is commonly referred to as overfitting in machine learning. The ideal binary classification solutions should have low variability, by producing consistent predictions across different datasets [10]. The goal of this work was to conduct a thorough investigation of the impact of employing a hybrid data-point strategy to handle the misclassification problem in credit card datasets that were imbalanced. Oversampling, undersampling, and feature selection are examples of strategies for resampling data used in dealing with imbalanced classes at the data-point level [11]. The data-point level technique, according to Sotiris, Dimitris, and Panayiotis in [12], includes data interventions to lessen the impact of imbalanced datasets, and it is flexible enough to proceed with modern classifiers such as logistic regression, decision trees, and support vector machines.

We present in this paper a hybrid method that amalgamates the advantages of the random forest and a hybrid data-point technique to deal with the problem of imbalance learning in credit card fraud. Random forest used for prediction has the advantage of being able to manage datasets with several predictor variables. We further combined feature selection using correlation coefficients in order to make it easier for machine learning to classify with Near Miss-based undersampling technique. Using well-known performance metrics, the model outperformed other recognised models.

2. Related Works

Machine learning models work well when the dataset contains evenly distributed classes, known as a balanced dataset [13]. Pes in [14] looked at the efficacy of hybrid learning procedures that combine dimensionality reduction and ways for dealing with class imbalance. The research combines univariate and multivariate feature selection strategies with cost-sensitive classification and sampling-based class balance methods [14]. The dominance of blended learning strategies presented a dependable choice for recurrent random sampling, and investigations proved that hybrid learning strategies outperformed feature selection solely for unbalanced datasets [14]. Several studies analyzed and compared existing financial fraud detection algorithms in order to find the most effective strategy [9, 15, 16]. Using the confusion matrix, Zhou and Liu in [16] discovered that the random forest model outperformed logistic regression and decision tree based on accuracy, precision, and recall matrices. Albashrawi et al. in [9] found that the logistic regression model is the superior data mining tool for detecting financial fraud.

A paper by Minku et al. in [17] looked into the scenario of classes progressively appearing or disappearing. The Class-Based Ensemble for Class Evolution (CBCE) was suggested as a class-based ensemble technique. CBCE can quickly adjust to class evolution by keeping a base learner for each class and constantly updating the basic learners with new data. To solve the dynamic class imbalance problem induced by the steady growth of classes, the study developed a novel undersampling strategy for the base learners. The empirical investigations, according to Minku et al. in [17], revealed the efficiency of CBCE in various class evolution scenarios when compared to the existing class evolution adaptation method. CBCE responded well to all three scenarios of class evolution as compared to previous approaches (i.e., occurrence, disappearance, and reoccurrence of classes). The empirical analysis confirmed undersampling's dependability, and CBCE demonstrated that it beats other recognised class evolution adaptation algorithms, not only in terms of the ability to adjust to varied evolution scenarios but also in terms of overall classification performance. Two learning algorithms were proposed by Wang et al. in [18]. Undersampling-based Online Bagging (UOB) and Oversampling-based Online Bagging (OOB) devised an ensemble approach to overcome the class imbalance in real-time using time-decayed and resampling metrics. The study also focused on the performance of OOB and UOB's resampling strategies in both static and dynamic data streams to see how they could be improved. In terms of data distributions, imbalance rates, and changes in class imbalance status, their work provides the first comprehensive examination of class imbalance in data streams. According to the findings, UOB is superior at detecting minority class cases in static data streams, while OOB is better at resisting fluctuations in class imbalance status. The supply of data was discovered to be a crucial element impacting their performance, and more research was required.

Liu and Wu experimented with two strategies to avoid the drawbacks of employing undersampling to deal with class imbalance. When undersampling is used, most majority classes are disregarded, which is a flaw. As a result, Easy-Ensemble and Balance-Cascade were proposed in the study. Easy-Ensemble breaks the majority class into numerous smaller chunks then uses each chunk to train the learner independently, and finally, all of the learners' outputs are combined [19]. Balance-Cascade employs a sequential training strategy, in which the majority class's properly classified instances are excluded from further evaluation in the next series [19]. According to the data, the Easy-Ensemble and the Balance-Cascade had higher G-mean, F-measure, and AUC values than other existing techniques.

Many studies have been conducted on class disparity; nonetheless, the efficacy of most existing technologies in detecting credit card fraud is still far from optimal. The goal of this research paper was to see how employing random forest and a hybrid data-point strategy integrating feature selection and Near Miss may help enhance the classification performance of two credit card datasets. Near Miss is an undersampling technique that aims to stabilize class distribution by randomly deleting majority class examples [20].

In general, four techniques handling the problem of class imbalance have been proposed in the literature. Ensemble approaches, algorithm approaches, cost-sensitive approaches, and data-level approaches are examples of these methodologies. In the algorithmic technique, learning algorithms that are supervised are designed to favour the instances of the minority class. The most often used data-level methods rebalance the imbalanced dataset. By establishing misclassification costs, cost-sensitive algorithms solve the data imbalance problem. Undersampling and learning that are cost sensitive, bagging and undersampling, boosting, and resampling are some of the tactics used in ensemble learning approaches. In addition to these methods, hybrid approaches such as UnderBagging, OverBagging, and SMOTEBoost combine undersampling and oversampling methods [21].

In undersampling findings, the most suitable representation is important for the accurate prediction of the supervised learning algorithms on the imbalanced dataset. Clustering provides a useful representation of the majority class in a class imbalance problem. To deal with uneven learning, Onan in [21] employed a consensus clustering-based undersampling method. He employed k-modes, k-means, k-means++, self-organizing maps, and the DIANA method, as well as their combinations. The data were categorised using five supervised learning algorithms, support vector machines, logistic regression, naive Bayes, random forests, and the k-nearest neighbour algorithm, as well as three ensemble learner methods, AdaBoost, Bagging, and the random subspace algorithm. The clustering undersampling strategy produced the best prediction results [21].

Onan and Korukoğlu in [22] introduced an ensemble technique to sentiment classification feature selection. The proposed aggregation model aggregates the lists from several feature selection methods utilizing a genetic algorithm-based rank aggregation. The selection methods used were filter-based. This method was efficient and outperformed individual filter-based feature selection methods. In another sentiment analysis grouping study by Onan in [23], linguistic inquiry and word count were used to extract psycholinguistic features from text documents. Four supervised learning algorithms and three ensemble learning methods were used for the classification. The datasets contained positive, negative, and neutral tweets. 10-fold cross validation was employed.

Borah and Gupta in [24] suggested a robust twin bounded support vector machine technique based on the truncated loss function to overcome the imbalance problem. The total error of the classes was scaled based on the number of samples in each class to implement cost-sensitive learning.

In resolving the problem of class imbalance, Gupta and Richhariya in [25] presented entropy-based fuzzy least squares support vector machine and entropy-based fuzzy least squares twin support vector machine. Fuzzy membership was calculated on entropy values of samples. In another study by Gupta et al. in [26], a new method was referred to as fuzzy Lagrangian twin parametric-margin support vector machine which used fuzzy membership values in decision learning to handle outlier points. Hazarika and Gupta in [27] used a support vector machine based on density weight to handle the imbalance of classes problem. A weight matrix was used to reduce the effect of the binary class imbalance.

