Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 46 (2025), No. 5, pp. 1705–1723
DOI : 10.47974/JIOS-1946

DOFCM-SMOTE : A technique to handle class imbalance problem

Shweta Sharma *
National Remote Sensing Centre
Indian Space Research Organisation
Hyderabad
Telangana
India

and

University School of Information, Communication, and Technology
Guru Gobind Singh Indraprastha University
Delhi
India

Anjana Gosain †
University School of Information, Communication, and Technology
Guru Gobind Singh Indraprastha University
Delhi
India

Abstract

Class imbalance is a common issue in real-world datasets, where there are relatively few instances of the class of interest compared to others. The classifier's performance is highly affected by the impurities built-in with the data like imbalanced data, noise, and class overlapping; therefore, an oversampling technique is proposed by fusing density-oriented fuzzy-c means clustering and synthetic minority oversampling technique(DOFCM-SMOTE). The density-oriented approach is intended to find outliers to address the performance issue. The proposed method works in 3 phases: 1) It detects and eliminates outliers in the dataset, 2) It Assigns membership degrees to the given samples by fuzzy c-means clustering, and 3) It then applies SMOTE to the minority cluster to balance the distribution of classes. To determine the performance, a decision tree classifier is employed as a learner. This study was conducted on ten publicly available data sets with varying imbalance ratios(IR) ranging from high to low. The findings of this study are centric on the process of clustering through

* *E-mail:* **shwetabhardwajj15@gmail.com; (Corresponding Author)**
shwetanrsc.isro@gmail.com
† *E-mail:* **anjana_gosain@yahoo.com**

which outliers are detected even before minority and majority class clusters are identified, and a general pre-processing technique, to balance the distribution of classes. The introduced approach is compared with four other oversampling techniques and the best balancing algorithm is determined. The performances are accessed through AUC and G-mean, and it was reported that the proposed technique significantly improved and outperformed the other over-sampling methods.

Subject Classification: *68T05:Learning and adaptive systems in artificial intelligence*

Keywords: *Imbalanced-data, Oversampling, Density-oriented, Clustering.*

1. Introduction

Problem Statement: In the last few years, data has become a vital component in numerous real-world applications, but it often poses a significant challenge when it is imbalanced. The Class Imbalance Problem (CIP) is the term used to describe the situation when one class, referred to as the minority class is substantially having fewer occurrences than the other class, known as the majority class. Traditional classifiers often favor the majority class and have difficulty in accurately classifying minority class instances. This challenge is exacerbated when the dataset contains impurities, such as outliers, which are frequently overlooked or dismissed as noise. Imbalanced classification is prevalent in fields like biomedical analysis, text classification, fraud detection, outlier detection, and information filtering, underscoring its relevance and impact.

Motivation: Addressing the issues[3] caused by imbalanced databases is motivated by the growing significance of data in real-world applications. The presence of imbalanced data adversely impacts the efficacy of machine learning (ML) classifiers, thereby necessitating the identification of effective solutions. Researchers have proposed various methods to tackle the CIP, broadly divided into data level, algorithm level, and hybrid approaches. Each of these approaches has its advantages and limitations. Data-level techniques involve balancing the class distribution through either oversampling or undersampling. However, these approaches come with trade-offs, as oversampling can inflate dataset size and computational complexity, while undersampling may result in the loss of crucial information. Algorithm-level methods modify the internal workings of classifiers, but this can be a complex and timeconsuming process. Hybrid approaches aim to combine the strengths of both data and algorithm-level techniques.

Innovation: This paper introduces an innovative hybrid approach called DOFCM-SMOTE. It addresses noisy data and outliers using the DOFCM algorithm. DOFCM operates within a fuzzy clustering framework and effectively identifies and eliminates noisy data points or outliers, a task that traditional fuzzy clustering algorithms struggle with. DOFCM, in contrast to the conventional Fuzzy c-means algorithm, has the capability to detect data points that are far from the core distribution of clusters, making it particularly valuable in imbalanced datasets. After implementing the noise reduction step, the dataset is balanced using the SMOTE approach. Synthetic minority class instances are produced by

SMOTE by considering the k-positive nearest neighbor of the original instances, avoiding random duplication of minority class data. Various SMOTE variants like Fuzzy-SMOTE (FSMOTE) [32], CBSO [4], Borderline-SMOTE(BSMOTE) variants [15] [17], Safe-level SMOTE(SSMOTE) [6], K-means SMOTE(KSMOTE) [23] based on [10] [11] [31], FF-SMOTE [21] are also available.

