CIT. Journal of Computing and Information Technology, Vol. 28, No. 4, December 2020, 269–285
doi: 10.20532/cit.2020.1005216

An Effective Data Sampling Procedure for Imbalanced Data Learning on Health Insurance Fraud Detection

Shamitha S. Kotekani[1, 2] and Ilango Velchamy[1, 2]

[1]Department of Computer Applications, CMRIT, Bengaluru, India
[2]VTU, Belgavi, India

Fraud detection has received considerable attention from many academic research and industries worldwide due to its increasing popularity. Insurance datasets are enormous, with skewed distributions and high dimensionality. Skewed class distribution and its volume are considered significant problems while analyzing insurance datasets, as these issues increase the misclassification rates. Although sampling approaches, such as random oversampling and SMOTE can help balance the data, they can also increase the computational complexity and lead to a deterioration of model's performance. So, more sophisticated techniques are needed to balance the skewed classes efficiently. This research focuses on optimizing the learner for fraud detection by applying a Fused Resampling and Cleaning Ensemble (FusedRCE) for effective sampling in health insurance fraud detection. We hypothesized that meticulous oversampling followed with a guided data cleaning would improve the prediction performance and learner's understanding of the minority fraudulent classes compared to other sampling techniques. The proposed model works in three steps. As a first step, PCA is applied to extract the necessary features and reduce the dimensions in the data. In the second step, a hybrid combination of *k*-means clustering and SMOTE oversampling is used to resample the imbalanced data. Oversampling introduces lots of noise in the data. A thorough cleaning is performed on the balanced data to remove the noisy samples generated during oversampling using the Tomek Link algorithm in the third step. Tomek Link algorithm clears the boundary between minority and majority class samples and makes the data more precise and freer from noise. The resultant dataset is used by four different classification algorithms: Logistic Regression, Decision Tree Classifier, k-Nearest Neighbors, and Neural Networks using repeated 5-fold cross-validation. Compared to other classifiers, Neural Networks with FusedRCE had the highest average prediction rate of 98.9%. The results were also measured using parameters such as F1 score, Precision, Recall and AUC values. The results obtained show that the proposed method performed significantly better than any other fraud detection approach in health insurance by predicting more fraudulent data with greater accuracy and a 3x increase in speed during training.

ACM CCS (2012) Classification: Computer methodologies → Machine learning → Learning paradigms → Supervised learning → Supervised learning by classification

Computer methodologies → Machine learning → Machine learning approaches → Factorization methods → Principal component analysis

Keywords: health insurance, fraud detection, class imbalance, *k*-means, SMOTE, classification algorithms

1. Introduction

Health care facilities throughout the globe are evolving and will continue to grow. The growth in data has become a pressing concern with the adaptive nature of the industry. One of the critical challenges faced by healthcare systems is the possibility of "fraud". Fraud covers a series of inappropriate activities to gain unlawful advantage from health insurance companies. Actions related to fraud could originate from various parties like patients, doctors, pharmacists or other medical providers. The patterns of

fraud performed by each category of people are of the following ways:

- patients – type of fraudulent activities performed from patients-end include providing false information while submitting claims, misleading the insurance providers by providing wrong medical history, filing claims for those services that were not rendered, having identity thefts, *etc.*;

- providers – providers can be anyone, including doctors, pharmacists, contractors *etc.* According to M. E. Johnson *et al.* [1], doctors play a significant role in defining medical procedures or prescriptions for a patient. Providers may prescribe inappropriate services which have not been used or are not required;

- insurance companies – fraudulent patterns from insurance companies appear mainly in the form of denying genuine claims with an aim to optimize their expenditure.

Different patterns of fraud have been explained in various literature pieces such as upcoding, phantom billing, kickback schemes, wrong diagnosis, maximizing care, identity fraud, multiple billing, doctor shopping, self-referral, *etc.* [2], [3].. All these frauds eventually lead to an overburdened health insurance system. The traditional method for detecting fraud was by developing rules and manually checking each case against the rules. A score is given based on the match, and by aggregating these scores, an alarm will be raised stating the transaction as fraud. The main challenge in these approaches is that it is purely dependent on manual intervention; it also demands in-depth domain knowledge. Here comes the advantage of implementing machine learning algorithms for fraud detection. Without any prior judgment about the data, we can feed it to the classifiers to learn the data's hidden patterns. Classifiers are learners that identify classes based on the learning criteria, which are applied in many real-life scenarios such as cancer prediction, brain tumor image classification, cancer prediction, fraud detection, spam detection, *etc.* [1], [3], [4]. Though applying learners provides a more significant advantage in detecting unknown fraudulent patterns, there exist some practical issues to be addressed from insurance data while resolving the problem, such as:

- curse of dimensionality – insurance data sets are large and high dimensional, containing information from various sources such as patient and physician demographics, drug details, billing details, prescription details, *etc.*;

- skewed class distributions – there is a considerable variation between the ratio of fraudulent cases to non-fraudulent cases.

In health insurance, minority classes (fraudulent classes) are of utmost importance and need to be accurately predicted. If there lies an imbalance between the classes, the classifier will not produce accurate results as they tend to deviate towards the majority class. This scenario is interpreted as the "Class Imbalance Problem", where classes present in the dataset are unequal. In such an environment, classes with a lesser number of examples are called minority classes, and the classes with bigger number of examples are referred to as majority classes [5]. As explained, the distribution of classes plays a major role in effective classification. The fraud detection system contains a majority of non-fraudulent cases and a significantly lower percentage of fraudulent claims, which shows that the non-fraudulent classes outnumber the fraudulent classes [6], [7], [8]. Here, fraudulent classes are underrepresented when comparing to non-fraudulent classes. When this type of dataset is fed into a classifier, the classifier tends to be biased towards majority classes and may predict only majority classes.

The paper proposes a combination of Principal Component Analysis (PCA) and an ensemble resampling and cleaning technique to enhance the learner's predictive performance. The entire framework helped in reducing the misclassification costs and producing a better-generalized fraud detection model. The proceedings of the work are as follows; Section 2 reviews the works related to the domain and explains the research gaps; Section 3 details the design and methodology used throughout the study. Section 4 discusses and visualizes the outcomes the study, and finally, Section 5 concludes the work.

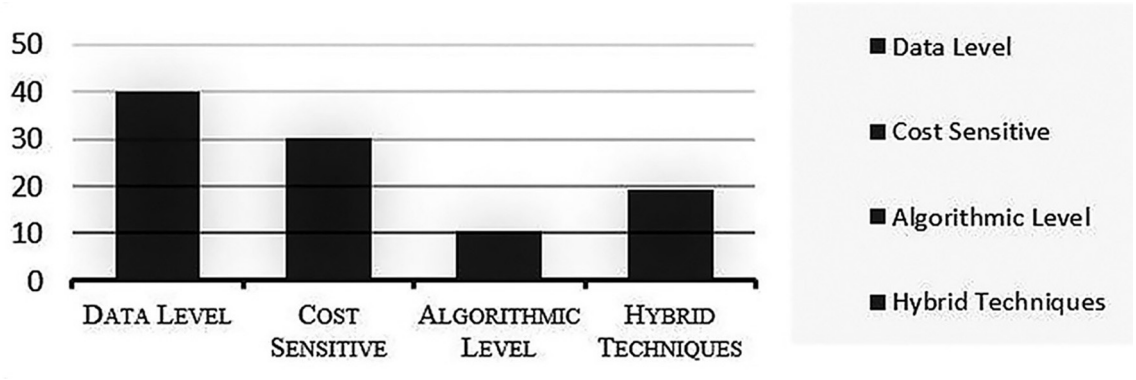


Figure 1. Types of Sampling methods used so far in literatures.