3. Materials and Methods

3.1. The Data-Point Approach. The data-point approach was used to investigate the class imbalance problem. The study proposed a 2-step hybrid data-point approach. The first step was using feature selection after data preprocessing and then undersampling with Near Miss to resample the data. Feature selection is the process of selecting those features that most contributed to the prediction variable or intended output [4].

3.2. Feature Selection. Feature selection was used as a step following preprocessing before the learning occurred. To overcome the drawbacks of an imbalanced distribution and improve the efficiency of classifiers, feature selection is used to choose appropriate variables. We performed feature selection using correlation coefficients which is a filter-based feature selection method that removes duplicate features, hence choosing the most relevant features. The feature selection was then utilized to determine which features were independent and which were dependent. The independent features were recorded in the X variable, while the dependent features were saved separately on the Y variable. The Y variable included the indicator of whether the transaction was normal (labeled as 0) or fraudulent (labeled as 1), which was the variable we were seeking to forecast. In this study, the class imbalance was investigated in the context of a binary (two-class) classification problem, with class 0 representing the majority and class 1 representing the minority.

3.3. Near Miss-Based Undersampling. The technique of balancing the class distribution for a classification dataset with a skewed class distribution is known as undersampling [28, 29]. To balance the class distribution, undersampling removes the training dataset examples which pertain to the majority class, such as reducing the skew from a 1 : 100 to a 1 : 10, 1 : 2, or even a 1 : 1 class distribution. To evaluate the influence of the data-point method, this paper used an undersampling strategy based on the Near Miss method. Near Miss was chosen based on its advantages to provide a more robust and fair class distribution boundary, which was found to improve the performance of classifies for detection in large-scale imbalanced datasets [30, 31]. The experiment used an imbalance-learn library, to call a class to perform

undersampling based on the Near Miss technique [7]. The Near Miss method was manipulated by passing parameters that are to meet the desired results. The Near Miss technique has three versions, namely [32],

(1) NearMiss-1 finds the test data with the least average distance to the negative class's nearest samples.

(2) NearMiss-2 chooses the positive samples with the shortest average distance to the negative class's farthest samples.

(3) NearMiss-3 is a two-step procedure. First, the nearest neighbours of each negative sample shall be preserved. The positive samples are then chosen based on the average distance between them and their nearest neighbours.

The presence of noise can affect NearMiss-1 when undersampling a specific class. It means that samples from the desired class will be chosen in the vicinity of these samples [33]. However, in most cases, samples around the limits will be chosen [34]. Because NearMiss-2 focuses on the farthest samples rather than the closest, it will not have this effect. The presence of noise can also be changed by sampling, especially when there are marginal outliers. Because of the first-step sample selection, NearMiss-3 will be less influenced by noise [35].

The following table is a snippet of parameters that were used to instantiate the Near Miss technique. The chosen variation for this study was the NearMiss-2 version after executing multiple iterations using all the three different versions to select the most suitable version for the credit card dataset. A uniform experiment was conducted on both datasets to ensure a fair cross-comparison.

Table 1 is a snippet of parameters that were used to instantiate the Near Miss technique.

Table 1 provides a list of all the parameters and their associated values, which were passed when instantiating the Near Miss method using an API call on the imbalance-learn library. Performance of the Near Miss method was optimized using parameter tuning, which was achieved by changing the default parameters for the version, *N* neighbours, and *N* neighbours' ver3 parameters.

Table 1: Near Miss method call parameters.

Parameter	Value
Sampling strategy	Auto
Return indices	False
Random state	None
Version	2
N neighbours	3
N neighbours ver3	3
N jobs	1
Ratio	None

3.4. Design of Study. The experimental method was used to examine the effect of using a hybrid data-point approach to solve the misclassification problem created by imbalanced datasets. The hybrid data-point technique was used on two imbalanced credit card datasets. This study investigated the undersampling technique instead of the oversampling technique because it balances the data by reducing the majority class. Therefore, undersampling avoids cloning the sensitive financial data, which means that only the authentic financial records were used during the experiment [36].

A lot of pieces of literature support undersampling, for example, a study by West and Bhattacharya in [3] found that undersampling gives better performance when the majority class highly outweighs the minority class. A cross-comparison amongst the two datasets was conducted to determine whether Near Miss-based undersampling could cater for distinct credit card datasets. The two datasets were collected from Kaggle, a public dataset source (https://www.kaggle.com/mlg-ulb/creditcardfraud/home) [37,38]. The datasets were considered because they are labeled, highly unbalanced, and handy for the researcher because they are freely accessible, making them more suited to the research's needs and budget. The study applied a supervised learning strategy that used the classification technique. Supervised learning gives powerful capabilities for using machine language to classify and handle data [39].

To infer a learning method, supervised learning was employed with labeled data, which was a dataset that had been classified. The datasets were used as the basis for predicting the classification of other unlabeled data using machine learning algorithms. The classification strategies utilized in the trials were those that focused on assessing data and recognising patterns to anticipate a qualitative response [40]. During the experiment, the classification algorithms were used to distinguish between the legitimate and fraudulent classes.

The experiment was executed in four stages: pretest stage, treatment stage, posttest stage, and review stage. During the pretest, the original dataset was fed into the machine learning classifiers and classification algorithms were used to train and test the predictive accuracy of the classifier. Each dataset was fed in the ML classifier using the 3-step loop: training, testing, and prediction. The data-point level approach methods were applied to the dataset during the treatment stage of the experiment to offset the area affected by class imbalance. The study investigated the hybrid technique to determine the resampling strategy that yields the best results. The resultant dataset from each procedure was the stage's output.

In the posttest stage, the resultant dataset was taken and again fed into the classifiers. Stages two and three were an iterative process; the aim was to solve the misclassification problem created by imbalanced data. Therefore, an in-depth review and analysis of accuracy for each result were conducted after each iteration to optimize the process for better accuracy. Lastly, the review stage carried out a comprehensive review of the performance of each algorithm for both the pretest and posttest results. Then, a cross-comparison of the two datasets was performed to determine the best performing algorithm for both datasets.

This supervised machine learning study was carried out with the help of Google Colab and the Python programming language. Python is suited for this study because it provides concise and human-readable code, as well as an extensive

choice of libraries and frameworks for implementing machine learning algorithms, reducing development time [37]. The code was performed on the Google Colab notebook, which runs on a Google browser and executes code on Google's cloud servers, using the power of Google hardware such as GPUs and Tensor Processing Units (TPUs) [38]. A high level hybrid data-point approach is presented in Algorithm 1.

3.5. Datasets. The first dataset included transactions from European cardholders, with 492 fraudulent activities out of a total of 284807 activities. Only 0.173 percent of all transactions in the sample were from the minority class, which were reported as real fraud incidents.

$$\text{Fraud}_{\text{cases}} = \frac{\text{fraud}}{\text{instance size}} * 100 = \frac{492}{284807} * 100 = 0.173\%. \tag{1}$$

Figure 1 shows a class distribution of the imbalanced European Credit Card dataset.

Figure 1 shows the bar graph representation of two classes found in the European cardholder's transactions. The *x*-axes represent the class, which indicates either normal or fraud. The *y*-axes represent the frequency of occurrence for each class. The short blue bar that is hardly visible shows the fraudulent transactions, which was the minority class. The figure shows a graphical representation of the imbalance ratio where the minority class accounts for 0.173% of the total dataset containing 284,807 transactions. The dataset has 31 characteristics. Due to confidentiality concerns, the primary components V1, V2, and up to V28 were translated using Principal Component Analysis (PCA); the only features not converted using PCA were "amount," "time," and "class." The 0 numeric value indicates a normal transaction and 1 indicates fraud in the "class" feature [14].