This paper follows the structure: Section 2 reviews prior research on addressing imbalanced data. Section 3 describes the proposed DOFCM-SMOTE method. Section 4 presents the experimental results using benchmark datasets, demonstrating the effectiveness of the proposed method. A summary of the study and recommendations for further research are given in Section 5.

2. Literature Review

As discussed earlier, several [16],[29] to imbalanced problems have been proposed in the literature. In this work, We adopted a data-level resampling approach to balance the given dataset by adding through oversampling. SMOTE is one of the famous oversampling techniques that has proven to be a benchmark for solving imbalance problems. With this approach, synthetic minority data points are created by oversampling the minor class.

However, SMOTE reduces over-fitting to some extent but does not work well with noisy data as it can lead to an increase of noisy instances while selecting the random points. Several modifications of SMOTE have been suggested to address the issue of noise, which aims to combat the original algorithmâ€™s flaws.

Han, Wang, and Mao [17] proposed BSMOTE in 2005 to enhance SMOTE by generating new instances only at the borderline. This approach assumes that instances near the class boundary are more likely to be misclassified. Minority class instances are categorized into noise, borderline, and safe based on their locality. However, BSMOTE's key limitation is that it oversamples only borderline instances rather than the entire minority class, as SMOTE does.

Bunkhumpornpat, Sinapiromsaran, and Lursinsap [6] proposed SSMOTE, which assigns a safe level to minority class instances before generating new samples, based on their neighborhood. However, the definition of the safe level is unclear and must be determined first.

Haibo et al. [18] proposed ADASYN, which focuses on hard-to-learn minority instances to generate new samples, reducing learning bias from class imbalance. The algorithm assigns weights to minority samples to create varying amounts of new data, but it may sometimes produce undesirable results and fail to assign appropriate weights.

Some notable work to handle the CIP has adopted clustering-based oversampling methods [20], [21], [22] also. The first cluster-based oversampling method was proposed as agglomerative hierarchical clustering[9]. The data sets are categorized into smaller clusters using this procedure, and cluster centroids are created as synthetic instances.

Barua et al. [4] proposed CBSO to minimize poor synthetic instance generation. Inspired by ADASYN, it uses unsupervised clustering to confine new

samples within the minority class, preventing overlap with the majority class and isolating noisy instances. However, its weight assignment, based on majority samples in k-nearest neighbors, may not always be optimal.

Easter et al. [14] proposed the DBSCAN algorithm, to identify clusters of random shapes to handle the CIP. It oversamples the region with higher density while the data points outside the cluster are considered outliers.

In 2012, Bunkhumpornpat, Sinapiromsaran, and Lursinsap [7] proposed DBSMOTE, inspired by BSMOTE, to oversample overlapping and safe regions using DBSCAN-based clustering. New instances are generated along the shortest path to the pseudo-centroid of a minority cluster, avoiding noisy areas. However, it tends to create more samples near the dataset's core rather than the border.

To overcome the limitation of ADASYN(poor distribution of generated synthetic samples), Barua et al. [5] proposed ProWsyn in 2013. To ensure equitable distribution of newly formed samples in the minority class, this technique generates effective weights for the minority data instances based on their proximity to the boundary.

In 2014, Lunardon, Menardi, and Torelli [24] a random oversampling technique was proposed, ROSE, which mitigates the possibility of model overfitting. It is based on the smoothed bootstrap approach for generating new artificial data from the classes. It creates synthetic data based on the density underlying the data, ensuring the data distribution is appropriate after balancing.

Tang and He [30] proposed Kernel ADASYN in 2015, an over-sampling approach based on kernel density estimation to combat the CIP. An adaptive resampling distribution is constructed to synthesize minority class data. Initially, the distribution is estimated using kernel density estimation techniques and is later adjusted by assigning weights according to the difficulty levels of different minority class samples.

To address class imbalance, Nekooeimehr and Lai-Yuen [26] proposed A-SUWO in 2016, using semi-supervised hierarchical clustering to group minority cases. It identifies challenging instances near the boundary, determines sub-cluster sizes based on misclassification error, and adjusts oversampling rates via weighted instance assignment to prevent over-generalization.

In 2017, Douzas and Bacao [13] proposed SOMO. The initial data points are divided into multiple clusters using a self-organizing map algorithm in this approach. It is widely accepted because it creates two-dimensional maps using a neural network and competing learning. Depending on cluster density, distribution among the created clusters is modified.