2. Background of the Study

2.1. Literature Review

This section aims to review the works that address imbalanced classification in health insurance fraud detection. First, the section explains the problem of imbalanced classification faced by various domains, and also summarizes the widely used techniques to tackle this problem. Later, we discuss the studies that have addressed imbalanced data learning in the health insurance domain. Lastly, we discuss the gaps that are found in the reviews. The problem of imbalanced classification has been faced in many fields, such as cancer prediction, fraud detection, intrusion detection, detecting oil spills from seabed, *etc.* A detailed study of the issues related to class imbalance has been discussed in the following papers: [5], [6], [9], [10]. These papers explain various approaches for dealing with the class imbalance concerning both binary and multiple class classification. Based on the articles reviewed, we can classify the techniques available to deal with imbalanced classification into four categories: data-level sampling, algorithmic-level, cost-sensitive sampling and hybrid methods (a combination of data- and algorithmic-level). Data-level and cost-sensitive approaches have been used widely in most of the literature because of their performance and simplicity to apply; Figure 1 shows the overall percentage of sampling techniques used in previous studies. Among all the methods, random oversampling (ROS) is the most widely applied technique for balancing the classes. The major drawback of using ROS is time efficiency; since ROS oversamples the minority classes. When considering large datasets, the time of execution of the model will increase because of the increase in the data size, which leads to overfitting the model [11]–[14].

Q. Wang *et al.* [13], in his paper, applied an ensemble method for imbalanced data learning. They used a combination of Borderline SMOTE and SVM, which they named as BEBS model. Borderline SMOTE was proposed to overcome the problem of overfitting while using SMOTE. Imbalanced data learning has also been addressed as a significant issue while detecting fraud in the health insurance domain [9], [10], [15]. X. Jian *et al.* [9] proposed a cost-sensitive learning framework using neural networks. The framework was applied on heterogeneous datasets and was able to produce good accuracy. In their work, R. A. Bauder *et al.* [16] reviewed several data sampling approaches and proposed a combination of random undersampling and random oversampling (RUS+ROS) for the health insurance domain. This hybrid combination reduced the data reduction rate that occurs while performing undersampling.

2.2. Research Gap

As a part of the study, several articles addressing the problem of misclassification were reviewed. Most of the literature related to fraud detection in health insurance either focused on oversampling, undersampling or cost-sensitive learning. Undersampling and oversampling cannot be considered feasible solutions, as the

former leads to the loss of information, and the latter tends to overfit the datasets. Cost-sensitive learning was not proven best comparing to data sampling techniques, because of its sensitivity towards large datasets. Another widely used technique was SMOTE oversampling. Though SMOTE is said to be less prone to outliers, it cannot be considered a practical solution because it introduces many noisy examples in the data during oversampling. So, our focus in the study will be on employing a heuristic approach by using oversampling, eliminating all its disadvantages. Considering the gap found in the literature, we understand that meticulous oversampling followed with data cleaning can help improve the prediction performance of large datasets and overcome the learner's sensitivity towards the minority fraudulent classes. The proposed method will help reduce the misclassification costs generated due to the problem of high dimensionality and skewed class distribution found in insurance databases.

3. Experimental Design and Methodology

Algorithms used for building and evaluating the model are discussed in this section.

3.1 Samplers for Imbalanced Classes

Imbalanced data learning has been a topic of discussion in various research works. We can broadly classify the sampling techniques into three categories [17]: cost-sensitive learning approaches, algorithmic-level approaches, and data-level approaches.

Cost-sensitive learning finds the misclassification error by calculating the cost of error that occurred during prediction while training the model and retraining it until the cost is reduced. In fraud detection, classifying non-fraudulent samples as fraudulent samples can be considered as misclassification. Here, the cost is regarded as a penalty in the wrong prediction, and the ultimate aim is to minimize the total cost [8].

Algorithmic-level approaches of sampling modify the existing classifier to adapt the model to the imbalanced dataset. Some of the popular classifiers used are decision tree classifier, support vector machine, *etc.* [18].

Data-level approaches balance classes' distribution by manipulating the training data. They include resampling the data in two different ways, either undersampling or oversampling. Undersampling balances the classes by removing the instances from majority classes. The major disadvantage of using undersampling is that we tend to lose data of utmost importance [19]. Oversampling balances the class by oversampling or replicating the minority samples. Several oversampling techniques are available across the literature, such as ROS, SMOTE, Adaptive Synthetic Minority Over Sampling Technique (ADASYN), *etc.* ROS randomly picks samples from minority classes and duplicates these instances until there occurs a balance between both the classes. One of the main disadvantages of oversampling is that it increases the minority samples, increasing the size and affecting the computational time. Duplicating the minority classes introduces unnecessary data noise, making the model complex [20], [21].

SMOTE oversampling creates artificial samples from minority classes to provide a balanced dataset. A significant difference between ROS and SMOTE is that ROS duplicates the data, while SMOTE interpolates samples from minority classes using the nearest neighboring (k-NN) technique. For example, a and a^x are two samples from minority classes, a new synthetic sample will be a linear combination from the samples (a, a^x) and is defined as follows:

$$s = a + q \cdot (a^x - a) \tag{1}$$

where q lies within $0 \leq q \leq 1$ and a^x is randomly picked from k-nearest neighbors of a from the minority classes. The nearest neighbor can be defined by the user based on the data distribution [22], [23].

ADASYN oversampling technique is an improved version of SMOTE. It adaptively generates synthetic samples from minority classes based on the data distribution. It concentrates on oversampling the areas adjacent to those minority samples which are incorrectly classified using k-NN classifier [15]. The capability of the algorithm to shift the boundaries will help in reducing the bias while learning [24].

Hybrid methods for imbalanced data sampling use two or more algorithms to better perform by complementing their flaws. A clustering, classification, bagging and boosting algorithms are

modelled along with ROS, RUS, and SMOTE [5]. Many hybrid methods tried using several combinations of algorithms. More explanation of these methodologies is out of the paper's scope and could be read in literature [25], [26].

3.2. Classification Algorithms

Classification algorithms are the popular fraud detection mechanisms experimented with in past literature to detect fraud from insurance claims [27], [28]. Among many of them, *k*-Nearest Neighbors (*k*-NN), Artificial Neural Networks (ANN), Decision Tree Classifier (CART) and Logistic Regression (LR) are the popular learners that are recommended by literature on fraud detection due to their simplicity, performance and efficiency [29] – [32].

k-NN based fraud detection uses a specific distance metric for measuring the distance between two nearest neighbors. A distance rule is formed to find out whether the incoming transaction is fraud or legitimate. An incoming transaction will be classified by measuring the closest point, *i.e.*, if it is a fraud, the new sample will be labelled as fraud [33], [34]. The distance measures are selected based on the type of data. Table 1 shows the equation for calculating the distance (d) between observations a and b [35].

Table 1. Distance function for *k*-NN algorithm.

Distance Function	Equation
Euclidean	$d = \sum_{i=1}^{j} \left(p_i - q_i \right)^2$
Manhattan	$d = \sum_{i=1}^{j} \left\| p_i - q_i \right\|^2$
Minkowski	$d = \left[\sum_{i=1}^{j} \left(\left\| p_i - q_i \right\| \right)^q \right]^{1/q}$
Hamming	$d = \sum_{i=1}^{k} \left\| p_i - q_i \right\|$

ANN is a family of structures that form a part of machine learning and that are built like the human brain, which uses interconnected neurons to make decisions. A neural network is a tree-like structure with input, output and hidden layers. Each neuron connected to an input layer will be assigned with a corresponding weight, and the product of weight and inputs will be passed on to the hidden layer. Later, with the activation function's help, an output neuron will be generated with the summation of input, weight and biases [36], so the output of a perceptron model is based on the total input. If the summed input is a positive number, the neuron fires an output +1, and −1 otherwise. Therefore, prediction results of a classification in ANN is represented by a hyperplane and could be defined by:

$$\sum_{j=1}^{n} w_j x_j + b = 0 \qquad (2)$$

where x_1, x_2, ..., x_n are input vectors and w_1, w_2, ..., w_n are weights, and b is the bias. For a given input x_j with w_j and b, a classification boundary will be either above or below the defined hyperplane [37]. For a binary classification problem, the samples lying above the hyperplane will belong to class 1, and those who lie below will belong to class 2 [38].