The second dataset called the UCI Credit Card dataset, which spanned from April 2005 to September 2005, comprises data on default payments, demographic variables, credit data, payment history, and bill statements for credit card clients in Taiwan [38]. The dataset is imbalanced and contains 30,000 instances, whereby there are 6,636 positive cases.

$$\text{Fraud}_{\text{cases}} = \frac{\text{fraud}}{\text{instance size}} * 100 = \frac{6636}{30000} * 100 = 22.12\%. \tag{2}$$

Figure 2 shows a class distribution for the imbalance UCI Credit Card datasets. The minority class caters for 22.12% of the total distribution containing 30,000 instances.

Figure 2 shows the UCI Credit Card dataset class distribution.

The short blue bar is the minority class that caters for 22.12% of the dataset and represents the credit card defaulters. The longer blue bar shows the normal transactions, which is the majority class. The UCI Credit Card dataset has 24 numeric attributes, which makes the dataset suitable for a classification problem. An attribute called "default.payment.next.month" contained the values of either 0 or 1. The "0" represents a legitimate case and the value of "1" represents the fraudulent case [38].

There were no unimportant values or misplaced columns in any of the datasets that were validated. To better understand the data, an exploratory data analysis was undertaken. After that, we used a class from the sklearn package to execute the train-test-split function to split the data into a training and testing set with a 70 : 30 ratio [41].

$$X\_\text{train}, X\_\text{test}, y\_\text{train}, y\_\text{test} = \text{train\_test\_split}(X, Y, \text{test\_size} = 0.30). \tag{3}$$

The dependent variable *Y*, independent variable *X*, and test size are all accepted by the train-test-split function. The test-size option indicates the split ratio of the dataset's original size, indicating that 30% of the dataset was used to test the model and 70% of the dataset was used to train the model. The experiment's next step was to create and train our classifiers. To create the classifiers, we employed each of the chosen algorithms. After that, each classifier was fitted using the *x*-train and y-train training data. The *x*-test data was then utilized to try to predict the *y*-test variable. The next section discusses the algorithms and classifiers in more detail.

3.6. Classification Algorithms. For the experiment, logistic regression, support vector machine (SVM), decision tree, and random forest algorithms were chosen. The literature revealed that decision tree, logistic regression, random forest, and SVM algorithms are the leading classical state-of-the-art detection algorithms [42–44]. The algorithms were used to train and validate the fraud detection model, following the train, test, and predict technique [45].

3.7. Performance Metrics. The measurement matrices used to evaluate the accuracy of the performance are the precision, recall, *F*1-score, average precision (AP), and confusion matrix [14, 46]. Precision is a metric that assesses a model's ability to forecast positive classifications [47–49]. Precision = TP/(TP + FP). "When the actual outcome is positive, recall describes how well the model predicts the positive class" [50]. Recall = TP/(TP + FN). Askari and Hussain in [48] claimed that utilizing both recall and precision to quantify the prediction powers of the model is beneficial. An in-depth review and analysis of accuracy was conducted using the following evaluation matrix: false-positive rate = FP/FP + TN, true-positive rate = TP/TP + FN, true-negative rate = TN/TN + FP, and false-negative rate = FN/FN + TP. A precision-recall curve is a plot of the precision (*y*-axis) and the recall (*x*-axis) for different thresholds [51].

Step 1: begin
Step 2: for $i = 1$ to k do begin
r = calculate correcoeff (n)
End
Step 3: data-point level approach—Near Miss undersampling
Find the distances between all instances of the majority class and the instances of the minority class
The majority class is to be undersampled
Then, n instances of the majority class that have the smallest distances to those in the minority class are selected
If there are k instances in the minority class, the nearest method will result in $k * n$ instances of the majority class
Step 4: train test split—split the data into a training set and a testing set using a (70 : 30) split ratio
Step 5: model prediction—for random forest model
Train the model by fitting the training set
Model evaluation (predict values for the testing set)
Step 6: output:
Analyze using performance metrics
Step 7: end

Algorithm 1: Hybrid data-point approach algorithm.

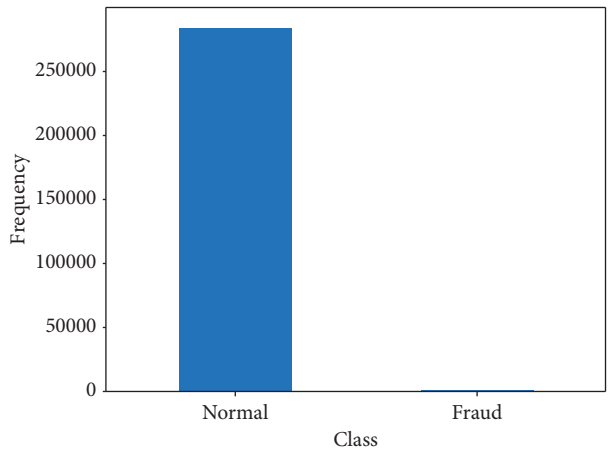

Figure 1: European Credit Card dataset.

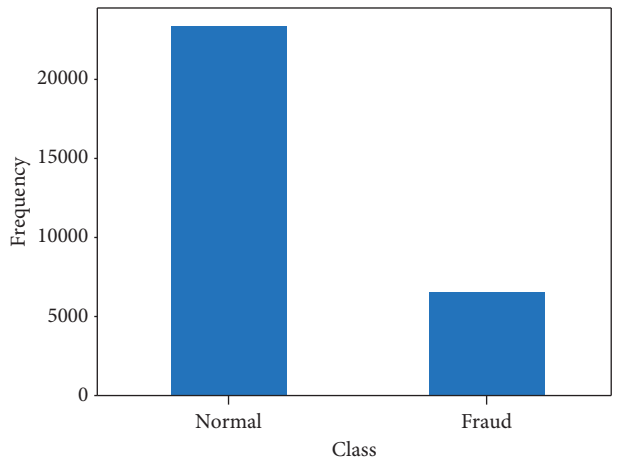

Figure 2: UCI Credit Card dataset distribution.

The $F1$-score is the test's precise measurement. When computing the $F1$-score, both the precision and recall scores are taken into account. "A confusion matrix is a table that shows how well a classification model works on a set of test data with known true values" [52].

4. Presentation of Results

This section presents a detailed report, comparison, and discussion of the results for both the European Credit Card dataset and the UCI Credit Card dataset. The performance metrics used to evaluate the accuracy of the performance are precision, recall, $F1$-score, average precision (AP), and confusion matrix. The results are shown for both the negative class (N) and the positive class (P).

4.1. Pretreatment Test Results. After samples of the original datasets were split into training datasets and testing datasets using a 70 : 30 ratio, the testing dataset was fed into the machine learning classifiers using each of the four algorithms that have been mentioned above to train and test the predictive performance of the classifier.

Table 2 shows the European Credit Card dataset results of the classification results before using undersampling.