In 2018, Last, Douzas, and Bacao [23] proposed KSMOTE, combining k-means clustering and SMOTE to rebalance class distribution. It generates more samples in key regions with many minority cases, clustering the entire dataset regardless of class label density. While it helps prevent overfitting, it does not account for noisy data points.

Vats and Sagar [31] explores how well K-means clustering works on the Hadoop framework, testing different approaches like MapReduce, IEC-based K-means, Mahout, and Spark. The results show that IEC-based K-means improves with more iterations, while Mahout processes data faster than Spark.

KFCM-SMOTE-SVM algorithm introduced by Huang, Zhang, and Yuan [19] appends the kernel method into FCM and combines it with kernel SMOTE to work with high-dimensional feature space. The kernel method was incorporated to improve the quality of synthetic minority instances. Also, fuzzy logic has collaborated with SMOTE. The objective of introducing the FCM-SMOTE algorithm is to deal with uncertainties present in data.

This study introduces an innovative hybrid approach called DOFCM-SMOTE for handling class imbalance, as it helps deal with uncertainties underlying data, outliers, for instance. Our major goal is to increase the accuracy of the minority class by removing outliers from the dataset and adding new minority examples to balance out data distributions that are skewed. This principle should be considered when determining the minority samples for generating new synthetic samples, as well as the appropriate number of synthetic samples to be produced for each minority instance.

3. Related Work

This section discusses the background information necessary for the proposed study, along with a brief overview of the FCM clustering strategy, the DOFCM approach, and the SMOTE technique which serves as the base for our proposed method.

3.1 *FCM*

Fuzzy c-means, introduced by [27], is one of the most well-known fuzzy clustering algorithms. This methodology incorporates the predetermined number of clusters within the dataset prior to allocating the membership degree for each cluster center, thereby facilitating the construction of the membership matrix. It optimizes the objective function (J_{FCM}) as:

$$J_{FCM} = \sum_{k=1}^{c} \sum_{i=1}^{n} u_{ik}^{m} d_{ik}^{2} \quad [1]$$

where u_{ki} is the membership value of x_i the data point for cluster 'k', and $d_{ik} = \|x_i - v_k\|$ is the Euclidean distance of data point 'i' to its centroid of cluster 'k', which gratifies the given relationship:

$$\sum_{l=1}^{m} u_{ik} = 1; i = 1,2,\ldots.n \quad [2]$$

Here, m stands for the fuzzification factor and J_{FCM}. Minimization of J_{FCM} is carried out by a fixed-point iterative approach referred to as the alternating optimization technique. The FCM algorithm is sensitive to outliers even if it is resilient since it gives noisy points and outliers equal membership. To avoid the noise constraint, Possibilistic C-Means clustering(PCM) [22] was introduced for reducing distance between data points and centroid. However, it is susceptible to initializations, resulting in overlapping or identical clusters. To address the notable limitations of the FCM algorithm, namely its susceptibility to noise, has introduced noise clustering. Though it deems noise as a separate class, it still fails to detect outliers between the centroids. Its main target is to minimize outliers' influence rather than identify them. To overcome the problems of noise clustering, DOFCM is proposed, which is discussed in the next section.

Algorithm 1 DOFCM

Input: Dataset (X), iteration count, cluster count $(K = c + 1)$, and stopping criteriaa ε.

Output: The matrix for cluster centroid, vector for outliers, and matrix for membership.

Outlier Identification:

for $i = 1, 2, 3, \ldots . n$ **do**
    Find the counts of data points in each point's immediate vicinity. i.e., $\eta^{i}_{neighborhood}$
    Select η_{max}
    Compute neighbourhood membership, $M^{i}_{neighborhood}(X)$ for each point
end for
Based on no of instances in X, a threshold value α is selected from whole range of membership values in the neighborhood.
Identify outliers with the given value of α.
    Clustering Process:
Find the initial centroid v_k.
Set all outliers' membership to zero after initializing the membership $u_k i$.
for $i = 1, 2, 3, \ldots . max_{iter}$ **do**
    Upgrade all centroids v_k with equation
    Upgrade all membership values u_{ki} with equation [4]
    Find objective function O_n
    If $E_n \leftarrow \varepsilon$, end; otherwise, Find $E_n = max|O_n - O^{n-1}|$.
    Else the value is; $n = n + 1$
end for
end of pseudo-code