A decision tree is a hierarchical structure with numerous branches, typically a root node (top of the tree), an internal node and a leaf node (bottom part). The output of a decision tree is based on If-Then expressions. For example, a transactional dataset contains four features (f1, f2, f3 and f4) and a target (fraud and non-fraud). The resultant metrics will be based on a certain question that will help us reach the target variable. Figure 2 illustrates the whole procedure during the learning [39].

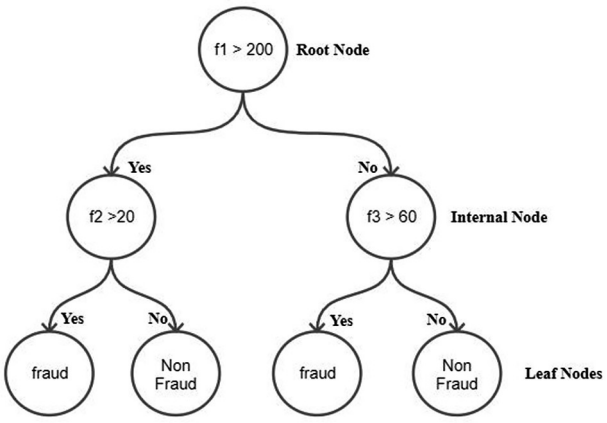


Figure 2. A standard decision tree classifier procedure.

LR is a generalized linear model for predicting binary or multinominal outcomes. In LR, a linear set of variables is produced using a logistic function. The variables take values between 0 and 1. It estimates the probability of a particular instance belonging to one class [40]. Since fraud detection is a classification problem, generalization could be done by specifying the probability of one class. A transaction can be labelled as fraud if the likelihood of a particular instance is more than 50%. The logistic function can be calculated as follows:

$$h_\theta(x) = \sigma((\theta^t x)) \quad (3)$$

where, $h_\theta(x)$ is the hypothesis in classification, $\sigma(z)$ is defined as a real number, and $\sigma(z) = 1/(1 + e^{-zb})$, A logistic sigmoid function squashes the values generated from the linear function to an interval between [0, 1]. In LR, it is $\sigma(z)$ that converts an arbitrary score of x to the probability range between 0 and 1.

3.3. Proposed Method

The details of the working procedure for fraud detection are explained as follows:

3.3.1. Dimensionality Reduction Using PCA

High dimensionality is a major issue while using insurance datasets. Insurance data includes data from various sources such as providers information, patient demographics, drug-related information, patient's medical history and many more. Accumulating all kinds of information from various sources increases the dimensionality of data. High dimensionality reduces the classification accuracy as well as increases the rate of misclassification. Feature reduction has become inevitable before applying any data sampling approaches [41], [42]. PCA provides variance and covariance of features in terms of the new principal components that depict the linear combination of existing variables [43]. The new features can be derived from the eigenvalues of the matrix from the original data. Let a dataset $x = [x_1, x_2, ..., x_n]$ be a matrix with n observations containing m variables. Let r be the covariance matrix of x, $r = [r_1, r_2, ..., r_n]$. If (λ_1, E_1), (λ_2, E_2), ..., (λ_n, E_n) are P eigenvalue-vector pairs of the covariance matrix x, the j-th principal component can be referred to as:

$$y_j = ze_j + ze_{1j} + ze_{2j} + ... + z_n e_{nj}, \; j = 1, 2, ..., n \quad (4)$$

3.3.2. Fused Resampling and Cleaning Ensemble (FusedRCE)

Oversampling using *k*-means-SMOTE. The proposed method uses k-means-SMOTE for oversampling. k-means-SMOTE combines k-means clustering and SMOTE oversampling to balance the classes. The method basically performs in three different steps: clustering, filtering and oversampling [44]. As a first step, k-means clustering is applied for dividing the training set into groups. The algorithm iteratively assigns the observations during the clustering stage and updates the centroid based on the density, converging once all the samples are clustered. As a second step, these groups of clusters are filtered, and the groups with a higher number of minority samples are oversampled. This helps the sampling procedure to restrict itself from generating synthetic samples only in the target area. Although the use of over samplers on an imbalanced dataset will mitigate the problem of skewed class distributions, there are certain issues it possesses. Since over sampler balances the data by interpolating the samples of the minority classes that lie together, its interest will be on increasing the minority samples, and there are chances that the procedure may generalize badly on only minority classes. Noisy data is another problem that persists after oversampling. It increases the misclassification rate, and when the database is highly skewed, this scenario will become even more problematic.

Cleaning noisy data using Tomek Link algorithm. Although the previous step helped us balance the class distributions, as explained, certain problems persist. Since the data we use is highly skewed and large, oversampling causes a drastic increase in size and introduces noise in the data. Training the model with such data could lead to an overfitting problem. An appropriate solution for the problem will be to remove the overlapping data. Defining the classes clearly can reduce the chances of overfitting. There are pros and cons to doing so. However, eliminating samples will reduce the data

Algorithm 1: Overall Pseudocode for the proposed model for fraud detection

Input:

X – number of observations

Y – target class

t – imbalance threshold ratio

Output:

Balanced data

START

1. Standardize the data and calculate covariance for the standardized data
2. Calculate eigen values and eigen vectors for covariance matric and sort the matrix in decreasing order.
3. Multiply the resultant eigen value matrix with the original dataset and produce a linear combination of original features with independent columns.
4. Generate new PCA components with the most relevant features from the combined dataset.
5. Find the imbalanced class ratio, imb_ratio:= (majorityClassCount+1)/(minorityClassCount+1)
6. Form desired clusters using *k*-means clustering
7. For each element in the cluster
   a. if the imbalanced ratio is less than the threshold (t), add the element to a filtered cluster.
8. Find the sampling weight for the filtered cluster based on the density of minority samples present.
9. Oversample the filtered cluster based on SMOTE, return the new balanced set of samples.
10. Apply Tomek Link algorithm to remove noise from the generated set of samples:
    a. For each sample in the new set of generated samples do:
       i. Pick an instance from the class and iterate over.
       ii. Compare the distance of each element with the nearest instance.
       If the distance is lesser or greater than the selected instance, the instance is a noisy one, remove it from the database.
       end
11. Return newly generated samples.

STOP

Figure 3. Pseudocode of the proposed fraud detection method.

size at the cost of data loss. An ideal solution would be to clean the data without losing relevant information. The Tomek Link algorithm is a cleaning technique that clears the samples in the boundary between minority and majority classes, especially the overlapping classes [45]. Its working principle is similar to the nearest neighbors principle. The algorithm chooses two neighboring samples from both classes and considers them as a pair. For example, suppose *a* & *b* are two neighboring instances from different classes. A pair of (*a*, *b*) is a Tomek link only if for a new instance 'c' when calculating the distance (dist), dist(*a*, *b*) should be lesser than dist(*a*, *c*) or dist(*a*, *b*) should be lesser than dist(*b*, *c*). If the said condition satisfies, one of them is considered as noise or overlapping class and will be eliminated from the dataset. The pseudocode of the algorithm is shown in Figure 3, while the workflow of the algorithm is depicted in Figure 4.

3.4. Experimental Setup

The experiment was carried out using an open-source Python based platform. To achieve this, we used: pandas library for preprocessing; Keras library was used to create and train multilayer perceptron model; sklearn library was used to perform standardization, dimensionality reduction and implementation of classification algorithms. Sampling techniques used in this study is imported from the python imblearn library. Python imblearn library, which offers algorithms for resampling, is used for dealing with datasets that show a strong imbalance in the classes.