The testing dataset for the European Credit Card dataset enclosed a sample size of 8,545 cases and the UCI Credit Card dataset contained a sample size of 900 cases. According to the classification report, there was a 100% accuracy from all the classifiers with the European Credit Card dataset, which is highly misleading. Looking only at the accuracy score with imbalance datasets does not reflect the true outcome of the classification. Focusing on the European Credit Card dataset classification, we can observe that, for the SVM classifier, there was a high bias towards the negative classes.

All 8545 cases were flagged as legitimate transactions; this is because there were only 17 fraudulent transactions in the testing dataset. The logistic regression performed better than the SVM and the classifier was biased, but looking at the precision, recall, and $F1$-score, some positive classes were able to be classified. The $F1$-score verifies that the test was not accurate. The report does not tell us if the positive classes identified were true positives or false positives, even though the recall score indicates that there was a great deal of misclassification. False positives and false negatives are the most common misclassification problems, which means that even though the classifier has 100% accuracy and can predict both positive and negatives classes, it fails to produce a successful prediction. A similar observation is seen on the decision tree and random forest, although the random forest performed much better compared to all the other three classifiers.

Table 3 shows the UCI Credit Card dataset results of the classification results before undersampling was used.

The classification report on the UCI Credit Card dataset shows similar results. The SVM classifier was 100% biased as seen with the European Credit Card dataset. The UCI Credit Card testing datasets have a lower imbalance ratio; there were 202 positive cases out of the total sample size. The accuracy recorded was 78%, which is far less than the ideal for a binary classification solution. Therefore, without even considering the bias and misclassification problem, the accuracy score alone shows that the SVM classifier is not consistent across multiple datasets. The logistic regression had an accuracy score of 78%, which is the same as the SVM classifier.

The major difference is the precision score, which was 100% for the logistic regression, implying that the classifier was able to predict all the positive classes. Therefore, we look at the recall score of 1%, and based on this value, we can conclude that the classifier was poor when the actual outcome was positive, which means that there were a lot of false positives and false negatives. Based on the precision score, we can conclude that the classifier is unbiased but the prediction was able to eliminate false positives and false negatives. The decision tree was the least effective in terms of the accuracy score, which was 72%. The precision, recall, and $F1$-score were all 37% for the positive class. The random forest continued to lead with an accuracy score of 81%. The precision was 63%. Recall and $F1$-score show that nearly half of the predicted was false positives. The initial finding reveals that there was a bias towards predicting the majority class, representing normal transactions.

4.1.1. The Confusion Matrix. "The confusion matrix table provides a mapping of the rate of true negative (TN), true positive (TP), false negative (FN), and false positive (FP)" [53, 54]. The following tables provide the results for each algorithm on the original dataset after using undersampling. The confusion matrix table is useful to quantify the number of misclassifications for both the negative and positive classes [55]. The total sample size used during testing is the sum of TN, FN, TP, and FP as per the blueprint of the confusion matrix. The confusion matrix also helps understand if the classification was biased [56]. The initial finding reveals that there was a prejudice towards predicting the majority class, representing normal transactions.

4.1.2. Import from sklearn.metrics. The confusion matrix class was introduced from sklearn using the snippet "from sklearn.metrics Import Confusion_matrix" and given that the dataset was labeled for both datasets, the parameters that indicate both class 0 and class 1 were already defined, and during data preprocessing, the parameter was stored in a prediction variable Y.

Table 4 shows the confusion matrix table (s) blueprint. The blueprint was used to present the classification results.

4.1.3. The Confusion Matrix without Undersampling. Table 5 shows the SVM confusion matrix results before undersampling was used to handle class imbalance.

The findings show that the classification was 100% biased to the majority class for both datasets. All the cases were predicted to be legitimate even though there was a total of 17 and 202 positive cases in both samples, respectively.

Table 2: Performance of imbalance European Credit Card dataset.

Classifier	Measure	N		P
SVM	Precision	1.00		0.00
	Recall	1.00		0.00
	$F1$-score	1.00		0.00
	Accuracy		1.00	
	Average precision		0.00	
Logistic regression	Precision	1.00		0.57
	Recall	1.00		0.47
	$F1$-score	1.00		0.52
	Accuracy		1.00	
	Average precision		0.48	
Decision tree	Precision	1.00		0.50
	Recall	1.00		0.47
	$F1$-score	1.00		0.48
	Accuracy		1.00	
	Average precision		0.24	
Random forest	Precision	1.00		0.90
	Recall	1.00		0.53
	$F1$-score	1.00		0.67
	Accuracy		1.00	
	Average precision		0.66	

Table 3: Performance of imbalance UCI Credit Card dataset.

	Measure	N		P
SVM	Precision	0.78		0.00
	Recall	1.00		0.00
	$F1$-score	0.87		0.00
	Accuracy		0.78	
	Average precision		0.22	
Logistic regression	Precision	0.78		1.00
	Recall	1.00		0.01
	$F1$-score	0.87		0.02
	Accuracy		0.78	
	Average precision		0.36	
Decision tree	Precision	0.82		0.37
	Recall	0.82		0.37
	$F1$-score	0.82		0.37
	Accuracy		0.72	
	Average precision		0.28	
Random forest	Precision	0.84		0.63
	Recall	0.94		0.36
	$F1$-score	0.88		0.46
	Accuracy		0.81	
	Average precision		0.37	

Table 6 shows the logistic regression confusion matrix results before undersampling was used to handle class imbalance.

The results show that the classifier was both biased and highly inaccurate. For example, out of a testing sample of 900 cases for the UCI Credit Card dataset, 94% of negative cases were correctly classified and only 37% of positive cases were correctly classified. The European cardholders' transactions dataset had a testing sample of 8545 transactions; 99.9% of negative cases were correctly classified and 47% of the positive cases were correctly classified.

Table 7 shows the decision tree confusion matrix results before undersampling was used to handle class imbalance.

The UCI dataset testing sample contained 698 negative cases and 202 positive cases. The total number of cases predicted as negative equals 700 and 200 for the positive cases. Looking at the prediction, we can assume that the model was accurate. However, the confusion matrix revealed that 128 of the 700 cases were falsely classified, and 126 of the 200 were falsely classified. A similar observation is made with the European cardholders' transactions dataset.

Table 4: Confusion matrix table (s) blueprint.

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

Table 5: Confusion matrix of the SVM classifier.

	European Credit Card dataset			UCI Credit Card dataset	
SVM	Predicted 0	Predicted 1	SVM	Predicted 0	Predicted 1
Actual 0	8528	0	Actual-0	698	0
Actual 1	17	0	Actual-1	202	0

Table 6: Confusion matrix of the logistic regression classifier.

	European Credit Card dataset			UCI Credit Card dataset	
LR	Predicted 0	Predicted 1	LR	Predicted 0	Predicted 1
Actual 0	8520	8	Actual 0	572	126
Actual 1	9	8	Actual 1	128	74

Therefore, even though there was minimum bias with the decision tree, the model was highly inaccurate.

Table 8 contains the random forest confusion matrix results before undersampling.

The confusion matrix for the random forest was both biased and highly inaccurate. For example, out of a testing sample of 900 cases for the UCI Credit Card dataset, 94% of negative cases were correctly classified and only 36% of positive cases were correctly classified. The European cardholders' transactions dataset had a testing sample of 8,545 transactions; 99.9% of negative cases were correctly classified and 53% of positive cases were correctly classified.

