Computers, Materials & Continua
DOI: 10.32604/cmc.2022.025960
Article

Tech Science Press

MCBC-SMOTE: A Majority Clustering Model for Classification of Imbalanced Data

Jyoti Arora[1], Meena Tushir[2], Keshav Sharma[1], Lalit Mohan[1], Aman Singh[3,*], Abdullah Alharbi[4] and Wael Alosaimi[4]

[1]Department of Information Technology, MSIT, GGSIPU, New Delhi, 110058, India
[2]Department of Electrical and Electronic Engineering, MSIT, GGSIPU, New Delhi, 110058, India
[3]School of Computer Science and Engineering, Lovely Professional University, 144411, Punjab, India
[4]Department of Information Technology, College of Computers and Information Technology, Taif University, 11099, Taif 21944, Saudi Arabia
*Corresponding Author: Aman Singh. Email: amansingh.x@gmail.com

Received: 10 December 2021; Accepted: 02 March 2022

Abstract: Datasets with the imbalanced class distribution are difficult to handle with the standard classification algorithms. In supervised learning, dealing with the problem of class imbalance is still considered to be a challenging research problem. Various machine learning techniques are designed to operate on balanced datasets; therefore, the state of the art, different under-sampling, over-sampling and hybrid strategies have been proposed to deal with the problem of imbalanced datasets, but highly skewed datasets still pose the problem of generalization and noise generation during resampling. To overcome these problems, this paper proposes a majority clustering model for classification of imbalanced datasets known as MCBC-SMOTE (Majority Clustering for balanced Classification-SMOTE). The model provides a method to convert the problem of binary classification into a multi-class problem. In the proposed algorithm, the number of clusters for the majority class is calculated using the elbow method and the minority class is over-sampled as an average of clustered majority classes to generate a symmetrical class distribution. The proposed technique is cost-effective, reduces the problem of noise generation and successfully disables the imbalances present in between and within classes. The results of the evaluations on diverse real datasets proved to provide better classification results as compared to state of the art existing methodologies based on several performance metrics.

Keywords: Imbalance class problem; classification; SMOTE; k-means; clustering; sampling

1 Introduction

In classification, the problem of class imbalance occurs in the scenario where the count of instances in the minority class is smaller than the ones from the majority class. The widely used supervised classification procedures are inclined towards the majority class samples since the learning rules for predicting the values are biased towards classes with more instances and classes with the minority samples are ignored because of unsymmetrical distribution. Sometimes, the decision is required based on the classes with the lowest number of instances. [1]. There are many different applications, such as fraud detection [2], remote sensing [3], diagnosis of medical tumors and finding unreliable telecommunication customers [4–6] where we need to deal with such issues. In these applications, widely used classifiers have a bias towards the classes with the greater number of instances. Standard classification algorithms are designed with the rules that are biased towards predicting the classes where the instances are positively weighted in favor of the accuracy metric; however the classes with minority samples are ignored (considering noise). In such cases, minority classes (desired class) are often ignored and are misclassified. Further classes with the skewed distribution of the data suffer from the issues of class overlapping and a small sample size causes difficulty in the learning of the classifier [7–9].

To overcome with the class imbalance problem, enormous numbers of algorithms have emerged in the literature. These techniques are based on algorithmic level approaches, data level handling approaches and cost-sensitive approaches. The algorithmic level approaches include modification of the classification algorithm by training the classifier as per the class [10,11]. In data level approaches, pre-processing of the data is done to reduce the effect of skewed class distribution, to enhance the learning process for classification. These approaches also include various integrated approaches that involve the combination of under-sampling and oversampling [12–14]. In cost-sensitive learning, the data-level and algorithmic level approaches are combined incorporating different misclassification costs and classifying the samples by optimizing the costs, referencing the minority class samples. The data level approaches involve resampling of the data using strategies such as under-sampling and oversampling to improve the accuracy of classification by modifying the original training sample set resulting into more balanced training set, which is used to train the model and improve the accuracy of the classification. In under-sampling, resampling of the data is done by removing the instances from the majority classes while in oversampling the instances are added to the minority classes.

The simplest over-sampling method, Random over-sampling, randomly creates the new samples of the data by replicating the original data [13–15]. In the imbalanced data with the multi-class problem, the samples of minority class might get subjugated with several majority classes. This may result in diversifying the minority class instances with other classes, which may cause the problem of over fitting. To overcome the problem of random oversampling, the emphasis was given on the expansion of the sample of the instances from the minority classes only. This problem can be overcome by adding new samples, interpolated from the existing samples. The concept was given as Synthetic Minority Over-Sampling Techniques (SMOTE) [10] proposed on the basis of expanding the regions of minority class by introducing the new instances from the feature space of the original instances of the minority class. There are different methods introduced to generate these samples of the minority class. In k-NN based SMOTE [16], the samples were generated in the direction of k-nearest minority neighbors. Generating synthetic instances on the basis of neighbor instances may result in the problem of over-generalization. In this method, samples of the minority class may be allocated to the region of majority class owing to the presence of noise or any other disturbances [17]. These synthetic instances can contain a lot of noise and generate samples which do not support the learning process as the instances produced are not the replication of original minority class instances.

The methods based on synthetic sampling normally have a high risk of introducing noise when a class with the minority samples has limited data and get overlapped with majority class. Furthermore, with the small number of minority observations as compared to majority class, it results into the generation of large number of oversampled data for a balanced distribution yielding oversized dataset. The efficiency of the learning algorithm decreases with the increase in the complexity of the large over-sampled training dataset. For some noisy datasets, it will result into the problem of over generalization. In order to overcome the above-mentioned problems, the motivation of this paper is to limit the size of the oversampled data of the minority class by clustering the majority class into the number of clusters estimated by the elbow method.

The key contributions of the proposed work are as follows:

- We proposed an imbalance classification technique using the concept of majority clustering with the SMOTE.
- Developed an effective way to deal with skewness present in the data by mapping a problem of binary class into a multi-class problem that result into symmetrical classification for imbalanced data.
- The majority class is partitioned to generate the size of the over-sampled minority class, thereby improving the robustness of the method to noise.
- Reduces the overgeneralization of resampled dataset, by restricting the size of the oversampled minority class. It helps to reduce the complexity of the algorithm.

The proposed technique involves various steps. The first step is cleaning of the data using pre-processing techniques [18,19]. The features are selected using correlation and *p*-value test. This helps to improve the recognition rate of the instances (minority or majority) by certain criteria to mitigate imbalances. The MCBC splits the majority class into a specific number of clusters, where the elbow method is used to select the number of clusters. This step also converts the binary classification problem into a multi-class problem. The clusters are generated using k-means clustering. Further, the minority class is over-sampled using the process of SMOTE, where the size of the sample depends on the average size of the clusters of the majority class.