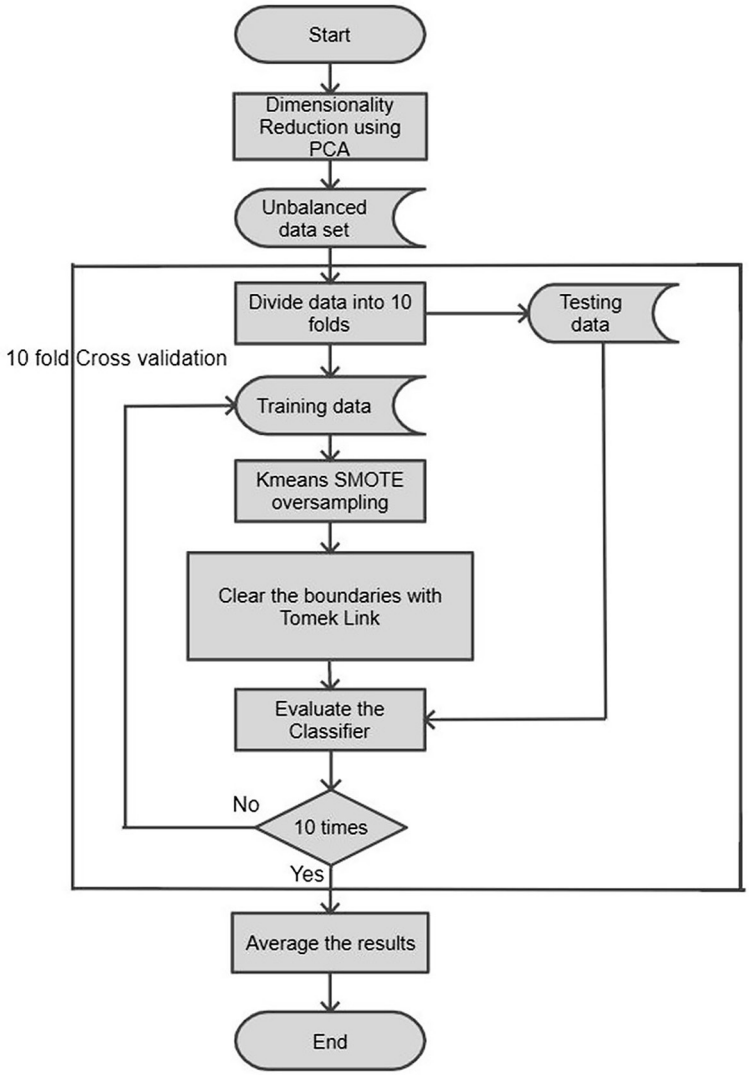


Figure 4. Workflow of the proposed model for fraud detection.

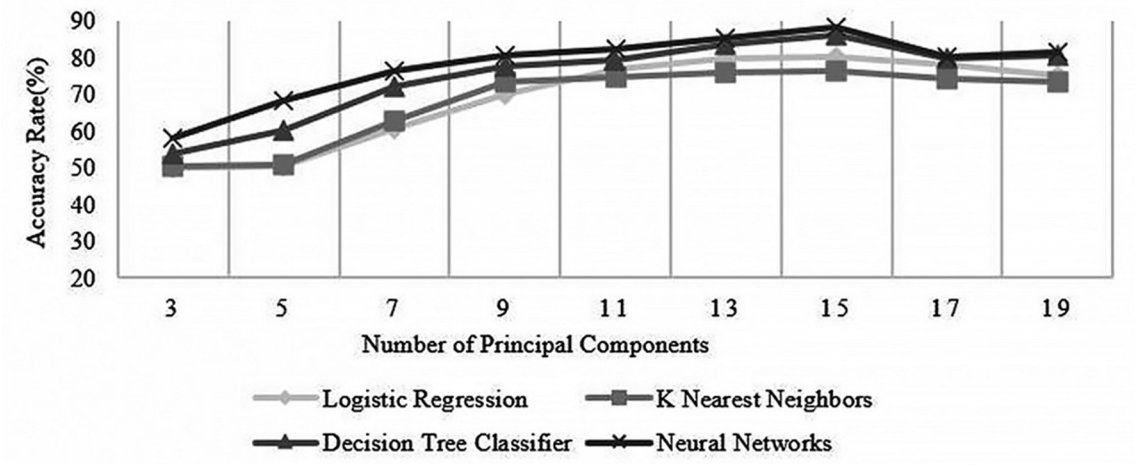


Figure 5. Accuracy Curve for different classifiers with each PC Components.

3.5. Evaluation Metrics

Several metrics were used in literature to assess the performance of a binary classification model, such as Confusion Matrix, ROC curve, Precision, Recall, Accuracy, F1-score, *etc.* [46], [47]. Evaluating results obtained from an imbalanced data set is not similar to a balanced data set. Accuracy could not be taken as the only medium of measurement because overall accuracy tends to be more biased towards majority classes. Precision is calculated as the ratio of accurately predicted positive samples against the total number of positive examples. If a model produces a higher precision and recall rate, we can say that the model had very well handled the classification task. If the recall rate is lower than the precision rate, the model does not classify the samples of a particular class. The major problem arises when the model produces a higher recall and a lower precision, this generates a greater misclassification between the classes. In this study, the positive samples represent a fraudulent sample and the cost of misclassification of a single sample is very high. So the best way to evaluate a fraud detection model under class imbalance is by using the metrics Precision, Recall and F1-score [6], [48]:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{5}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negatives} \tag{6}$$

$$F1-score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{7}$$

4. Results

4.1. Experimental Data Set

We have used two data sets for the study; providers claim data [49] and LEIE data [50] from the CMS Medicare database. Providers claim data (called Part B) provide necessary information related to the number of services a physician has performed, billed, submitted and allowed charges for a particular service, where he has shown his service, *etc.* LEIE data consist of providers who have committed fraud and have been removed from their service based on the crime they have committed [18]. Part B and LEIE databases are related to each other as they share a unique identifier, the NPI (National Provider Identifier). So, to label the primary database, *i.e.*, Part B, we joined both data based on its Unique NPI. Similar work was carried out in many other papers, such as [4], [15], [17]. The records with matching NPI are marked as fraud, and the rest are marked as non-fraud. THE initial LEIE database published on the CMS website contained a lot of missing NPI's. Combining the data with this initial version could match only 465 fraud cases. We checked the LEIE data using other unique features, such as UPIN with NPPES NPI Registry to fill out the NPI's. This procedure helped us in identifying 9862 fraudulent cases.

Since the scope of the study was limited to certain kinds of fraud, *i.e.*, upcoding fraud, we further filtered the data. The final details of the dataset used for experimenting are as follows:

- total number of instances: 573941;
- number of majority data samples (legitimate transactions): 571350;
- number of minority data samples (fraud transactions): 2591;
- original class imbalance ratio (majority:minority): [99:1].

4.2 Performance Analysis of the FusedRCE Method for Fraud Detection

4.2.1 Reducing the Dimensionality by Applying PCA

To find the most relevant features representing the data set, we mapped the dataset with a range of k principal components, where k values range from 3–19. When the classifier was applied to each value of k starting with 3, we could see from Figure 5 that there was a steady growth in the accuracy rate. We can also ob-

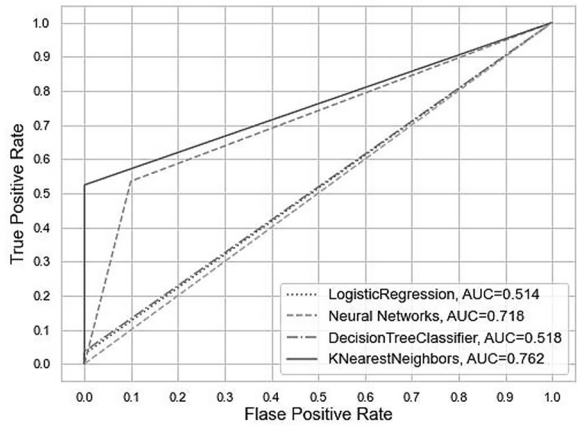
serve that once the value of k reached 15, the growth stagnated and started dropping slowly. Hence, we can state that an ideal number of components could be 15. To understand the change in performance of the learner on applying PCA based on the classification thresholds, we plotted a ROC (Receiver Operating Curve) curve for all the learners. Figure 6(a) shows a huge misclassification of classes, and the prediction probability was around 50% for LR and CART. Neural Networks and k-NN showed a better classification probability comparing to other classifiers. After applying PCA, from Figure 6(b), we can find that the misclassification rates have been decreased. The rate of prediction probability has increased by a minimum of 4% for every learner.