4.2. Posttreatment Test after Undersampling. The next phase of the experiment was to apply the data-point level approach methods on the dataset, whereby, to counteract the effect of the class imbalance, undersampling was applied. The Near Miss technique was used to undersample the majority instances and made them equal to the minority class. The class with majority has been decreased to the total number of records in the minority class, resulting in an equal number of records for both classes. The treatment stage was an iterative process; the aim was to solve the problem of imbalanced data; therefore, an in-depth review and analysis were conducted after each iteration to optimize the process.

Table 9 shows the European Credit Card results for the classification of the imbalanced datasets before application of the undersampling with Near Miss technique.

The dataset was balanced with a subset containing a sample size of 98 instances evenly distributed between the two classes, namely, normal and fraudulent transactions. The accuracy score for the SVM classifier decreased from 1.00 to 0.73. However, the ability to predict positive classes improved, and the precision score for the positive class increased from 0.00 to 1.00, a 100% improvement. The recall score increased from 0.00 to 0.47, an improvement of 47%, which means that the SVM classifier could predict true positives after undersampling with Near Miss, even though the percentage achieved is not ideal. The $F1$-score also increased from 0.00 to 0.64, and the improvement verifies the accuracy of the test. The logistic regression reported an accuracy score of 90%, which is a decrease of 10% compared to the results achieved before undersampling. However, the average precision increased from 0.48 to 0.87, which is an increase of 39%.

The increase in average precision reveals that even though accuracy decreased, the overall predictive accuracy increased. The increase in predictive accuracy is observed by the increase in precision, recall, and $F1$-score for positive classes. Precision increased from 0.57 to a decent 0.93, recall increased from 0.47 to 0.87, and the $F1$-score increased from 0.52 to 0.90 for the positive class. The negative class performed fairly well too, even though the initial 100% accuracy was not achieved, and the classifier was not biased on either class. The precision was 0.88, the recall was 0.93, and the $F1$-score was 0.90 for the negative class. The random forest classification was similar to the logistic regression, which also reported an accuracy of 90%. The precision was 0.83 for the negative class and 1.00 for the positive class. The recall was 1.00 for the negative class and 0.80 for the positive class. The $F1$-score was 0.91 for the negative class and 0.89 for the positive class.

The random forest performed better than all other classifiers before using undersampling but was closely matched by the decision tree in second place. However, the decision tress surpassed the random forest and gave the best results after undersampling with Near Miss. The decision tree maintained an accuracy score of 100% and the average precision increased from 28% to 100%. The precision, recall, and $F1$-score for both the negative and positive classes were impressive 100%. Based on these results, the classification report of the European Credit Card dataset after undersampling with Near Miss to solve the imbalance problem showed a significant improvement in the ability to predict fraudulent transactions.

Table 7: Confusion matrix of the decision tree classifier.

	European Credit Card dataset			UCI Credit Card dataset	
DT	Predicted 0	Predicted 1	DT	Predicted 0	Predicted 1
Actual 0	8520	8	Actual 0	572	126
Actual 1	9	8	Actual 1	128	74

Table 8: Confusion matrix of the random forest classifier.

	European Credit Card dataset			UCI Credit Card dataset	
RF	Predicted 0	Predicted 1	RF	Predicted 0	Predicted 1
Actual 0	8527	1	Actual 0	656	42
Actual 1	8	9	Actual 1	129	73

Table 9: Performance of the European Credit Card dataset.

Classifier	Measure	*N*		*P*
SVM	Precision	0.65		1.00
	Recall	1.00		0.47
	*F*1-score	0.79		0.64
	Accuracy		0.73	
	Average precision		0.73	
Logistic regression	Precision	0.88		0.93
	Recall	0.93		0.87
	*F*1-score	0.90		0.90
	Accuracy		0.90	
	Average precision		0.87	
Decision tree	Precision	1.00		1.00
	Recall	1.00		1.00
	*F*1-score	1.00		1.00
	Accuracy		1.00	
	Average precision		1.00	
Random forest	Precision	0.83		1.00
	Recall	1.00		0.80
	*F*1-score	0.91		0.89
	Accuracy		0.90	
	Average precision		1.00	

Table 10 shows the UCI Credit Card dataset results of the classification results before undersampling was used.

The SVM reported an accuracy score of 85%, which is an increase of 7% compared to the accuracy achieved before undersampling. The ability to predict the positive class improved as the average precision increased from 0.22 to 0.84, an improvement of 62%. The logistic regression accuracy decreased from 0.78 to 0.73. However, the average precision improved from 0.36 to 0.79. These results show that the logistic regression improved its ability to predict positive classes. The decision tree reported an improved accuracy of 85%, and the accuracy increased from 0.72 to 0.85.

The average precision also increased from 0.28 to 0.81, an improvement of 53%. The random forest reported an accuracy score of 89%, which was the highest out of the four classifiers. The average precision also increased from 0.37 to 0.86, an improvement of 49%. All the classifiers reported improved precision, recall, and *F*1-score after using undersampling. The classification report for the UCI Credit Card dataset revealed that there was an overall improvement in the ability to predict positive classes.

4.2.1. The Confusion Matrix with the Data-Point Approach. Table 11 contains the SVM confusion matrix after undersampling with Near Miss.

Even though some confusion level still exists, the effect of Near Miss was observed on both datasets. The ability to predict positive cases improved by 46% on the European Credit Card dataset and improved by 73% on the UCI Credit Card dataset. The SVM confusion matrix showed improvement in the ability to predict positive classes.

Table 12 shows the confusion matrix of the logistic regression after undersampling with Near Miss.

There was 100% predictive accuracy for negative cases and 87% for positive cases on the European cardholders' transactions. The UCI Credit Card dataset had an accuracy of 80% for negative classes and 66% for positive classes. The confusion matrix for the logistic regression model also

Table 10: Performance of imbalance UCI Credit Card dataset.

Classifier	Measure	N		P
SVM	Precision	0.77		0.96
	Recall	0.97		0.73
	$F1$-score	0.86		0.83
	Accuracy		0.85	
	Average precision		0.84	
Logistic regression	Precision	0.70		0.76
	Recall	0.79		0.66
	$F1$-score	0.74		0.71
	Accuracy		0.73	
	Average precision		0.79	
Decision tree	Precision	0.85		0.86
	Recall	0.85		0.86
	$F1$-score	0.85		0.86
	Accuracy		0.85	
	Average precision		0.81	
Random forest	Precision	0.86		0.92
	Recall	0.92		0.86
	$F1$-score	0.89		0.89
	Accuracy		0.89	
	Average precision		0.86	

shows that the Near Miss technique worked well for both datasets.

Table 13 contains the decision tree confusion matrix after undersampling with Near Miss.

There was no confusion with 100% accuracy for both classes on the European cardholders' transactions dataset. That means the ability to predict positive classes improved by 47% after undersampling with Near Miss. Therefore, using the Near Miss technique with the decision tree produced the best results with the European cardholders' transactions dataset. There was 85% accuracy for negative classes and 86% accuracy for positive classes on the UCI Credit Card dataset.

Table 14 shows that there was a predictive accuracy of 100% on the European cardholders' transactions dataset and 92% on the UCI Credit Card dataset for negative cases, respectively. There was a predictive accuracy of 80% and 86%, respectively, on both datasets for positive cases. The random forest also performed well.