The remaining paper is structured in the following sections: Section 2 provides the background of the imbalance classification problem. Section 3 defines the proposed approach in detail. Section 4 provides the experimental framework with the results and the discussions. Section 5 concludes the paper followed by future scope.

2 Background

In a problem of binary classification, data samples belong to two classes; the issue of class imbalance arises when the class with the minority samples is the target class as compared to the class with majority sample. In numerous issues [20,21], the class of interest is the minority group, i.e., the positive class. As an example, class imbalance occurs frequently in the area of banking, i.e., credit card fraud detection. Significantly, credit card authorities should be able to discover fraudulent credit card transactions to prevent the customers from paying for the items which they have not purchased. However, conventional machine learning algorithms will automatically over-classify the majority classes because of increased prior likelihood. Subsequently, the occurrences being a part of the minority group are not classified accurately as compared to instances belonging to the majority group. Different resampling methods in the literature have been proved to provide a solution to the class imbalance problem. It includes under-sampling, over-sampling and hybrid methods of resampling (includes combination of under-sampling and over-sampling). The importance of resampling of data to overcome the problem of class imbalance has been discussed in the literature [22]. Among them, over-sampling technique SMOTE and its extensions has been widely used and proved to give better results as compared to under-sampling methods.

2.1 Synthetic Minority Over-Sampling Technique (SMOTE) and Extensions

Chawla et al. [10] in 2002 proposed a new algorithm using data resampling to overcome the problem of over-fitting faced by random over-sampling. This method enlarges the region of the minority class by generating the synthetic instances by the process of interpolation in between the present minority samples and their adjoining minority neighbours. Although SMOTE decreases the risk of over fitting but it has some drawbacks while handling imbalance and noise. The SMOTE randomly selects the minority class sample to interpolate new sample with an even probability. This allows the process to successively raise an issue in the problems related with class imbalance, disputes of within class imbalance and small disjuncts are unnoticed. The parts with large number of minority samples have a more probability of being increased further, though parts with low minority samples have a sparse distribution after sampling [23].

SMOTE may also result in further increase of noise present in the data. This may happen when samples are interpolated from the initial noise samples, which are located between minority class samples and majority class neighbors. The method does not differentiate between overlapping class groups from so called safe regions due to which it results in the generation of noise and problem of over-generalization [24,25]. It also fails to overcome the problem of small disjoints [26].

Despite the problems, SMOTE has been extensively used in the area of class imbalance by different researchers because of its simplicity and effective results as compared to random sampling. Various modifications of the SMOTE have been developed to resolve the weakness. They may be modified according to the required goal, while aiming minority class or majority class, to solve the problem of noise and to combat within-class imbalance. Among its other variants such as Borderline-SMOTE [27], emphasized in the selection of the class region. Here the random selection of the samples selected in the SMOTE is replaced by the instances closed to the border of the class. The labels of the samples are considered for the purpose of interpolation or discarded as the noise. Another method is Cluster-SMOTE [28] that employs clustering procedure k-means, to cluster the instances of the minority class and applied SMOTE within the selected clusters. This technique boosts the regions of the class where the oversampling of the instances is required. It does not specify the method opted to select the number of clusters. Safe-Level-SMOTE [24], helps to solve the problem of noise by oversampling only at safe positions. Furthermore, it focuses on the imbalance in between the class and within the class, opposing the small disjoints problem by expanding sparse minority areas. It is very simple to implement and is used in many approaches. It is exclusively different from many existing methods due to its low complexity and its efficiency achieved while over-sampling synthetic samples on the basis of the density of the cluster.

Further to overcome the problem of noise generation, an approach called CURE–SMOTE [29] uses hierarchical clustering method CURE to clean the noise present prior to over-sampling. This will help in restricting the generation of the noise by the SMOTE and possible imbalances within the minority classes are avoided. Santos et al. [28] applied k-means clustering over the entire data regardless of the class labels, where clusters with the few samples are chosen for over-sampling using SMOTE. The method helps to balance dataset but does not solve the problem of class-imbalance. Douzas et al. [30] proposed k-means SMOTE for class imbalance problem. The algorithm uses k-means clustering algorithm to cluster the data into k clusters.

3 Proposed Model

In this section, the proposed model Majority Clustering for Balanced Classification using SMOTE (MCBC-SMOTE) is presented in detail. The proposed model consists of three main steps (i) Data pre-processing (ii) The majority class is divided into the number of clusters using k-means clustering where the number of cluster is calculated using elbow method. (iii) Instances of minority class are generated using SMOTE.

The proposed MCBC-SMOTE employs clustering of the majority class using k-means clustering where the numbers of clusters are calculated using elbow method. The minority class is oversampled with the help of SMOTE. The proposed algorithm helps to balance the skewed dataset and helps to prevent the over generalization of the resampled data by restricting the size of the minority class. It allows the in-between class imbalance and within class imbalance. Equations and mathematical expressions must be inserted in the main text.

3.1 Data Pre-processing

The data is cleaned using efficient preprocessing technique. In this, features are selected using correlation and *p*-value test. Correlation can be represented by the population correlation coefficient is $\rho\,(X, Y) = \mathrm{corr}\,(X, Y)$ given as:

$$\rho\,(X, Y) = corr\,(X, Y) = \frac{cov\,(X, Y)}{\sigma_X \sigma_Y} = \frac{E\,[(X - \mu_X)\,(Y - \mu_Y)]}{\sigma_X \sigma_Y} \tag{1}$$

The correlation matrix is calculated and is used to compare the features for the process of cleaning. The features with a higher correlation value than the specific threshold value are removed. The parameters to be used are selected based on the effect of *p*-value. The normalized data is used for the purpose of resampling by selecting minority and majority classes.

3.2 Majority Clustering for Balanced Classification

After the data is pre-processed, the next step employs popular k-means clustering algorithm on the majority data to cluster the input space into k groups. Majority Clustering for Balanced Classification (MCBC) method selects an optimal number of clusters to divide majority class data using elbow method [20,21]. This approach allows minimizing a sum of squared errors (SSE).

$$SSE = \sum\nolimits_{k=1}^{K} \sum\nolimits_{x_i \epsilon X} \|x_i - c_k\| \tag{2}$$

where $k = N_c$ for optimal number of clusters, x_i data present in each cluster and c_k is the predicted cluster centre. Here cluster_n is set manually.

Algorithm: Elbow Method

1. Compute k-means clustering algorithm on majority samples by varying values of ***k*** from 1 to cluster_n.
2. The total within-cluster sum of square error (SSE) represented by Eq. (2) is calculated for each ***k***.
3. Plot the curve with the calculated value of SSE for the value of ***k*** as shown in Fig. 1.
4. The location of the curve where the elbow is formed is assumed as the indicator to consider appropriate number of clusters.