4.2.2. Analyzing the Results Based on Data Distribution

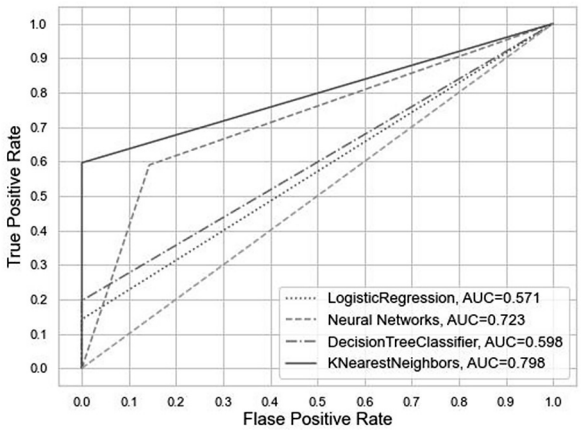
To find the change in the distribution of samples at each stage of sampling, we plotted the instances of each class using a scatter plot. Due to the difficulty in representing the entire data on a single plot because of its larger size, we randomly sampled 1000 observations from the dataset and plotted the distribution. The results are shown in Figure 7; lighter grey dots represent fraudulent transactions, *i.e.*, minority samples, and darker grey dots represent majority samples *i.e.*, legitimate transactions. The original distribution of the dataset is shown in Figure 7(a); from the figure, we can see only very few minority samples compared to the majority samples. From Figure 7(b), we can find that the distribution seems to be balanced after the application of SMOTE. However, we can also notice from the figure that the samples that are oversampled are concentrated at the corner. Also, there is a massive overlapping between minority and majority samples. Figure 7(c) represents the distribution of samples after applying the proposed k-means-SMOTE method, we can see that the distribution is spread across the entire area but overlapping still persists. Figure 7(d) explains the distribution of data after the application of Tomek Link as the cleaning method. It eliminated the noisy samples from the overlapped area and made the distribution precise and distinct to a particular extent. Figure 8 represents the performance comparison of FusedRCE with other samplers.

4.2.3. Tuning the Appropriate Parameters for the Proposed Algorithm

The performance of a sampling procedure is always dependent on its parameters to an extent. The performance of the proposed method depends on certain parameters, such as the cluster size (k), the value of nearest neighbor (nn), threshold ratio (irt) and sampling ratio. We executed the learners with repeated cross-valida-

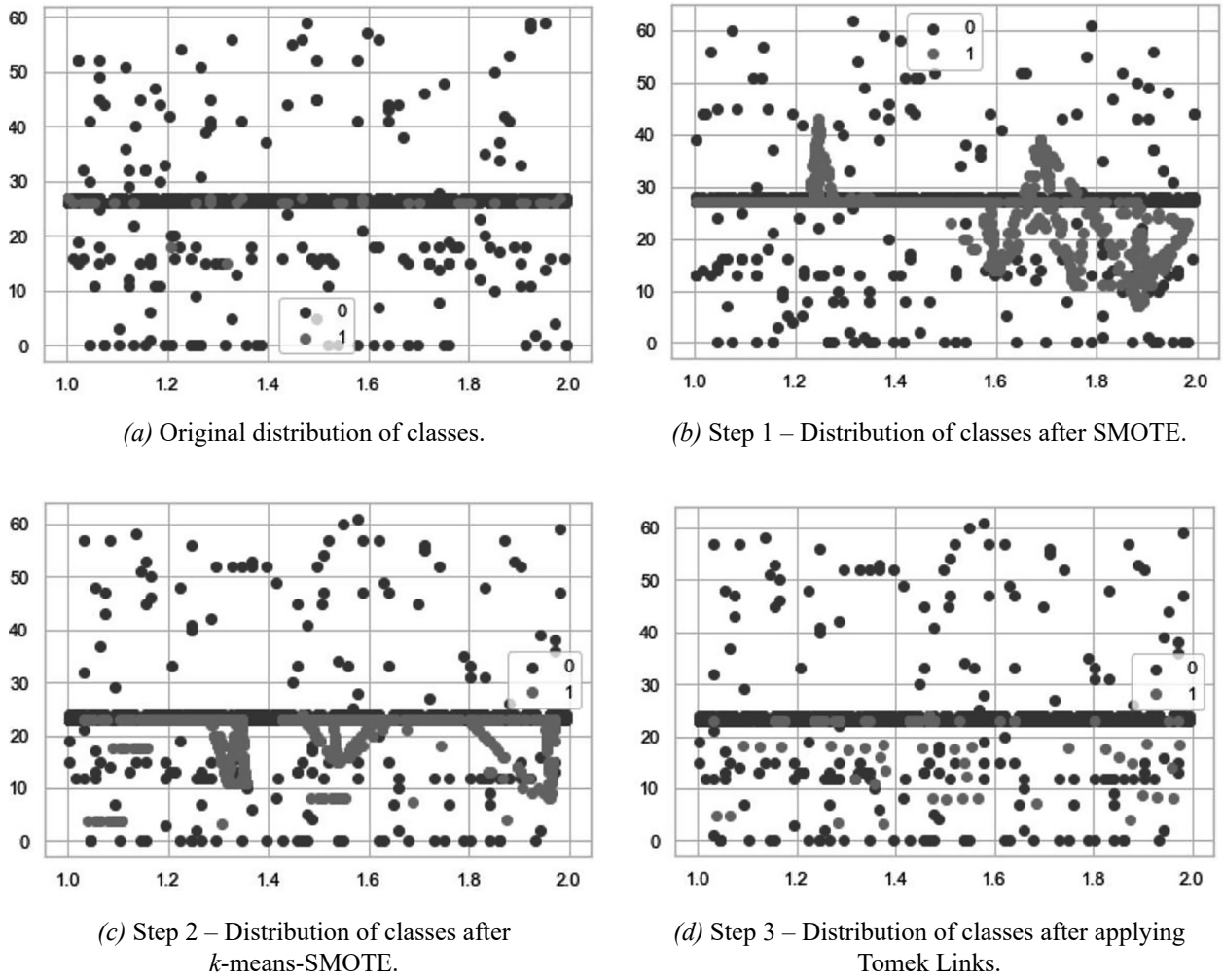


(a) Performance of learners before dimensionality reduction.



(b) Performance of learners after dimensionality reduction.

Figure 6. Performance of each learner before and after dimensionality reduction at the classification thresholds.



(a) Original distribution of classes.

(b) Step 1 – Distribution of classes after SMOTE.

(c) Step 2 – Distribution of classes after *k*-means-SMOTE.

(d) Step 3 – Distribution of classes after applying Tomek Links.

Figure 7. Change in the distribution of data after each phase of sampling.

tion to determine the most appropriate combination of parameters for FusedRCE, using the set of *k* values with {10, 20, 30}, *nn* with values {3, 7, 20}, *irt* with {1, 0.1, 0.01} and sampling ratios with values {(90:10), (50:50), (25:75)}. The appropriate values were observed after running the procedure for *k*, *nn* and *irt* equal to 20, 3 and 0.1, respectively. The sampling ratio was an important parameter that determined the performance of the algorithm. The default sampling rate of every sampling algorithm was 1:1, *i.e.*, increasing the minority samples to the size of the majority. To find the appropriate ratio for sampling, we evaluated each algorithm using three sampling ratios. The balanced dataset obtained after sampling was applied on four different classifiers explained in section 3.2 with the parameters listed in Table 2. The highest score obtained from FusedRCE was with the sampling ratio of (90:10) with an F1-score of LR at 97.5%, CART at 98.3%, *k*-NN at 98.1%, and Neural Networks at 98.9%. To justify the superiority of the proposed FusedRCE over other sampling procedures, we compared its performance with the tuned parameters over the other samplers. The results of the comparison are plotted as a bar graph and are shown in Figure 8.