3.2 *DOFCM*

The technique proposed by [28], known as Density-oriented Fuzzy C-means(DOFCM), as demonstrated in Algorithm 1, seeks to reduce the effect of noise in fuzzy clustering by detecting outliers before the clustering procedure. It measures the density of a given data point based on its neighborhood. A data point 'i''s degree of participation in dataset 'X' is defined as follows:

$$M^{i}_{neighborhood}(X) = \frac{\eta^{i}_{neighborhood}}{\eta_{max}} \quad [3]$$

The objective function of DOFCM is given below

$$u_{ki} = \begin{cases} \dfrac{1}{\Sigma_{j=1}^{c}\left(\frac{d_{ki}}{d_{ji}}\right)^{\frac{2}{m-1}}} & k, i, if\ M^{i}_{neighbourhood} \geq \alpha \\ 0(zero); & M^{i}_{neighbourhood} < \alpha \end{cases} \quad [4]$$

In this way, DOFCM identifies outliers by alloting them zero fuzzy membership so that it could not affect the locus of centroids.

3.2 *SMOTE*

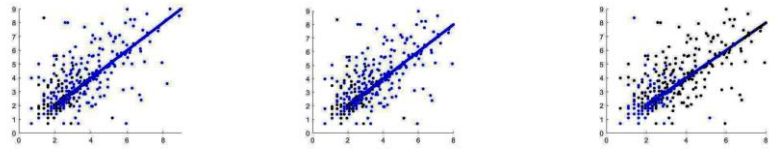
The extensively utilized oversampling method, SMOTE generates a synthetic minority by interpolating between the candidate minority instance and one of its K-nearest neighbors. It somewhat minimizes overfitting issue as it does not generate new examples by replication, like random oversampling [16]. Rather than selecting neighbors uniformly, they are randomly chosen based on oversampling rate, defined by the ratio between the number of majority class instances and minority class instances [15]. The steps of the SMOTE algorithm are presented in Algorithm 2, adapted from [8],

Algorithm 2 $\boldsymbol{SMOTE(T, N, k)}$

Input: Minority class examples T; SMOTE Rate $N\%$; k is the number of closest neighbors.
Output: S samples of synthetic minority class.
(* If N is less than 100%, generate the samples of smaller class*)
if $N < 100$ **then**
    Set the minority class's T samples at random.
    $T = (N/100) * T$
    $N = 100$
end if
$N = (int)(N/100)$
for $i \leftarrow 1$ **to** T **do**
    For i, find k closest neighbors and save in $nnarray$
    while $N \neq 0$ **do**
        $nn = random(0,1)$
        for $att = 1$ to num do **do**
            Define att as the index of the feature/attribute
            $diff = M[nn\_arr[nn]][att] - M[i][att]$ Calculate the difference $diff$ for the current feature
            $g = random(0,1)$
            $Syn[new\_indx][att] = M[i][att] + (g \times diff)$
        for for
        $new\_indx = new\_indx + 1$
        $N = N - 1$
    end while
end for

end of pseudo-code

The synthetic minority instances increase the total number of instances in the minority class, thereby achieving a more balanced distribution of the dataset depending upon the percentage of oversampling as shown in Figure 1. However, it does not effectively handle noisy data, as randomly selecting points for interpolation can amplify the noise level. Hence to tackle the noise problem, DOFCM-SMOTE is proposed.



(a) 60% of oversampling (b) 80% of oversampling (c) 100% of oversampling

Figure 1
Oversampling using SMOTE Technique

4. Proposed Approach: DOFCM-SMOTE

4.1 *Principle*

We propose a hybrid algorithm, DOFCM-SMOTE, in this work as presented in Algorithm 3, where we combine a clustering algorithm (DOFCM) and an oversampling algorithm (SMOTE). The introduced algorithm precisely tackles noise or outliers in a fuzzy clustering framework. DOFCM pre-processes the dataset by eliminating outliers(based on a selected threshold value, α) even before making clusters by considering the data points' density. Further, it uses the FCM approach to create clusters, and then it uses SMOTE algorithm is used to balance the distribution between the minority and majority classes. The principle idea behind this approach is that in order to generate synthetic samples for every point within a cluster, the surrounding area must include a minimum number of points. The framework of the proposed model is shown in figure 2.