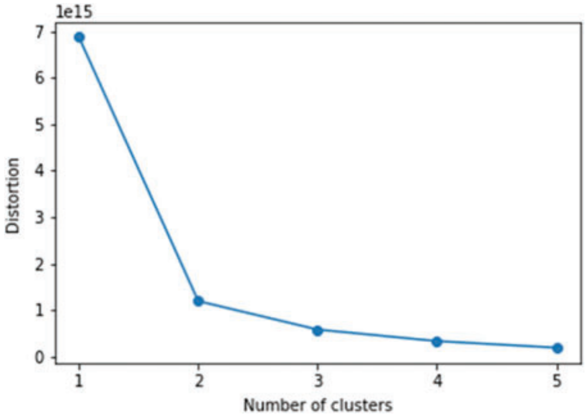

Figure 1: Graph of elbow method for financial stress dataset

Fig. 1 represents the graph of elbow method for optimal value of k. At k = 2, elbow of the curve is formed. The number of cluster obtained from the elbow method, is used to cluster the majority class. The majority data M_1 is divided into N_c number of clusters using k-means clustering and a new class label is assigned to each cluster. This operation reduces the number of data-points per class as the number of majority classes increases to N_c. This converts the binary-class problem to a multi-class problem with $N_c + 1$ as the total number of classes. $(N_c + 1)_{th}$ class is minority class.

3.3 Majority Clustering for Balanced Classification

The minority class sample $M_{2,}$ is oversampled with the number of samples N_m^* equal to the average size of the majority class cluster as

$$N_m^* = \frac{size\ of\ M_1}{N_c} \tag{3}$$

This is achieved by customizing the widely popular SMOTE to yield a limited number of samples. The minority sample M_2 is oversampled to the size of average majority class sample N_m^*. While using SMOTE to oversample the minority class samples, decision boundaries are taken in between the clusters to avoid the noise generation from the overlapping clusters as shown in Fig. 2. To interpolate synthetic samples, SMOTE chooses a random minority sample $\vec{a}$ within the minority class, a neighbouring minority sample $\vec{b}$ is selected randomly considering decision boundary and it determines a new sample $\vec{x}$ by randomly interpolating as shown in Eq. (4). The new interpolated samples are generated till the required number is reached.

$$\vec{x} = \vec{a} + w * \left(\vec{b} - \vec{a}\right) \tag{4}$$

where w is defined as random weight in [0, 1].

Fig. 3 graphically represents the proposed MCBC-SMOTE. In Fig. 3, feature selection is done using data pre-processing, then data is divided for training and testing purpose. The training data is oversampled as per the proposed MCBC-SMOTE. The balanced data obtained after applying the proposed technique is predicted by using different classifiers.

A high level illustration of the algorithm is given in Fig. 4. In Fig. 4a, data points are represented as majority and minority class. Fig. 4b shows the clustering majority class M_1 into N_c number of clusters. Fig. 4c shows the up-sampling of the minority class. Fig. 4d shows depiction of the minority class.