4.3. The Precision-Recall Curve. The prediction score was used to calculate the average precision (AP). At each threshold, the weighted mean of precisions achieved, with the increase in recall from the preceding threshold used as the weight, is how AP summarizes a precision-recall curve [55]:

$$\mathrm{AP} = \sum_{n} * \left(R_n - R_{n-1}\right)P_n. \tag{4}$$

The average precision is calculated using the method above where P_n and R_n are the precision and recall at the nth threshold, respectively, and precision and recall are always in the range of zero to one. As a result, AP falls between 0 and 1. AP is a metric used to quantify the accuracy of a classifier; the closer the number is to 1, the more accurate the classifier is. A precision-recall ($P - R$) curve is a graph comparing precision (y-axis) with recall (x-axis) for various thresholds. In circumstances where the distribution between the two classes is unbalanced, using both recall and precision to measure the model's prediction powers is beneficial [56].

The following graphs represent the $P - R$ curves for the random forest classifier on both datasets, namely, the European Credit Card dataset and the UCI Credit Card dataset. The $P - R$ curve was only presented to the best performing algorithm for further analysis. The goal was to see if the $P - R$ curve was pointing towards the chart's upper right corner. The higher the quality is, the closer the curve comes to the value of one in the upper right corner.

4.3.1. The Precision-Recall Curve without Near Miss. Figure 3 shows the European Credit Card dataset precision-recall curve for random forest before the data-point approach.

The random forest precision-recall curve for the European Credit Card dataset starts straight across the highest point and halfway through gradually start curving towards the lower right corner. The average precision was 0.66.

Figure 4 shows the UCI Credit Card dataset precision-recall curve for random forest before the data-point approach.

The random forest P-R curve for the UCI Credit Card dataset gradually leaned towards the lower right corner from the beginning. The average precision was 0.37 and this can be observed on the P-R curve. The performance was better on European Credit Card dataset but was not consistent across both datasets. However, both the above results show poor quality in the ability to predict positive classes for both datasets. The P-R curve is a simple way to analyze the quality of a classifier without having to perform complex analysis. The next step was to apply the data-point approach and observe the change in quality.

Table 11: Confusion matrix of the SVM classifier with Near Miss.

	European Credit Card dataset			UCI Credit Card dataset	
SVM	Predicted 0	Predicted 1	SVM	Predicted 0	Predicted 1
Actual 0	15	0	Actual 0	191	6
Actual 1	8	7	Actual 1	56	153

Table 12: Logistic regression confusion matrix after undersampling with Near Miss.

	European Credit Card dataset			UCI Credit Card dataset	
LR	Predicted 0	Predicted 1	LR	Predicted 0	Predicted 1
Actual 0	15	0	Actual 0	161	42
Actual 1	2	13	Actual 1	69	134

Table 13: Confusion matrix of the decision tree after undersampling with Near Miss.

	European Credit Card dataset			UCI Credit Card dataset	
DT	Predicted 0	Predicted 1	DT	Predicted 0	Predicted 1
Actual 0	15	0	Actual 0	167	30
Actual 1	0	15	Actual 1	30	179

Table 14: Random forest confusion matrix after undersampling with Near Miss.

	European Credit Card dataset			UCI Credit Card dataset	
RF	Predicted 0	Predicted 1	RF	Predicted 0	Predicted 1
Actual 0	15	0	Actual 0	181	16
Actual 1	3	12	Actual 1	30	179

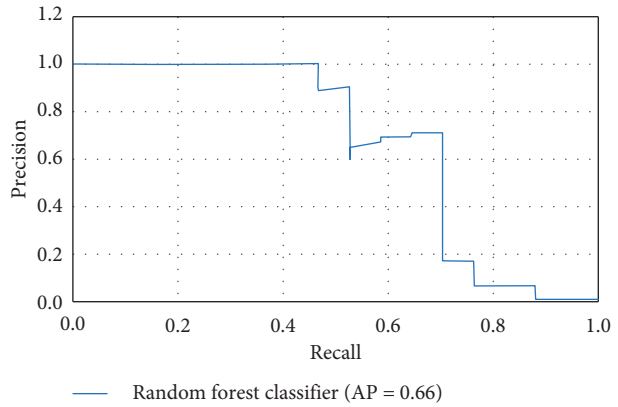

Figure 3: European Credit Card dataset (RF) precision-recall curve.

4.3.2. The Precision-Recall Curve with Near Miss. The figures below show the precision-recall curve after treatment using feature selection with the Near Miss-based undersampling technique was applied. A $P - R$ curve is a brilliant way to see a graphical representation of a classifier's quality. The $P - R$ curves show the improvement in the quality of the classifiers after using the data-point approach.

Figure 5 shows the European Credit Card dataset precision-recall curve for random forest before the data-point approach.

Figure 5 shows the random forest $P - R$ curve on the European Credit Card dataset. The classifier improved by 33% as the average precision increased from 0.66 and 1.00, indicated by the straight line on the value of 1 across the y-axis.

Figure 6 shows the UCI Credit Card dataset precision-recall curve for random forest before the data-point approach.

Figure 6 shows the random forest $P - R$ curve on the UCI Credit Card dataset. The curve starts straight on the

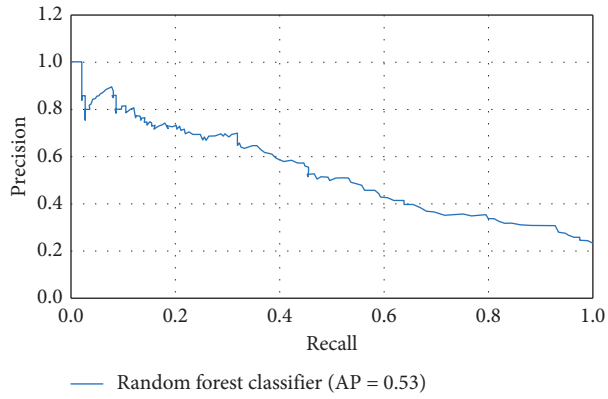

Figure 4: UCI Credit Card dataset (RF) precision-recall curve.

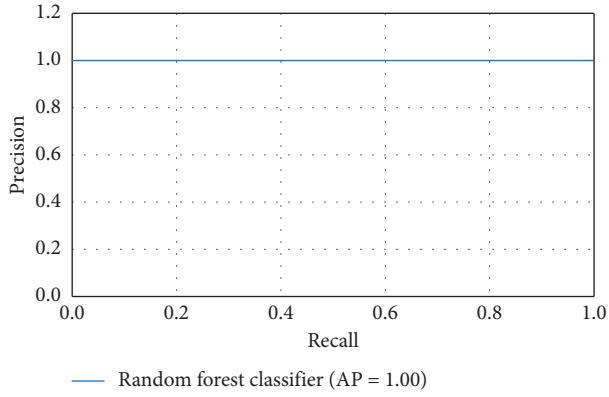

Figure 5: European Credit Card dataset (RF with Near Miss) precision-recall curve.