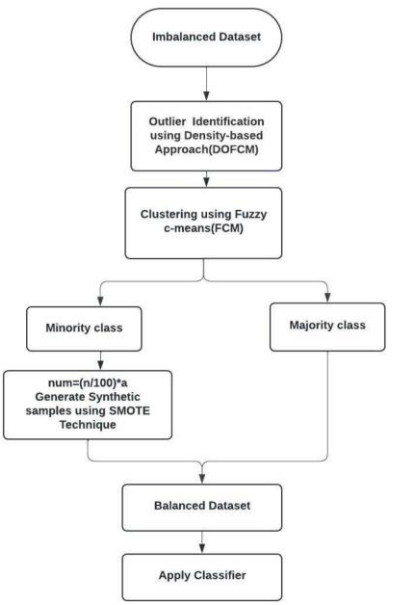


Figure 2
Proposed model for DOFCM-SMOTE

Algorithm 3: DOFCM-SMOTE(M, N)

Require: Number of minority class instances(M), percentage of oversampling(N)
Ensure: Balanced data(D)

Let $\mathcal{P}^+$ be the set of minority class instances (positive instances).
for all $i \in \mathcal{P}^+$
Determine the neighborhood association of an instance i in dataset X.

$$M^i_{neighbourhood}(X) \equiv \frac{n^i_{neighbourhood}}{n_{max}}$$

Identify and remove outliers as given in equation
After removing outliers, FCM is used to cluster given data into minority (Min) and majority (Maj) class.
Calculate how many points will be generated artificially.

$$n = \left(\frac{n}{100}\right) \times a$$

Min [][]: Array of original minority instances
Find the number of random instances from Min
$Syndata$[][]: Array of synthetic instances.
repeat
    for $i \leftarrow 1\ to\ R$ **do**
        call SMOTE()
    end for
    Combine Syndata and Min and keep it in a balanced Dataset.
until$Min \leq Maj$
procedure: SMOTE(M, K, N)

Require: Number of minority observations(M), Number of nearest neighbors(k), percentage of oversampling(N)
Ensure: balanced MinClass(m)

Get neighbors (M, K) using euclidean based KNN approach
for $i \leftarrow M$ **do**
    for $j \leftarrow N$ **do**
        Select a set of neighbors at random
        Evaluate the distinction between the individual neighbors and the minority number feature vector.
        Difference found will be multiplied by random numbers in range(0,1)
        Then, it added back to feature vector
        Store the current data points to Syndata
    end for
    return Syndata[ ][ ]
end for
end of pseudo-code

4.2 *Steps involved in the proposed algorithm: DOFCM-SMOTE*

In traditional SMOTE, each sample has equal emphasis and would be blindly resampled n times. In contrast, DOFCM-SMOTE utilizes the factor of SMOTE and conducts oversampling for every minority sample based on its locality in the feature space and density. Thus, DOFCM-SMOTE oversamples each region in advance without modifying the total number of synthetic instances to be created. The detailing of the algorithm is ellaborated in steps below:

1. We detect outliers prior to the clustering process to reduce noise sensitivity in fuzzy clustering. The algorithm first determines the neighborhood membership of point $'i'$ from $'X'$ based on the density of an object in its local neighborhood which is measured using the following provided equation below.

   Let 'q' lies in proximity of point 'i', so 'q' should satisfy the following conditions:"

   $$\{q \in X | dist(i,q) \leq r_{neighborhood}\} \quad [5]$$

   where dis(i,q) is the distance between points 'i' and 'q', and denote the neighborhood's radius.

2. A threshold value $'\alpha'$ is selected, depending upon the data set density. For our case, this $'\alpha'$ is set to 0.06(as per the density of the LBTF dataset). An in-depth analysis of the dataset's density distribution revealed that a threshold of 0.06 effectively distinguishes outliers from the main data cluster. If the point's neighborhood membership is less than 0.06($'\alpha'$), After observing density, any value that falls outside the range of $M_{neighborhood}$ will be considered an outlier and can be selected, with the aim of minimizing its distance from zero. Finally, outliers are removed based on the below equation.

   $$M^{i}_{neighborhood} = \begin{cases} < \alpha; & outlier \\ \geq \alpha; & non-outlier \end{cases} \quad [6]$$

   In our case,

   $$outliers = (neibour_{mem} < 0.06)$$

3. FCM algorithm is afterwards applied to the newly created data set (after removing outliers from the real data set). It produces minority and majority clusters based on fuzzy logic. As density-oriented approach operatesindependently of the dataset's number of clusters, it does not incorporate any parameters that could influence the outcome of the clustering process, hence it was chosen randomly. Since our primary focus is on oversampling, we concentrate solely on the minority cluster, leaving the majority class unchanged.