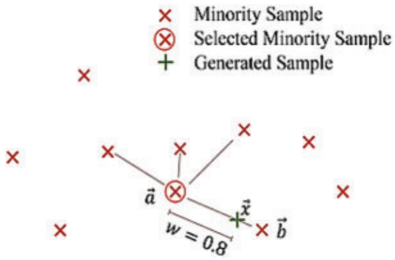

Figure 2: Interpolation of the synthetic sample using SMOTE

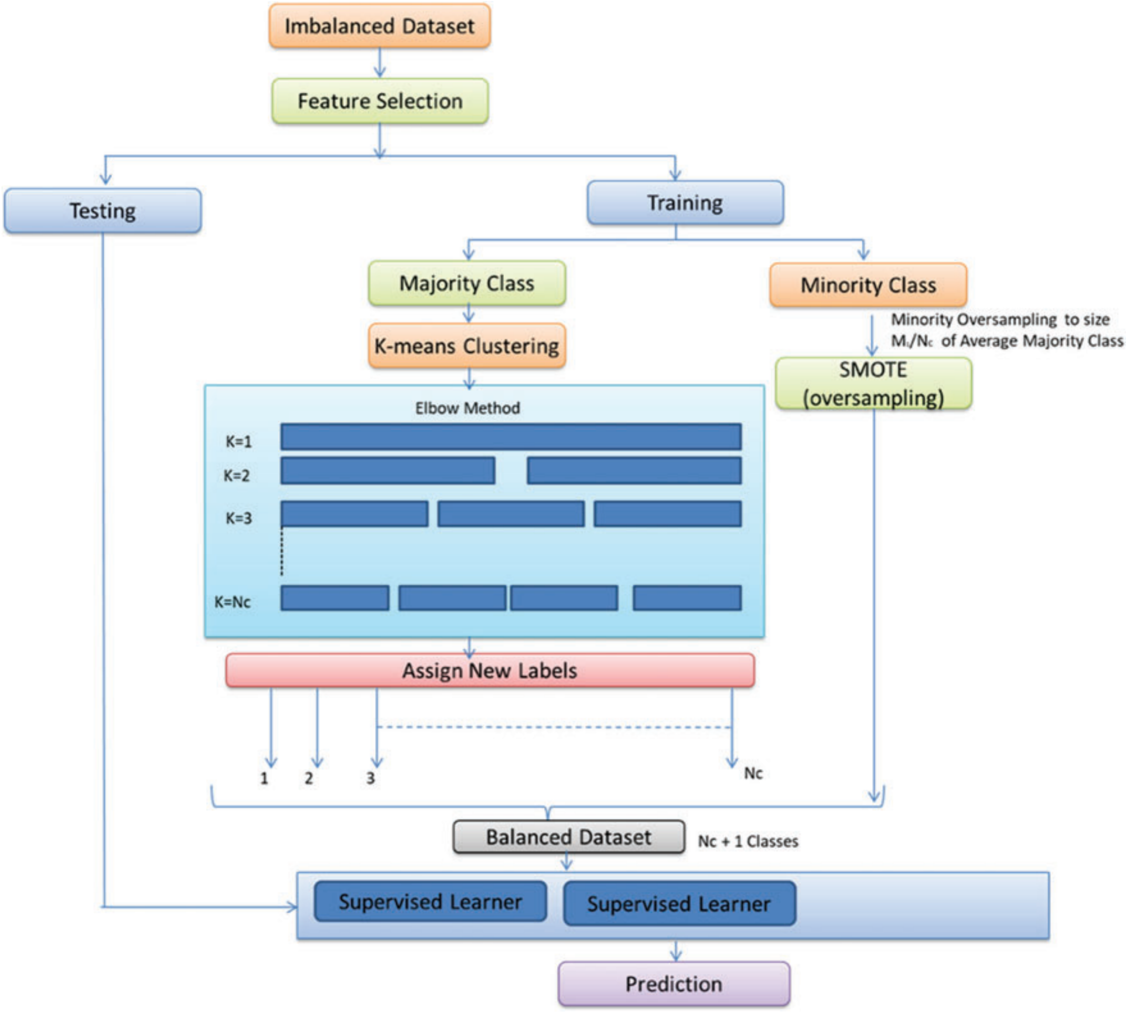

Figure 3: Graphical representation of the MCBC-SMOTE

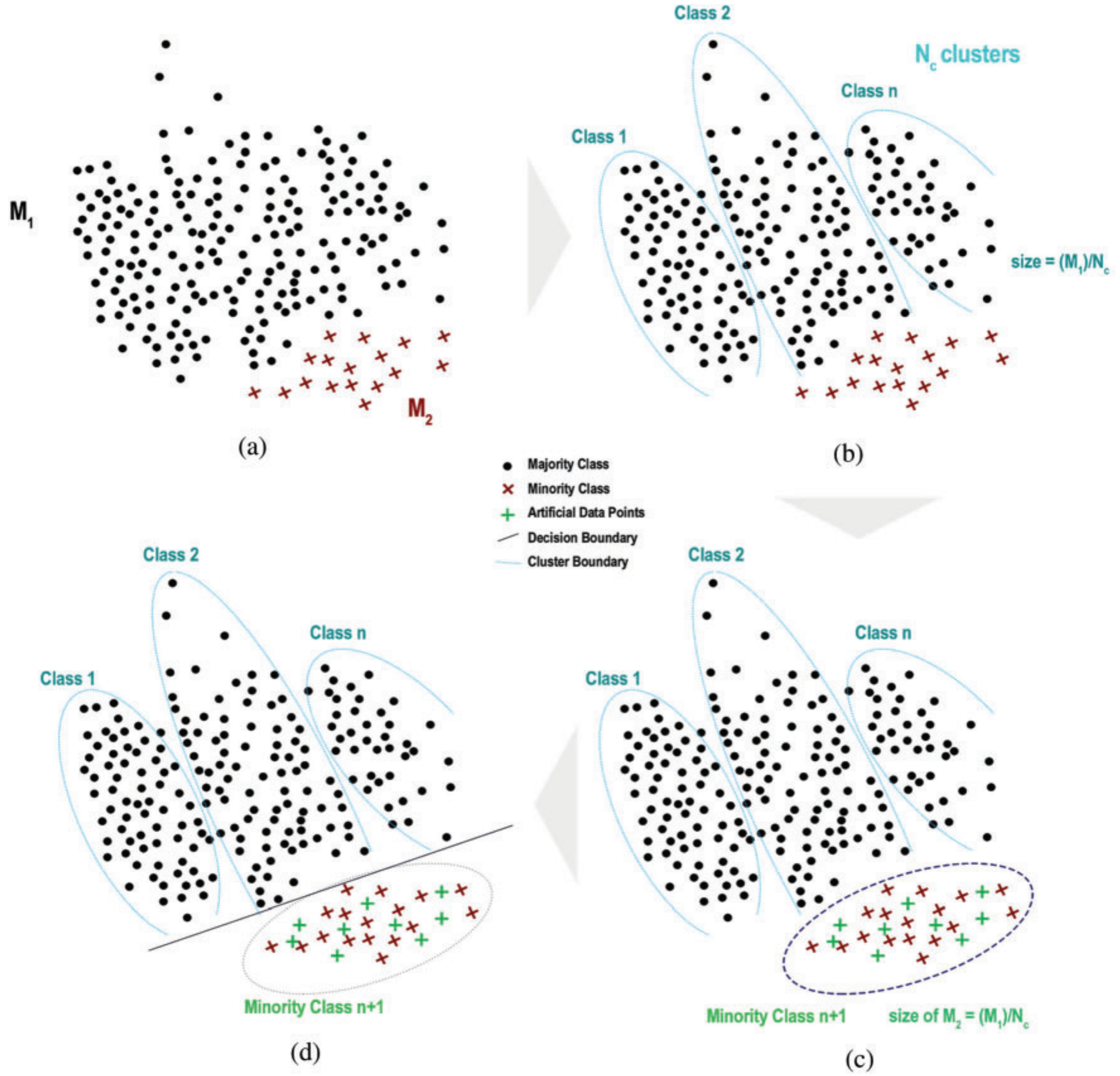

Figure 4: Process of majority clustering for balanced classification. (a) Data points represented as majority and minority. (b) Clustering majority class into n number of clusters giving n classes. (c) Minority class is up-sampled using SMOTE to the size of one cluster of majority class. (d) Decision boundaries are obtained between all classes. Only Minority class is taken into account

4 Experimental Framework

To assess the performance of the proposed model, various datasets used, evaluation metrics and classifiers are discussed in this section. For each dataset, performance metrics are evaluated by averaging their results across a fixed number of iterations for each classifier, respectively. In addition to the metrics, the macro average and weighted average is also calculated.

4.1 Datasets

The first data set used is a WNS dataset [31]. It is an employee dataset, collected over time of employee features in relation to the class 'is_promoted' to identify if they are promoted or not. It is a binary imbalanced class data. With the minority class (1 in is_promoted) with the ratio ∼0.0931, the dataset is highly imbalanced. The dataset presents promotions corresponding to various employee IDs. Here we have 4668 promotions against 50,140 not promoted. It contains both numerical and non-numerical features like department, region, education, gender, etc. PCA (Principal component analysis) is applied over various columns to obtain dummies for multi-category classes and are converted to binary classes. For instance, the class region has 34 categories, and hence dummies are created to have binary labels with respect to each region. In another example, gender is the only categorical column with two classes, so there is no need for creating dummy variables for gender column. Further, null values are also eliminated. The data is cleaned for further application of the proposed technique on the dataset.

Credit Card Fraud Detection [32] is used as the second data set: This dataset is of European cardholders consists of transactions made by them in September 2013 using a credit card. In this dataset, 492 frauds have occurred out of 284,807 transactions that happened in two days. The classes present in the dataset are extremely unbalanced and skewed; the target class (frauds) account for 0.172% of all transactions. It comprises of only numerical values of the input, which are the outcome attained after applying PCA transformation. The data set does not provide complete original features due to privacy issues. Features V1, V2, .., V28 are the principal components obtained by PCA. 'Time' and 'Amount' are the features which are not transformed. Feature 'Time' taken in seconds, defines the elapsed time in between each transaction and the first transaction in the dataset. Transaction amount is defined by 'Amount'. This attribute can be used for example-dependent cost-sensitive learning. The value of the feature 'Class' acts as the response variable and it takes value 1 in case of fraud and 0 otherwise.