4.2.4. Analyzing Results of Individual Learners Using FusedRCE

The section analyzes and shows the proposed FusedRCE performance analysis on different learners for fraud detection. We applied and evaluated the proposed method with all the

learners mentioned above to find the best combination for detecting fraudulent samples. The parameters used for building the classifier are detailed in Table 2. Cross-validation was used with five splits and three repeats during evaluation to avoid the chances of overfitting. The experimental results of the change in performance based on data distribution from each sampling technique is shown in Table 3. The table indicates that by cleaning the data after *k*-means-SMOTE oversampling, there is no degradation in the performance, and the performance has even increased for a few learners.

In Figure 9, we plotted a box-whisker plot to summarize the obtained scores from repeated k-fold cross-validation after each repetition. The triangle between the lines indicates the mean of the distribution. If the line and triangle coincide, it shows that the average mean of the scores has captured the center tendency well. Figure 9(a) and 9(b) compare the results obtained on each classifier after applying the proposed sampling procedure. We can see that all classifiers were performing well on each fold after using the proposed framework. The boxplot shows that there was no variation in the results obtained after each fold. Except in LR, we couldn't find any outliers on other classifiers. Considering the prediction results, ANN showed better results with the lowest F1-score of 98.8% and the highest F1-score of 99%, leading to a mean F1-score of 98.9%.

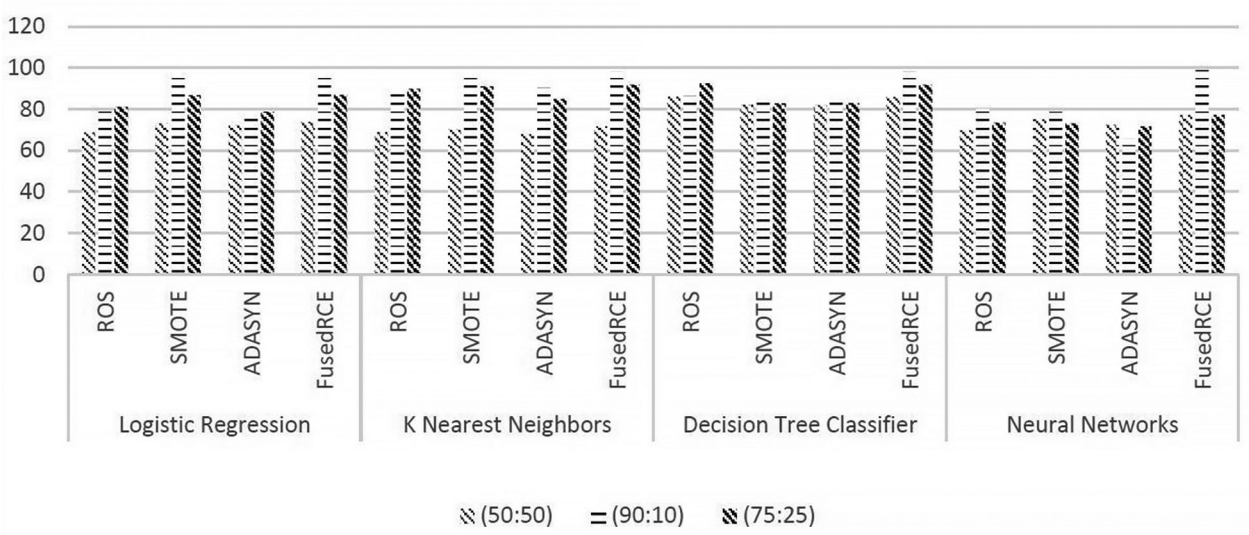


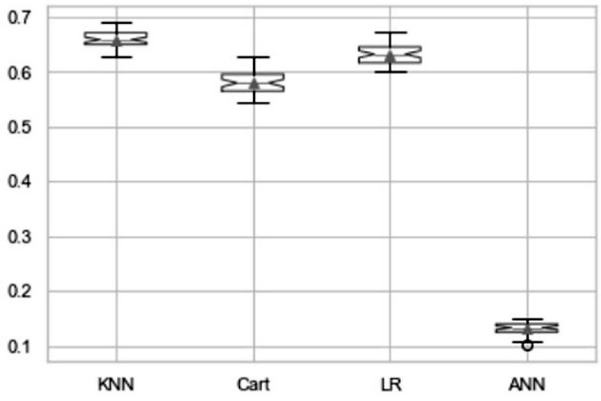
Figure 8. Performance comparison of different sampling algorithms with sampling ratios.

Table 2. Parameter list for each classifier applied during the experiment.

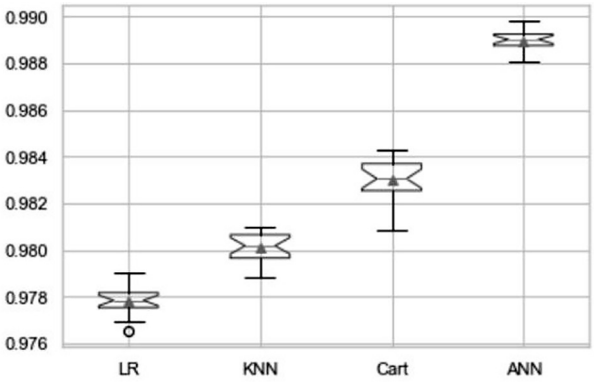
Acronym	Classifier	Parameters
LR	Logistic regression	Penalty: L2 (Ridge Regression), Solver: lbfgs Iterations: 100, Inverse Reg Parameter: 1.0
k-NN	*k*-Nearest Neighbors	number of Nearest Neighbors = 5, distance = minkowski, power parameter for minkowski(p) = 2, n_jobs = 1
DTC	Decision Tree Classifier	criterion of split = gini, splitter = best, max_dept = 8 minimum leaf = 1, sample split = 2
ANN	(Artificial Neural Network) Multilayer Perceptron	Learning_rate=0.1, No of epochs = 34 Momentum = 0.6, Batch_size = 256 No. of hidden nodes = 3, Optimizer = adam

Table 3. Performance evaluation of sampling algorithms using classification algorithms.

Learners	Data Sampling Algorithms	Precision		Recall		Accuracy	F1-score	AUC
		Class 1	Class 0	Class 1	Class 0			
Logistic Regression	SMOTE	18	96	40	88	85	25	64
	kmeans-SMOTE	02	0	100	04	02	04	50
	FusedRCE	12	97	66	69	68	20	67
k-Nearest Neighbors	SMOTE	63	98	66	97	96	64	81
	kmeans-SMOTE	91	100	93	100	99	92	96
	FusedRCE	96	99	93	100	98	93	95
Decision Tree Classifier	SMOTE	61	98	71	97	95	65	84
	kmeans-SMOTE	87	99	86	100	98	86	92
	FusedRCE	88	99	87	99	98	87	93
Artificial Neural Networks	SMOTE	83	99	83	99	98	82	90
	kmeans-SMOTE	95	98	75	99	96	83	87
	FusedRCE	96	99	78	100	99	86	88



(a) F1-score of classifiers before sampling and dimensionality reduction.



(b) F1-score after applying FusedRCE and dimensionality reduction.

Figure 9. F1-score of 5-fold cross-validation for each learning algorithm with and without the application of the proposed algorithm.

4.2.5. Discussion

We tested the effectiveness of the overall framework with the Medicare CMS Part B database. Detecting a maximum number of fraudulent insurance cases is of utmost interest in the study. We were also very keen on keeping the whole procedure cost-effective by keeping the most relevant features and the appropriate sampling procedure. We can also justify the application of the proposed framework for the following reasons:

- the use of undersampling approaches alone leads to information loss. Since we had only 2591 fraud cases out of 573941 instances, reducing 571350 (legitimate transactions) to 2591 case will result in significant data loss;

- if we go only for oversampling the minority classes, we will end up creating an overfitted model. There will be considerable growth in the data due to oversampling, increasing the computational cost [23]. The size of our dataset is enormous, so an additional increase of classes with random sampling or synthetic sampling will resulting in a massive increase in volume.