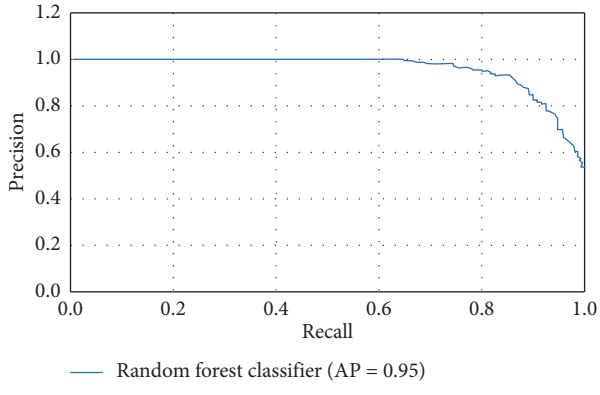

Figure 6: UCI Credit Card dataset (RF with Near Miss) precision-recall curve.

value of 1 on the y-axis, moving across the x-axis, and ends by a gentle fall while leaning towards the upper right corner. The average precision increased from 0.28 to 0.81. Both the results indicate great quality.

A $P - R$ curve that is a straight line on the y-axis value of 1 across the x-axis, such as Figure 5 of the random forest with the European Credit Card dataset, represents the best possible quality. A $P - R$ curve that is leaning more towards the upper right corner is also a sign that the classifier has good quality such as Figure 6 on the UCI Credit Card datasets.

5. Conclusions

All the algorithms scored an average score of 1.00 for legitimate cases with the European cardholder's credit card transactions dataset ($D1$) and an average score of 0.87 with the UCI Credit Card dataset ($D2$) for the precision, recall, and $F1$-score. These results indicate that the majority class was dominant due to the imbalance level, and the challenge is successfully anticipating the minority class.

Recording an average precision score of 0.77 and an average recall score of 0.45, the random forest model was the

best performer for detecting minority classes in the weighted average classification report with both original datasets. However, comparing both precisions and recall scores shows that the model did not perform well. The combined calculated average precision of 0.43 was used to further validate the model, indicating that it was not generating optimal results and that additional treatment was required. In both datasets, the SVM model performed the worst, with accuracy and recall scores of 0.00. Due to the uneven class distribution, the SVM model was biased and utterly failed to identify minority classes with a score of 0.00.

The average precision score for the positive class improved by 98% for SVM, 49.5% for decision tree, 19.5% for random forest, and 5.5% for logistic regression after utilizing undersampling with the Near Miss approach. The recall score for the positive class shows that that the strength of identifying true positive (which are actually fraudulent cases) improved by 60% for SVM, 51.5% for logistic regression, 51% for the decision tree, and 38.5% for random forest and improved their ability to identify true positive (fraudulent cases) by 60% for SVM, 51.5% for logistic regression, 51% for the decision tree, and 38.5% for random forest. *F*1-score improved by 73.5% for SVM, 52.5% for logistic regression, 50.5% for decision tree, and 32.5% for random forest in the positive class, according to the findings. When the capacity to detect affirmative classes was improved, the *F*1-score improved as well. After using the data-point approach, the predicting accuracy improved for all the algorithms on both datasets. Using a determined average score of accuracy, recall, and *F*1-score for each classifier, the random forest method is the leading algorithm. Ordered from best to worst, the performance of the machine learning techniques were as follows: random forest, decision tree, logistic regression, and SVM.

The findings reveal that when the data is significantly skewed, the model has difficulty detecting fraudulent transactions. There was a considerable improvement in the capacity to forecast positive classes after applying the hybrid data-point strategy combining feature selection and the Near Miss-based undersampling technique. Based on the findings, the hybrid data-point approach improved the predictive accuracy of all the four algorithms used in this study. However, even though there was a significant improvement on all classification algorithms, the results revealed that the proposed method with the random forest algorithm produced the best performance on the two credit card datasets.

The findings of this study can be used in future research to look at developing and deploying a real-time system that can detect fraud while the transaction is taking place.

Data Availability

The data on credit card fraud are available online at https://www.kaggle.com/mlg-ulb/creditcardfraud/home.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

The authors acknowledge the Durban University of Technology for making funding opportunities and materials for experiments available for this research project.

References

[1] Sabric Annual Crime Stats 2019. https://www.sabric.co.za/media-and-news/press-releases/sabric-annual-crime-stats-2019/.

[2] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi, "Credit card fraud detection using AdaBoost and majority voting," *IEEE Access*, vol. 6, no. 1, pp. 14277–14284, 2018.

[3] J. West and M. Bhattacharya, "Some experimental issues in financial fraud mining," *Procedia Computer Science*, vol. 80, no. 1, pp. 1734–1744, 2016.

[4] M. Wasikowski and X.-w. Chen, "Combating the small sample class imbalance problem using feature selection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1388–1400, 2010.

[5] W. Lin, Z. Wu, L. Lin, A. Wen, and J. Li, "An ensemble random forest algorithm for insurance big data analysis," *IEEE Access*, vol. 5, pp. 16568–16575, 2017.

[6] E. M. Hassib, A. I. El-Desouky, E.-S. M. El-Kenawy, and S. M. El-Ghamrawy, "An imbalanced big data mining framework for improving optimization algorithms performance," *IEEE Access*, vol. 7, no. 1, pp. 170774–170795, 2019.

[7] D. Chen, X.-J. Wang, C. Zhou, and B. Wang, "The distance-based balancing ensemble method for data with a high imbalance ratio," *IEEE Access*, vol. 7, no. 1, pp. 68940–68956, 2019.

[8] S. A. Shevchik, F. Saeidi, B. Meylan, and K. Wasmer, "Prediction of failure in lubricated surfaces using acoustic time-frequency features and random forest algorithm," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1541–1553, 2017.

[9] I. Sadgali, N. Sael, and F. Benabbou, "Performance of machine learning techniques in the detection of financial frauds," *Procedia Computer Science*, vol. 148, no. 1, pp. 45–54, 2019.

[10] A. Adedoyin, "Predicting fraud in mobile money transfer," *IEEE Transactions On Knowledge and Data Engineering*, vol. 28, no. 6, pp. 1–203, 2016.

[11] T. Hasanin, T. M. Khoshgoftaar, J. Leevy, and N. Seliya, "Investigating random undersampling and feature selection on bioinformatics big data," in *Proceedings of the 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (Big Data Service)*, pp. 346–356, Newark, CA, USA, November 2019.

[12] K. Sotiris, K. Dimitris, and P. Panayiotis, "Handling imbalanced datasets: a review," *GESTS International Transactions on Computer Science and Engineering*, vol. 30, no. 1, pp. 1–12, 2016.

[13] T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Supervised neural network modeling: an empirical investigation into learning from imbalanced data with labeling errors," *IEEE Transactions on Neural Networks*, vol. 21, no. 5, pp. 813–830, 2010.

[14] B. Pes, "Learning from high-dimensional biomedical datasets: the issue of class imbalance," *IEEE Access*, vol. 8, no. 1, pp. 13527–13540, 2020.

[15] G. Ditzler and R. Polikar, "Incremental learning of concept drift from streaming imbalanced data," *IEEE Transactions On*

Knowledge and Data Engineering, vol. 25, no. 10, pp. 2283–2301, 2013.

[16] Z. Zhou and X. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 63–77, 2006.

[17] L. L. Minku, S. Wang, and X. Yao, "Online ensemble learning of data streams with gradually evolved classes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 6, pp. 1532–1545, 2016.

[18] S. Wang, L. L. Minku, and X. Yao, "Resampling-based ensemble methods for online class imbalance learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 5, pp. 1356–1368, 2015.

[19] X. Liu, J. Wu, and Z. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 2, pp. 539–550, 2009.