Financial Distress [33] is the third data set. It specifies the prediction of final distress for a sample of companies. The data represents the sample companies in the first column. The second column represents different time periods that the data belongs to. The length of the time series varies between 1 and 14 for all companies. For example, company 1 is financially healthy at time 4, but company 2 is distressed at time 14. This makes the dataset a class imbalance as well as multivariate time series classification, thus, adding the diversity to the results. There are some financial and non-financial characteristics of sampled companies, denoted as x1 to x83 columns as features. The target variable 'Financial Distress', if it is greater than ∼0.50 then the company is considered healthy (0), otherwise, it would be considered distressed (1).

4.2 Evaluation Metrics

Generally, if the distribution of the class is not uniform, not all the traditional assessment metrics are suitable for the evaluation of the outcomes. There are some evaluation metrics which have been widely used to measure the performance with imbalanced data. In the process of evaluation, ground truth of every observation is compared with the calculated results from the classifier. A confusion matrix is defined in Fig. 5 to understand the alignment of predictions when compared with the true distribution. A prediction from the minority class specifies a positive outcome. It is an infrequent event when an instance is a part of the minority class. In those scenarios, the majority class is taken as negative. All the necessary information predicted by classifiers are analysed as per the confusion matrix

[34,35]. However, when the comparison is being made in between the different classifiers or assessing a single classifier in variable scenarios, the observations of the confusion matrix are non-trivial.

	P Positives	N Negatives	
PP Predicted Positives	TP True Positives	FP False Positives	Precision $\frac{TP}{PP}$
PN Predicted Negatives	FN False Negatives	TN True Negatives	
	Sensitivity / Recall $\frac{TP}{P}$	Specificity $\frac{TN}{N}$	

Figure 5: Confusion matrix

The widely used metrics for the problem of classification is accuracy and error rate.

$$Accuracy = \frac{TP + TN}{P + N} \qquad ErrorRate = 1 - Accuracy \tag{5}$$

These metrics when used for analysis for imbalanced datasets, generally show biased towards majority class. For example, in imbalanced datasets, a predicted accuracy for a naïve classifier which calculates all observations as negative would achieve 99% accuracy but in actually 1% of instances are only positive. Here the metrics fails to show the results correctly and predicts false results.

The weighted mean of sensitivity and precision is defined as F1-score, or F-measure. The F1-score can also be defined as the overall exactness and completeness of positive predictions.

$$F1 = \frac{(1 + \alpha) \times (sensitivity \times precision)}{sensitivity + \alpha \times precision} \tag{6}$$

However, to determine a novel method to oversampling, we place our focus on the following unweighted metrics for the evaluation, along with their macro average and weighted average results across both classes [21,23].

- Precision
- F1-score
- Recall

4.3 Classifiers

The evaluation of the proposed technique MCBC-SMOTE with the other state of the art techniques, two widely used classifiers are taken to confirm that the outcomes achieved can be comprehensive and are not governed by any specific classifier. The classifiers are selected on the basis of number of hyper-parameters: classification techniques with small or no hyper-parameters are considered more promising due to their definite configuration.

4.3.1 KNN Classifier

The KNN classifier [36] achieves the classification by recognizing the nearest neighbours to a problem given and using those neighbours to identify the class of the problem. K-nearest neighbours (KNN) allocates an instance to the class with the measure of the most nearest neighbours belong to that class. The number of neighbours considered is determined by the method of hyper parameter, the value of K.

4.3.2 XGBoost Classifier

An ensemble technique used for the purpose of the classification is the Gradient boosting over decision trees, or can be termed as gradient boosting machine (GBM). A scalable decision tree is made at each phase of the algorithm, in binary classification problem. All the trees are adjusted with the specific observations which could not be accurately classified by the decision trees of former stages. The outcomes are made by the majority vote of all trees. In the algorithm of GBM, several simple models are combined (referred to as weak learners) to generate the functioning of an effective classifier. However, there is a difference in modelling details in XGBoost as compared to conventional GBM. Specifically, more regularized model formalization is used by XGBoost in order to overcome the problem of over-fitting, which helps to perform better [37].

4.4 Experimental Results and Discussions

To investigate whether the proposed method performs consistently better than others, a comparison is made with k-means SMOTE as the baseline method. k-means SMOTE is state of the art, well-evaluated over-sampler. We present our analysis with the help of confusion matrix results between k-means SMOTE and proposed method, MCBC-SMOTE for the three datasets and across both classifiers as well. Both methods are integrated with KNN and XGBoost classifier. After a direct comparison with the baseline method, k-means SMOTE, it is observed that precision has greatly improved for all the datasets and across both classifiers. The XGBoost classifier appears to profit most from the MCBC-SMOTE based algorithm for WNS dataset while the KNN classifier profits most from MCBC-SMOTE for Credit Card Fraud detection data. The biggest mean score improvements are also achieved using the MCBC based algorithm. It can be further observed that all classifiers benefitted from MCBC-SMOTE.

It has been observed from the results of the experimentation that the proposed methods with the combination of classifier proved to give improved results with every metric taken into consideration. Tabs. 1–3 show the F1-score attained by the combination of proposed method with the XGBoost and KNN classifier used for the three data sets. However, F1-score achieved by the above methods are highly influenced by the choice of metric and classifier for a particular dataset. Further Tab. 4 gives the accuracy of classification obtained after the application of the proposed technique (MCBC-SMOTE) and the baseline method (k-means SMOTE) on all the datasets. The results proved the validity of the proposed technique with both the classifiers in comparison to the baseline method.

Table 1: Evaluation of WNS datasets

Method	Metric	Precision	Recall	F1-score	Support
KNN Classifier (Baseline Method)	0	0.95	0.87	0.91	12571
KNN Classifier (MCBC-SMOTE)		**0.94**	**0.97**	**0.95**	12571
XGBoost (Baseline Method)		0.96	0.87	0.91	12571
XGBoost (MCBC-SMOTE)		**0.94**	**1.00**	**0.97**	12571
KNN Classifier (Baseline Method)	1	0.27	0.52	0.32	1131
KNN Classifier (MCBC-SMOTE)		**0.42**	**0.25**	**0.32**	1131

(Continued)

Table 1: Continued

Method	Metric	Precision	Recall	F1-score	Support
XGBoost (Baseline Method)		0.29	0.58	0.38	1131
XGBoost (MCBC-SMOTE)		**0.93**	**0.28**	**0.44**	1131
KNN Classifier (Baseline Method)	Macro avg.	0.61	0.70	0.63	13702
KNN Classifier (MCBC-SMOTE)		**0.68**	**0.61**	**0.63**	13702
XGBoost (Baseline Method)		0.62	0.73	0.65	13702
XGBoost (MCBC-SMOTE)		**0.93**	**0.64**	**0.70**	13702
KNN Classifier (Baseline Method)	Weighted avg.	0.90	0.84	0.87	13702
KNN Classifier (MCBC-SMOTE)		**0.89**	**0.91**	**0.90**	13702
XGBoost (Baseline Method)		0.90	0.85	0.87	13702
XGBoost (MCBC-SMOTE)		**0.94**	**0.94**	**0.92**	13702