Cost-effectiveness could be achieved by keeping the false alarm rate lower, which is the precision rate in the classification algorithm. There is always a negotiation between recall and precision. Oversampling will help us in increasing the recall but at the cost of precision. False alarms might lead to a loss of faith among customers, and indirectly, they might be attracted to competitors. Oversampling could help us in improving the recall but will end up giving poor precision and accuracy. The application of the proposed FusedRCE on classifiers helped us in maintaining a good recall. Also, we could increase the precision at an acceptable level (>90%) on all the classifiers with a sampling ratio of [0.90, 0.10]. From the results, we could also see that the model had produced a good separability between classes, showing an AUC value of 95% using *k*-NN, 93% with CART and 88% with ANN. We can objectively state that the proposed FusedRCE through concentrated oversampling and noise removal had improved prediction performance with the above results. Also, it helped in reducing the overlapping and misclassifications caused during sampling.

5. Conclusion

The work analyzes different oversampling techniques to deal with the problem of imbalanced data learning. When classifying large datasets with high-class imbalance, certain issues persist while using oversampling algorithms. Oversampling increases the data size and introduces noise in the data by imputing a similar pattern of minority samples everywhere, further generalizing the model. To solve the problem, we propose a Fused Resampling and Cleaning Ensemble (FusedRCE) that uses kmeans-SMOTE to oversample the data and filter noise through data cleaning using the Tomek Link algorithm. The experiments prove that the proposed sampling method helped us overcome the disadvantages caused by balancing the data. We have also applied an appropriate feature reduction technique to select the most relevant features required for the study. The entire framework was executed with repeated 5-fold cross-validation for avoiding any chances of overfitting. The model was applied to many classifiers to find a suitable learner based on its performance. Out of the four classification algorithms used, the FusedRCE algorithm outperformed other sampling methods on three classifiers. Based on the above results, we can ascertain that the combined approach showed superior performance compared to other sampling approaches mentioned. Among classifiers, ANN with FusedRCE proved to have the highest F1-score of 98.9%, with a shorter delay in execution time.

References

[1] M. E. Johnson and N. Nagarur, "Multi-Stage Methodology to Detect Health Insurance Claim Fraud", *Health Care Management Science*, vol. 19, no. 3, pp. 249–260, 2016. https://doi.org/10.1007/s10729-015-9317-3

[2] S. Wang, "A Comprehensive Survey of Data Mining-Based Accounting-Fraud Detection Research", in *Proc. of the 2010 International Conference on Intelligent Computation Technology and Automation*, 2010. http://dx.doi.org/10.1109/icicta.2010.831

[3] S. K. Shamitha and V. Ilango, "A Survey on Machine Learning Techniques for Fraud Detection in Healthcare", vol. 7, no. 4, pp. 5862–5868, 2018. http://dx.doi.org/10.14419/ijet.v7i4.15696

[4] S. Kareem *et al.*, "Framework for the Identification of Fraudulent Health Insurance Claims Using Association Rule Mining", in *Proc. of the IEEE Conference on Big Data and Analytics (ICBDA)*, 2017, pp. 99–104. http://dx.doi.org/10.1109/ICBDAA.2017.8284114

[5] A. Ali *et al.*, "Classification with Class Imbalance Problem: A Review", *International Journal of Advances in Soft Computing and its Applications*, vol. 7, no. 3, pp. 176–204, 2015.

[6] P. Kaur and A. Gosain, "Issues and Challenges of Class Imbalance Problem in Classification", *International Journal of Information Technology*, 2018. https://doi.org/10.1007/s41870-018-0251-8

[7] G. G. Sundarkumar and V. Ravi, "A Novel Hybrid Undersampling Method for Mining Unbalanced Datasets in Banking and Insurance", *Engineering Applications of Artificial Intelligence*, vol. 37, pp. 368–377, 2015. https://doi.org/10.1016/j.engappai.2014.09.019

[8] X. Jiang *et al.*, "Cost-Sensitive Parallel Learning Framework for Insurance Intelligence Operation", *IEEE Transactions on Industrial Electronics*, vol. PP, no. c, p. 1, 2018. http://dx.doi.org/10.1109/TIE.2018.2873526

[9] S. T. Jishan *et al.*, "Improving Accuracy of Students' Final Grade Prediction Model Using Optimal Equal Width Binning and Synthetic Minority

Over-Sampling Technique", *Decision Analytics*, vol. 2, no. 1, 2015. https://doi.org/10.1186/s40165-014-0010-2

[10] E. M. Hassib *et al.*, "An Imbalanced Big Data Mining Framework for Improving Optimization Algorithms Performance", *IEEE Access*, vol. 7, pp. 170774–170795, 2019. http://dx.doi.org/10.1109/access.2019.2955983

[11] C. Phua *et al.*, "Minority Report in Fraud Detection", *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 50–59, 2004. http://dx.doi.org/10.1145/1007730.1007738

[12] E. Kurniawan *et al.*, "C5.0 Algorithm and Synthetic Minority Oversampling Technique (SMOTE) for Rainfall Forecasting in Bandung Regency", in *Proc. of the 7th International Conference on Information and Communication Technology (ICoICT)*, 2019. http://dx.doi.org/10.1109/icoict.2019.8835324

[13] Q. Wang *et al.*, "A Novel Ensemble Method for Imbalanced Data Learning: Bagging of Extrapolation-SMOTE SVM", *Computational Intelligence and Neuroscience*, pp. 1–11, 2017. http://dx.doi.org/10.1155/2017/1827016

[14] R. A. Bauder and T. M. Khoshgoftaar, "Medicare Fraud Detection Using Machine Learning Methods", in *Proc. of the 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2017. http://dx.doi.org/10.1109/icmla.2017.00-48

[15] R. Bauder and T. Khoshgoftaar, "Medicare Fraud Detection Using Random Forest with Class Imbalanced Big Data", in *Proc. of the IEEE International Conference on Information Reuse and Integration (IRI)*, 2018. http://dx.doi.org/10.1109/iri.2018.00019

[16] R. A. Bauder *et al.*, "Data Sampling Approaches with Severely Imbalanced Big Data for Medicare Fraud Detection", in *Proc. of the IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, 2018. http://dx.doi.org/10.1109/ictai.2018.00030

[17] Y. Sun *et al.*, "Classification of Imbalanced Data: A review", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 4, pp. 687–719, 2009. http://dx.doi.org/10.1142/S0218001409007326

[18] A. Mahani and A. R. Ali, "Classification Problem in Imbalanced Datasets", *Recent Trends in Computational Intelligence*, 2020. http://dx.doi.org/10.5772/intechopen.89603

[19] S. Kotsiantis *et al.*, "Handling Imbalanced Datasets: A Review", 2006.