4. As the minority class is the most prevalent one, we try to incorporate oversampling in the same class by taking into account the length of the minority class and the rate of oversampling based on the dataset taken.

   $$num(insgen) = \left(\frac{n}{100}\right) \times a \quad [7]$$

5. The feature difference between a minority class object and one of its randomly chosen k-nearest neighbors is calculated to create a synthetic instance. A random value between 0 and 1 is then added to this difference for each attribute. The adjusted differences are added to the original minority class object, producing a synthetic instance that is incorporated into the training set. Finally, for each attribute:

$$r \leftarrow (0,1)$$
$$diff \leftarrow X_f - Xn_f$$
$$syn = X_f + diff * r \quad [8]$$

5. Methods for evaluating Performance

For assessing the performance of our proposed algorithm, appropriate measures have been adopted while handling the class imbalance issue.

1. Confusion Matrix: It provides a tabular representation used to evaluate the performance of a classification model.
2. Accuracy: It is calculated by dividing the total number of predictions generated by the ML algorithm by the occurrences that were properly anticipated.

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \quad [9]$$

When classes in datasets are imbalanced, accuracy creates a misleading impression of accomplishment [25].

3. ROC AUC: For each potential classification threshold, the True Positive Rate (y-axis) is plotted against the False Positive Rate (x-axis) to create the Receiver Operating Characteristic (ROC) curve.

$$ROC - AUC = \frac{1+TP_{rate}-FP_{rate}}{2} \quad [10]$$
$$Precision(P) = \left(\frac{TP}{TP+FP}\right) \quad [11]$$
$$Recall(R) = \left(\frac{TP}{TP+FN}\right) \quad [12]$$
$$F1 - score = \left(\frac{2t*R*P}{R+P}\right) \quad [13]$$

4. The geometric mean,G-mean, a measure of central tendency (3). It is defined as follows:

$$G - Mean = \sqrt{TPR * TNR} \quad [14]$$

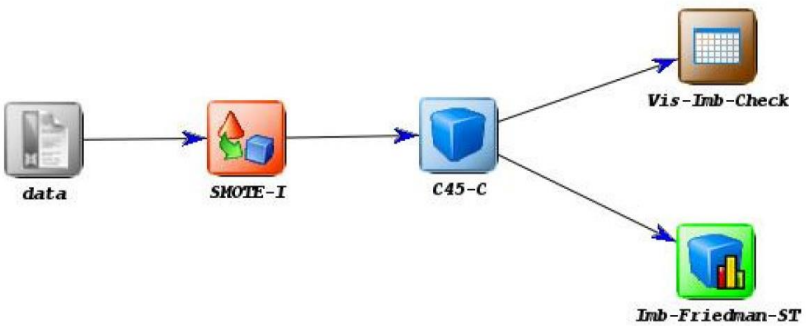


Figure 3
The KEEL software tool's setup

6. Experiment and Discussion

In this section, we demonstrate the working and performance of the proposed method, DOFCM-SMOTE. The experiment uses several benchmark datasets from WEBSPAM-UK2007 and KEEL repository with varying IRs, as shown in Table 1. All the analysis are performed using MATLAB Version 2021b to produce the results, and KEEL Tool [2] is used for the comparative analysis. This setup is shown in Figure 3. Firstly, the original dataset is divided into training and testing sets using a 5-fold cross-validation procedure, which averages the estimates from each fold. Next, we evaluate the proposed and other oversampling methods using ten real-world imbalanced datasets with varying IR. All experiments are conducted using the C4.5 decision tree as the base classifier.