Table 2: Evaluation of credit card fraud detection dataset

Method	Metric	Precision	Recall	F1-score	Support
KNN Classifier (Baseline Method)	0	1	1	1	71089
KNN Classifier (MCBC-SMOTE)		**1**	**1**	**1**	71089
XGBoost (Baseline Method)		1	0.99	1	71089
XGBoost (MCBC-SMOTE)		**1**	**1**	**1**	71089
KNN Classifier (Baseline Method)	1	0.53	0.87	0.66	113
KNN Classifier (MCBC-SMOTE)		**0.85**	**0.79**	**0.82**	113
XGBoost (Baseline Method)		0.13	0.89	0.23	113
XGBoost (MCBC-SMOTE)		**0.82**	**0.74**	**0.78**	113
KNN Classifier (Baseline Method)	Macro avg.	0.76	0.93	0.83	71202
KNN Classifier (MCBC-SMOTE)		**0.92**	**0.89**	**0.91**	71202
XGBoost (Baseline Method)		0.57	0.94	0.61	71202
XGBoost (MCBC-SMOTE)		**0.91**	**0.87**	**0.89**	71202
KNN Classifier (Baseline Method)	Weighted avg.	1	1	1	71202
KNN Classifier (MCBC-SMOTE)		**1**	**1**	**1**	71202
XGBoost (Baseline Method)		1	0.99	0.99	71202
XGBoost (MCBC-SMOTE)		**1**	**1**	**1**	71202

Table 3: Evaluation of financial distress dataset

Method	Metric	Precision	Recall	F1-score	Support
KNN Classifier (Baseline Method)	0	0.97	0.91	0.94	874
KNN Classifier (MCBC-SMOTE)		**0.97**	**0.93**	**0.95**	874
XGBoost (Baseline Method)		0.97	0.96	0.97	874
XGBoost (MCBC-SMOTE)		**0.97**	**0.97**	**0.97**	874
KNN Classifier (Baseline Method)	1	0.21	0.45	0.28	44
KNN Classifier (MCBC-SMOTE)		**0.23**	**0.43**	**0.30**	44
XGBoost (Baseline Method)		0.37	0.50	0.42	44

(Continued)

Table 3: Continued

Method	Metric	Precision	Recall	F1-score	Support
XGBoost (MCBC-SMOTE)		**0.45**	**0.48**	**0.46**	44
KNN Classifier (Baseline Method)	Macro avg.	0.59	0.68	0.61	918
KNN Classifier (MCBC-SMOTE)		**0.60**	**0.68**	**0.62**	918
XGBoost (Baseline Method)		0.67	0.73	0.69	918
XGBoost (MCBC-SMOTE)		**0.71**	**0.72**	**0.72**	918
KNN Classifier (Baseline Method)	Weighted avg.	0.93	0.89	0.91	918
KNN Classifier (MCBC-SMOTE)		**0.93**	**0.90**	**0.92**	918
XGBoost (Baseline Method)		0.95	0.93	0.94	918
XGBoost (MCBC-SMOTE)		**0.95**	**0.95**	**0.95**	918

Table 4: Measurement of accuracy of classification

Technique\Dataset	WNS	Credit Card Fraud Detection	Financial Distress
KNN Classifier Baseline Method)	0.84	1	0.89
KNN Classifier (MCBC-SMOTE)	**0.91**	**1**	**0.90**
XGBoost (Baseline Method)	0.85	0.99	0.93
XGBoost (MCBC-SMOTE)	**0.94**	**1**	**0.95**

Further, Fig. 6 shows macro average i.e., across both classes, F1-score and precision improvements for Credit card Fraud detection dataset. XGBoost benefitted most from the MCBC method coupled with SMOTE. Fig. 7 shows macro average F1-score and precision improvements for WNS employee dataset. Fig. 8 shows macro average F1-score and precision improvements for Financial Distress dataset. Although the choice of metric and classifier highly influences absolute scores pertaining to two classes and score differences between the methods, their results follow the same trend as average results and are omitted for clarity. In all the cases MCBC-SMOTE outperforms k-means SMOTE alone. On average, MCBC-SMOTE achieves a precision improvement of 0.12 with XGBoost and 0.33 with KNN classifier.

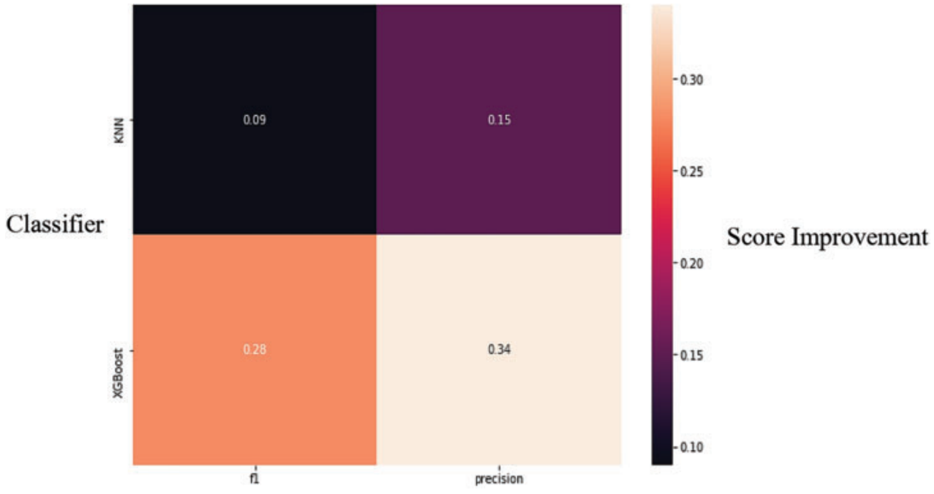

Figure 6: Score improvement of the MCBC-SMOTE *vs.* k-means SMOTE over credit card data