[20] M. S. Santos *et al.*, "Cross-Validation for Imbalanced Datasets: Avoiding Overoptimistic and Overfitting Approaches [Research Frontier]", *IEEE Computational Intelligence Magazine*, vol. 13, no. 4, pp. 59–76, 2018. http://dx.doi.org/10.1109/mci.2018.2866730

[21] F. Hu and H. Li, "A Novel Boundary Oversampling Algorithm Based on Neighborhood Rough Set Model: NRSBoundary-SMOTE", *Mathematical Problems in Engineering*, pp. 1–10, 2013. http://dx.doi.org/10.1155/2013/694809

[22] R. Blagus and L. Lusa, "Evaluation of SMOTE for High-Dimensional Class-Imbalanced Microarray Data", in *Proc. of the 11th International Conference on Machine Learning and Applications*, 2012. http://dx.doi.org/10.1109/icmla.2012.183

[23] N. V. Chawla *et al.*, "SMOTE: Synthetic Minority Over-sampling Technique", *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002. http://dx.doi.org/10.1613/jair.953

[24] H. He *et al.*, "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning", in *Proc. of the IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008. http://dx.doi.org/10.1109/ijcANN.2008.4633969

[25] R. C. P. Gustavo and A. P. A. Batista, "A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data," Soziale Systeme, vol. 8, no. 2, pp. 20–29, 2016. http://dx.doi.org/10.1515/sosys-2002-0206

[26] E. A. Gustavo *et al.*, "A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data", *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, 2004. http://dx.doi.org/10.1145/1007730.1007735

[27] H. Zhou *et al.*, "A Scalable Approach for Fraud Detection in Online E-Commerce Transactions with Big Data Analytics", *Computers, Materials & Continua*, vol. 60, no. 1, pp. 179–192, 2019. http://dx.doi.org/10.32604/cmc.2019.05214

[28] V. Rawte and G. Anuradha, "Fraud Detection in Health Insurance Using Data Mining Techniques", in *Proc. of the International Conference on Communication, Information & Computing Technology (ICCICT)*, 2015. http://dx.doi.org/10.1109/iccict.2015.7045689

[29] N. Malini and M. Pushpa, "Analysis on Credit Card Fraud Identification Techniques Based on K-NNand Outlier Detection", in *Proc. of the 3rd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, 2017. http://dx.doi.org/10.1109/aeeicb.2017.7972424

[30] A. Husejinović, "Credit Card Fraud Detection Using Naive Bayesian and c4.5 Decision Tree Classifiers", *Periodicals of Engineering and Natural Sciences*, vol. 8, no. 1, pp. 1–5, 2020. http://dx.doi.org/10.21533/pen.v

[31] M. S. Kumar *et al.*, "Credit Card Fraud Detection Using Random Forest Algorithm", in *Proc. of the 3rd International Conference on Computing and Communications Technologies (ICCCT)*, 2019. http://dx.doi.org/10.1109/iccct2.2019.8824930

[32] M. M. Badža and M. Č. Barjaktarović, "Classification of Brain Tumors from MRI Images Using a Convolutional Neural Network", *Applied Sciences*, vol. 10, no. 6, 1999. http://dx.doi.org/10.3390/app10061999

[33] M. Khodabakhshi, "Archive of SID Fraud Detection in Banking Using K-NN (K-Nearest Neighbor) Algorithm Archive of SID", *International Conference on Research in Science and Technology*, pp. 26–34, 2016.

[34] C. Sudha and T. N. Raj, "Credit Card Fraud Detection in Internet Using K-Nearest Neighbor Algorithm," vol. 5, no. 11, pp. 22–30, 2017.

[35] L. Hu *et al.*, "The Distance Function Effect on k-Nearest Neighbor Classification for Medical Datasets", *SpringerPlus*, vol. 5, no. 1, 2016. http://dx.doi.org/10.1186/s40064-016-2941-7

[36] R. B. Asha. and S. K. S. Kumar, "Credit Card Fraud Detection Using Artificial Neural Network", *Global Transitions Proceedings*, vol. 2, no. 1, pp. 35–41, 2021. http://dx.doi.org/10.1016/j.gltp.2021.01.006

[37] Department of Mathematics Uppsala University. (2017, October). "Deep Neural Networks and Fraud Detection" (No. Lu2017DeepANN). UPPSALA UNIVERSITET.

[38] A. Gulati *et al.*, "Credit Card Fraud Detection Using Neural Network and Geolocation", *IOP Conference Series: Materials Science and Engineering*, vol. 263, 2017. http://dx.doi.org/10.1088/1757899x/263/4/042039

[39] Y. Y. Song and Y. Lu, "Decision Tree Methods: Applications for Classification and Prediction", *Shanghai Archives of Psychiatry*, vol. 27, no. 2, pp. 130–135, 2015. http://dx.doi.org/10.11919/j.issn.1002-0829.215044

[40] S. Makki *et al.*, "An Experimental Study With Imbalanced Classification Approaches for Credit Card Fraud Detection", *IEEE Access*, vol. 7, pp. 93010–93022, 2019. http://dx.doi.org/10.1109/access.2019.2927266

[41] S. Karimi *et al.*, "Application of Structural Equation Modelling to Assess the Effect of Entrepreneurial Characteristics on Students' Entrepreneurial Intentions", *SSRN Electronic Journal*, 2012. http://dx.doi.org/10.2139/ssrn.2152932

[42] D. Feldman *et al.*, "Turning Big Data Into Tiny Data: Constant-Size Coresets for *k*-Means, PCA, and Projective Clustering", *SIAM Journal on Computing*, vol. 49, no. 3, pp. 601–657, 2020. http://dx.doi.org/10.1137/18m1209854

[43] A. Krishnaraj and P. C. Deka, "Spatial and Temporal Variations in River Water Quality of the Middle Ganga Basin Using Unsupervised Machine Learning Techniques", *Environmental Monitoring and Assessment*, vol. 192, no. 12, 2020. http://dx.doi.org/10.1007/s10661-020-08624-4

[44] G. Douzas *et al.*, "Improving Imbalanced Learning Through a Heuristic Oversampling Method Based on *k*-Means and SMOTE", *Information Sciences*, vol. 465, pp. 1–20, 2018. http://dx.doi.org/10.1016/j.ins.2018.06.056

[45] I. Tomek, "Two Modifications of CNN", *IEEE Transactions on Systems, Man, and Cybernetics, SMC* , vol. 6, no. 11, pp. 769–772, 1976. http://dx.doi.org/10.1109/tsmc.1976.4309452

[46] M. Hossin and M. N. Sulaiman, "A Review on Evaluation Metrics for Data Classification Evaluations", *International Journal of Data Mining & Knowledge Management Process*, vol. 5, no. 2, pp. 1–11, 2015. http://dx.doi.org/10.5121/ijdkp.2015.5201

[47] Y. Liu *et al.*, "A Strategy on Selecting Performance Metrics for Classifier Evaluation", *International Journal of Mobile Computing and Multimedia Communications*, vol. 6, no. 4, pp. 20–35, 2014. http://dx.doi.org/10.4018/ijmcmc.2014100102

[48] M. Bekkar *et al.*, "Evaluation Measures for Models Assessment over Imbalanced Data Sets", *Journal of Information Engineering and Applications*, vol. 3, no. 10, pp. 27–38, 2013. [Online]. Available: http://www.iiste.org/Journals/index.php/JIEA/article/view/7633

[49] "Part B National Summary Data File (Previously known as BESS)," Data base _ Medicare C., p. 21244, 2018. [Online]. Available: https://www.cms.gov/Research-Statistics-Data-and-Systems/Downloadable-Public-Use-Files/Part-B-National-Summary-Data-File

[50] Office of Inspector General, "Exclusions – Office of Inspector General," Exclusions database U.S. Department of Health and Human Services. 2018, [Online]. Available: https://oig.hhs.gov/exclusions/background.asp

Received: March 2021
Revised: August 2021
Accepted: August 2021

Contact addresses:

Shamitha S. Kotekani
Department of Computer Applications
CMRIT
Bengaluru
India
VTU
Belgavi
India
e-mail: shamithashibu@gmail.com

Ilango Velchamy
Department of Computer Applications
CMRIT
Bengaluru
India
VTU
Belgavi
India
e-mail: thirukkural69@gmail.com

Shamitha S. Kotekani is currently pursuing her PhD in computer applications at Visveswaraya Technological University (Karnataka, India). She has obtained her MCA degree in computer applications from Kannur University in 2008. Her current research interests include developing an automated fraud detection system, machine learning for big data, and artificial intelligence.

Ilango Velchamy is a Professor and Head of the Centre of Excellence for Intelligent Human Computer Interaction, Head-Research Centre, at CMR Institute of Technology, Bengaluru, India. He has published 16 Patents as an Inventor in the area of Information and Communication Technology. He is the author of over 90 scholarly research papers, including 40+ reputed journal papers (Springer, IEEE, WOS, and SCI). His current research interests include healthcare, data science and analytics, cognitive engineering, personalization, user experience, and medical image and video data analysis.