[20] C. Jiang, J. Song, G. Liu, L. Zheng, and W. Luan, "Credit card fraud detection: a novel approach using aggregation strategy and feedback mechanism," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3637–3647, 2018.

[21] A. Onan, "Consensus clustering-based undersampling approach to imbalanced learning," *Scientific Programming*, vol. 2019, Article ID 5901087, 2019.

[22] A. Onan and S. Korukoğlu, "A feature selection model based on genetic rank aggregation for text sentiment classification," *Journal of Information Science*, vol. 43, no. 1, pp. 25–38, 2017.

[23] A. Onan, "Sentiment analysis on Twitter based on ensemble of psychological and linguistic feature sets," *Balkan Journal of Electrical and Computer Engineering*, vol. 6, no. 2, pp. 69–77, 2018.

[24] P. Borah and D. Gupta, "Robust twin bounded support vector machines for outliers and imbalanced data," *Applied Intelligence*, vol. 51, no. 3, pp. 1–30, 2021.

[25] D. Gupta and B. Richhariya, "Entropy based fuzzy least squares twin support vector machine for class imbalance learning," *Applied Intelligence*, vol. 48, no. 11, pp. 4212–4231, 2018.

[26] D. Gupta, P. Borah, and M. Prasad, "A fuzzy based Lagrangian twin parametric-margin support vector machine (FLTPMSVM)," in *Proceedings of the 2017 IEEE symposium series on computational intelligence (SSCI)*, pp. 1–7, IEEE, Honolulu, HI, USA, November 2017.

[27] B. B. Hazarika and D. Gupta, "Density-weighted support vector machines for binary class imbalance learning," *Neural Computing and Applications*, vol. 33, pp. 1–19, 2020.

[28] X. Zhang, C. Zhu, H. Wu, Z. Liu, and Y. Xu, "An imbalance compensation framework for background subtraction," *IEEE Transactions on Multimedia*, vol. 19, no. 11, pp. 2425–2438, 2017.

[29] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: a hybrid approach to alleviating class imbalance," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 1, pp. 185–197, 2010.

[30] L. Bao, C. Juan, J. Li, and Y. Zhang, "Boosted Near-miss under-sampling on SVM ensembles for concept detection in large-scale imbalanced datasets," *Neurocomputing*, vol. 172, no. 1, pp. 198–206, 2016.

[31] M. Peng, Q. Zhang, X. Xing et al., "Trainable undersampling for class-imbalance learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 4707–4714, 2019.

[32] Imbalanced-learn. 2020, https://imbalanced-learn.readthedocs.io/en/stable.

[33] L. Zheng, G. Liu, C. Yan, and C. Jiang, "Transaction fraud detection based on total order relation and behavior diversity," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 3, pp. 796–806, 2018.

[34] S. Patil, V. Nemade, and P. K. Soni, "Predictive modelling for credit card fraud detection using data analytics," *Procedia Computer Science*, vol. 132, no. 1, pp. 385–395, 2018.

[35] A. Tarjo and N. Herawati, "Application of beneish M-score models and data mining to detect financial fraud," *Procedia-Social and Behavioral Sciences*, vol. 211, no. 1, pp. 924–930, 2015.

[36] A. Somasundaran and U. S. Reddy, "Data imbalance: effects and solutions for classification of large and highly imbalanced data," *Proceedings of the 1st International Conference on Research in Engineering, Computers and Technology*, vol. 25, no. 10, pp. 28–34, 2016.

[37] Google Colaboratory Frequently Asked Questions. 2019, https://research.google.com/colaboratory/faq.html.

[38] Scikit Learn. 2020, https://scikit-learn.org/stable/supervised_learning.html.

[39] M. Albashrawi, "Detecting financial fraud using data mining techniques: a decade review from 2004 to 2015," *Journal of Data Science*, vol. 14, no. 1, pp. 553–570, 2016.

[40] G. Baader and H. Krcmar, "Reducing false positives in fraud detection: combining the red flag approach with process mining," *International Journal of Accounting Information Systems*, vol. 31, no. 1, pp. 1–16, 2018.

[41] C.-T. Su and Y.-H. Hsiao, "An evaluation of the robustness of MTS for imbalanced data," *IEEE Transactions On Knowledge and Data Engineering*, vol. 19, no. 10, pp. 1321–1332, 2007.

[42] R. Batuwita and V. Palade, "FSVM-CIL: fuzzy support vector machines for class imbalance learning," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 3, pp. 558–571, 2010.

[43] S. Subudhi and S. Panigrahi, "Quarter-sphere support vector machine for fraud detection in mobile telecommunication networks," *Procedia Computer Science*, vol. 48, no. 1, pp. 353–359, 2015.

[44] Z. Liu, T. Wen, W. Sun, and Q. Zhang, "Semi-supervised self-training feature weighted clustering decision tree and random forest," *IEEE Access*, vol. 8, pp. 128337–128348, 2020.

[45] E. Scornet, "Random forests and Kernel methods," *IEEE Transactions on Information Theory*, vol. 62, no. 3, pp. 1485–1500, 2016.

[46] J. T. Raj, "What to do when your classification data is imbalanced," 2019, https://towardsdatascience.com/what-to-do-when-your-classification-dataset-is-imbalanced-6af031b12a36.

[47] A. Shen, R. Tong, and Y. Deng, "Application of classification models on credit card fraud detection," in *Proceedings of the 2007 International Conference on Service Systems and Service Management*, pp. 1–4, Chengdu, China, June 2007.

[48] S. M. S. Askari and M. A. Hussain, "Credit card fraud detection using fuzzy ID3," in *Proceedings of the 2017 International Conference on Computing, Communication and Automation (ICCCA)*, pp. 446–452, Noida, India, May 2017.

[49] J. Li, H. He, and L. Li, "CGAN-MBL for reliability assessment with imbalanced transmission gear data," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 9, pp. 3173–3183, 2019.

[50] C.-C. Lin, D.-J. Deng, C.-H. Kuo, and L. Chen, "Concept drift detection and adaption in big imbalance industrial IoT data using an ensemble learning method of offline classifiers," *IEEE Access*, vol. 7, no. 1, pp. 56198–56207, 2019.

[51] J. Shao, "Linear model selection by cross-validation," *Journal of the American Statistical Association*, vol. 88, no. 422, pp. 486–494, 1993.
[52] P. Zhang, "On the distributional properties of model selection criteria," *Journal of the American Statistical Association*, vol. 87, no. 419, pp. 732–737, 1992.
[53] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: one-sided selection," *ICML*, vol. 97, no. 1, pp. 179–186, 1997.
[54] A. Rashad, S. Riaz, and L. Jiao, "Semi-supervised deep fuzzy C-mean clustering for imbalanced multi-class classification," *IEEE Access*, vol. 7, no. 1, pp. 28100–28112, 2019.
[55] J. Wei, Z. Lu, K. Qiu, P. Li, and H. Sun, "Predicting drug Risk level from adverse drug reactions using SMOTE and machine learning approaches," *IEEE Access*, vol. 8, no. 1, pp. 185761–185775, 2020.
[56] R. Yao, J. Li, M. Hui, L. Bai, and Q. Wu, "Feature selection based on random forest for partial discharges characteristic set," *IEEE Access*, vol. 8, Article ID 159151, 2020.