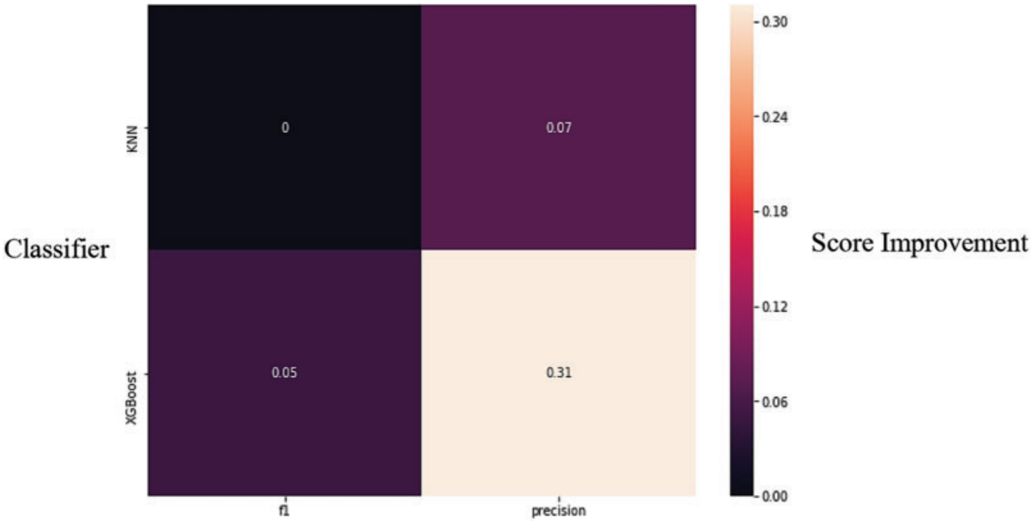

Figure 7: Score improvement of the MCBC-SMOTE method *vs*. k-means SMOTE over WNS data

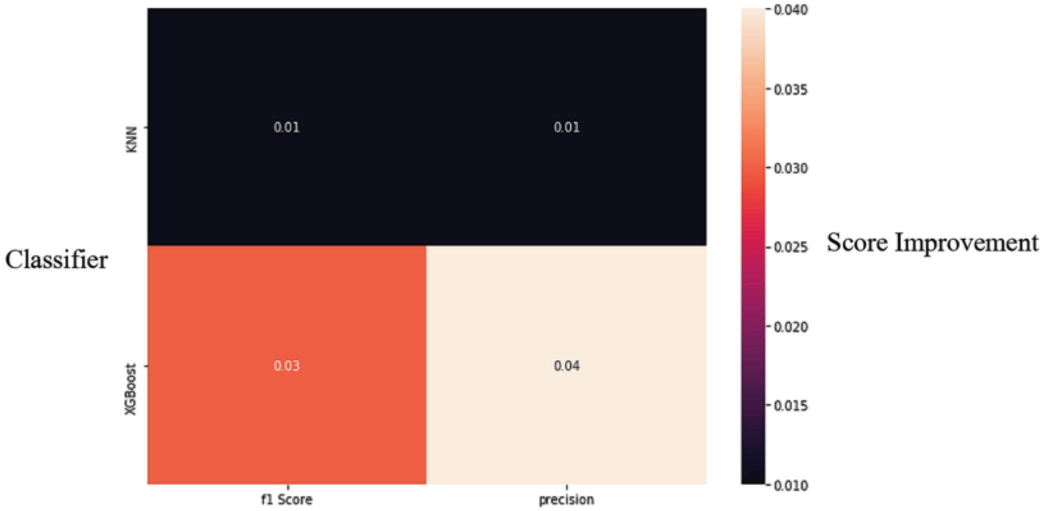

Figure 8: Score improvement of the MCBC-SMOTE method *vs*. k-means SMOTE over financial distress dataset

By observing at the gains of the proposed method in comparison to the baseline method, it is found that both classifiers used in the experimentation benefit from the proposed hybrid model methodology over k-means SMOTE. Significantly, the proposed method consistently outperformed state-of-the-art oversampling and random oversampling. The prime achievement can be found in the problems of classification which are highly skewed and have limited data. The results are statistically robust and proved the effectiveness for the process of imbalanced data classification. The three datasets used are instances of naturally occurring data lying at ends of a range of skewness, size and even time variance. The results, hence, underline its efficiency and scope in more datasets and over various applications. Thus, without high complexity, high efficiency is achieved using the proposed MCBC-SMOTE.

5 Conclusions and Future Scope

In this paper, a novel hybrid method of Majority Clustering for Balanced Classification (MCBC) integrated with Synthetic Minority Oversampling Technique (SMOTE) as MCBC-SMOTE is proposed to handle the problems that occur while classifying the highly skewed data-sets. The proposed method allows limiting the oversampling to the smaller size of a cluster formed from the majority-class. This cluster is obtained by clustering the majority class into n-clusters, obtained via the elbow method in k-means clustering. This leads to synthetic data-generation without reaching outside the safe boundary space. The proposed method achieves the benefits of oversampling with minimal distortion of the naturally occurring imbalance in the dataset. This ability is of great significance for highly skewed datasets. The proposed method MCBC-SMOTE with different multi-class classifiers proved to give robust results for three diverse datasets. Sparsely populated minority clusters are generated with more synthetic samples, which resolve the problem of within-class imbalance. Lastly, it also discouraged the problem of over-fitting by allowing genuinely new points using SMOTE rather than replicating existing ones.

In the proposed technique, weight parameters of the SMOTE are chosen randomly due to which SMOTE interpolates the data blindly and results into the problem of collinearity between the generated and existing data points. In the future, some adaptive method can be developed to tune the weight parameter depending upon the characteristics of the datasets. Evolutionary techniques can be integrated with SMOTE to optimize these parameters adaptively.

Acknowledgement: This research was supported by Taif University Researchers Supporting Project number (TURSP-2020/254), Taif University, Taif, Saudi Arabia.

Funding Statement: This research was supported by Taif University Researchers Supporting Project number (TURSP-2020/254), Taif University, Taif, Saudi Arabia.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] N. V. Chawla, N. Japkowicz and A. Kolcz, “Editorial: Special issue learning imbalanced datasets,” *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 1–6, 2004.

[2] D. A. Cieslak, N. Chawla and A. Striegel, “Combating imbalance in network intrusion datasets,” in *IEEE Int. Conf. on Granular Computing*, Atlanta, GA, USA, pp. 732–737, 2006.

[3] D. Williams, V. Myers and M. Silvious, “Mine classification with imbalanced data,” *IEEE Geoscience and Remote Sensing Letters*, vol. 6, no. 3, pp. 528–532, 2009.

[4] S. Kotsiantis, D. Kanellopoulos and P. E. Pintelas, “Handling imbalanced datasets: A review,” *GESTS International Transactions on Computer Science and Engineering*, vol. 30, no. 1, pp. 25–36, 2006.

[5] D. Devarriya, G. Gulati, V. Mansharamani, A. Sakalle and A. Bhardwaj, “Unbalanced breast cancer data classification using novel fitness functions in genetic programming,” *Expert Systems with Applications*, vol. 140, no. 14, pp. 1–21, 2020.

[6] S. Fotouhi, S. Asadi and M. W. Kattan, “A comprehensive data level analysis for cancer diagnosis on imbalanced data,” *Journal of Biomedical Informatics*, vol. 90, no. 103089, pp. 1–30, 2019.

[7] G. M. Weiss and F. Provost, “Learning when training data are costly: The effect of class distribution on tree induction,” *Journal of Artificial Intelligence Research*, vol. 19, pp. 315–354, 2003.

[8] G. Wu and E. Chang, “Class-Boundary alignment for imbalanced dataset learning,” in *ICML 2003 Workshop on Learning from Imbalance Data Sets II*, Washington, DC, pp. 49–56, 2003.

[9] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince and F. Herrera, “A review on ensembles for the class imbalance problem: Bagging-, boostong-and hybrid-based approaches,” *IEEE Transactions on Systems, Man and Cybernetics-Part C: Applications and Reviews*, vol. 42, no. 4, pp. 463–484, 2012.

[10] N. V. Chawla, L. O. Hall, K. O. Bowyer and W. P. Kegelmeyer, “SMOTE: Synthetic minority oversampling technique,” *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[11] S. Kotsiantis and P. E. Pintelas, “Mixture of expert agents for handling imbalanced datasets,” *Annals of Mathematics, Computing and TeleInformatics*, vol. 1, no. 1, pp. 46–55, 2003.

[12] M. Kubat and S. Matwin, “Addressing the curse of imbalance training sets: One sided selection,” in *Proc. of the Fourteenth Int. Conf. on Machine Learning*, pp. 179–186, 1997.

[13] Z. Zheng, Y. Cai and Y. Li, “Oversampling method for imbalanced classification,” *Computing and Informatics*, vol. 34, pp. 1017–1037, 2016.

[14] D. A. Cieslak and N. V. Chawla, “Learning decision trees for unbalanced data,” in *Machine Learning and Knowledge Discovery in Databases. ECML PKDD*. Berlin, Heidelberg: Lecture Notes in Computer Science, Springer, pp. 241–256, 2008.

[15] N. Japkowicz and S. Stephen, “The class imbalance problem: A systematic study,” *Intelligent Data Analytics Journal*, vol. 6, no. 5, pp. 429–450, 2002.

[16] S. Barua, M. M. Islam, X. Yao and K. Murase, “Mwmote-majority weighted minority oversampling technique for imbalanced data set learning,” *IEEE Transaction Knowledge Data Engineering*, vol. 26, no. 2, pp. 405–425, 2014.

[17] P. Kaur and A. Gosain, “FF-SMOTE: A metaheuristic approach to combat class imbalance in binary classification,” *Applied Artificial Intelligence*, vol. 33, no. 5, pp. 420–439, 2019.

[18] S. Cateni, V. Colla and M. Vanucci, “A method for resampling imbalanced datasets for binary classification tasks for real world problems,” *Neurocomputing*, vol. 135, no. 1, pp. 32–41, 2014.

[19] G. Y. Wong, F. H. F. Leung and S. H. Ling, “A hybrid evolutionary preprocessing method for imbalanced datasets,” *Information Sciences*, vol. 454–455, pp. 161–177, 2018.

[20] W. Wei, J. Li, L. Cao, Y. Ou and J. Chen, “Effective detection of sophisticated online banking fraud on extremely imbalanced data,” *World Wide Web-internet and Web Information Systems*, vol. 16, no. 4, pp. 449–475, 2013.

[21] M. Herland, T. M. Khoshgoftaar and R. A. Bauder, “Big data fraud detection using multiple medicare data sources,” *Journal of Big Data*, vol. 5, no. 1, pp. 29–38, 2018.

[22] F. Provost, “Machine learning from imbalanced data sets 101,” in *Proc. of the AAAI’2000 Workshop on Imbalanced Data Sets*, pp. 1–3, 2000.

[23] R. C. Prati, G. E. A. P. A. Batista and M. C. Monard, “Learning with class skews and small disjuncts,” In: Bazzan, A. L. C., Labidi, S. (eds) *Advances in Artificial Intelligence–SBIA. Lecture Notes in Computer Science*, vol. 3171, Springer, Berlin, Heidelberg. 2004. https://doi.org/10.1007/978-3-540-28645-5_30.

[24] C. Bunkhumpornpat, K. Sinapiromsaran and C. Lursincap, “Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem,” in *Advances in Knowledge Discovery and Data Mining. PAKDD 2009*, In: T. Theeramunkong, B. Kijsirikul, N. Cercone, TB. Ho (Eds.), Berlin, Heidelberg: Lecture Notes in Computer Science (LNCS 5476), Springer, pp. 475–482, 2009.

[25] P. Kaur and A. Gosain, “Robust hybrid data-level sampling approach to handle imbalanced data during classification,” *Soft Computing*, vol. 24, no. 20, pp. 15715–15732, 2020.

[26] P. Kaur and A. Gosain, “GT2FS-SMOTE: An intelligent oversampling approach based upon general type-2 fuzzy sets to detect web spam,” *Arabian Journal of Science and Engineering*, vol. 46, pp. 1–18, 2020.

[27] H. Han, W. Y. Wang and B. H. Mao, “Borderline-smote: A new over-sampling method in imbalanced data sets learning,” in *Proc. of the 2005 Int. Conf. on Advances in Intelligent Computing, LNCS 3644*, Berlin, Heidelberg, Springer-Verlag, pp. 878–887, 2005.

[28] M. S. Santos, P. H. Abreu, P. J. Garcia-Laencina, A. Simao and A. Carvalho, “A new cluster-based oversampling method for improving survival prediction of hepatocellular carcinoma patients,” *Journal of Biomedical Informatics*, vol. 58, no. 10, pp. 49–59, 2015.

[29] L. Ma and S. Fan, “Cure-smote algorithm and hybrid algorithm for feature selection and parameter optimization, based on random forests,” *BMC Bioinformatics*, vol. 18, no. 1, pp. 169, 2017.

[30] G. Douzas, F. Bacao and F. Last, “Oversampling for imbalanced learning based on k-means and smote,” *Information Sciences*, vol. 465, no. 1, pp. 1–20, 2018.

[31] Wns_inno dataset. [Online]. Available: https://www.kaggle.com/rednivrug/wns-inno. 2018.

[32] Credit card fraud detection dataset. [Online]. Available: https://www.kaggle.com/mlg-ulb/creditcardfraud. 2018.

[33] Financial distress prediction dataset. [Online]. Available: https://www.kaggle.com/shebrahimi/financial-distress. 2018.

[34] N. Japkowicz, “Assessment metrics for imbalanced learning,” *Imbalanced Learning: Foundations, Algorithms, and Applications*, pp. 187–206, 2013.

[35] H. He and E. A. Garcia, “Learning from imbalanced data,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[36] G. Guo, H. Wang, D. Bell, Y. Bee and K. Greer, “KNN model-based approach in classification,” in *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. OTM 2003. Lecture Notes in Computer Science (LNCS 2888)*, In: R. Meersman, Z. Tari, D. C. Schmidt (Eds.), Berlin, Heidelberg: Springer, pp. 986–996, 2003.

[37] T. Chen and C. Guestrin, “XGBOOST: A scalable tree boosting system,” in *Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD ’16)*, New York, NY, USA, Association for Computing Machinery, pp. 785–794, 2016.