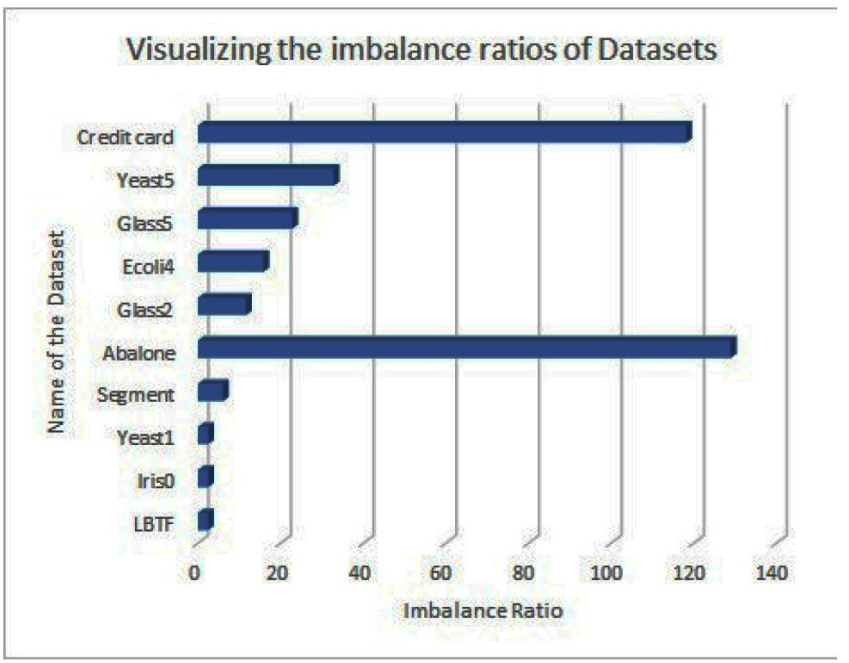


Figure 4
Imbalance ratios of datasets

6.1 *Dataset*

Experiments are conducted on ten imbalanced datasets from WEBSPAM-UK2007 [12] and KEEL dataset repository [1], with the IR ranging from 2 to 128.87, as shown in Figure 4. The dataset size varies from a minimum of 150 data points to a maximum of 150780 data points, and the dimension ranges from 4 to 138 as seen in Table 1. In this research, we examine only the two class datasets: minority and majority.

Table 1
Overview of the Dataset

S No.	Dataset	No. of Data Points	Dimensions	Minority Class %	IR
1	LBTF	3849	138	1122	2.43
2	iris0	150	4	33.33	2.46
3	yeast1	1484	8	28.91	2.46
4	Segment0	2308	19	14.26	6.02
5	Glass2	214	9	8.78	11.59
6	Ecoli4	336	7	6.74	15.80
7	Glass5	214	9	4.20	22.78
8	Yeast5	1484	8	2.96	32.78
9	Credit card	150780	31	490	118.12
10	Abalone	4174	8	0.77	128.87

Table 2
Cutting-edge oversampling techniques

Terminology	Description
SMOTE	It creates new instances by randomly selecting points between a candidate instance and its neighboring instances [8]
BSMOTE	This technique oversamples all samples exists within the borderline [17]
Safe-level	It finds the safe-level and generates a new instance closer to the minority one [6]
AHC	Undersampling using K-means and oversampling using agglomerative hierarchical clustering [9]

6.2 *Classification Performance*

In our study, we compare the proposed algorithm, DOFCM-SMOTE, with the four existing algorithms, namely SMOTE, BSMOTE, SSMOTE, and AHC on ten real-world datasets. For the experimental setup, we use ten datasets of varying IRs. The properties of the dataset are shown in Table 2. We have used the KEEL Tool to analyze these five algorithms. We have considered AUC and G-mean scores for the efficacy of our proposed algorithm as the outcome indicate, when the IR increases, the biasness towards the majority class creates faulty results if we have taken accuracy as the measure. Hence more precious metrics like AUC and g-mean are taken to conduct this study. DOFCM-SMOTE calculates the membership degree of each minority example to determine its association with the minority class. Any membership degree lower than 0.06 (for the LBTF dataset) is identified and removed by addressing the outliers present in the dataset.

The performance of this algorithm is recorded with 5-fold cross-validation with different parameters like the number of neighbors selected, k, and the oversampling rate α (may be varied as per the dataset), and the results are collected repeatedly for further analysis.

The performance of any oversampling technique is highly impacted by the noise present in the dataset, which is evenly shown by the given comparative study. The effectiveness of the proposed algorithm in identifying outliers and

oversampling the minority class is evaluated by analyzing classification performance using the same classifier across 10 different datasets. We have taken the observation on both the training(tr) and testing(tst) dataset as shown in Table 3, 4, and It has been observed that the proposed method performs effectively when IR is high, like in the abalone,yeast5, and credit card dataset. However, the results are not that much improved with glass2, sement0, and iris datasets with low IR. In the case of these datasets, the proposed algorithm behaves like other oversampling techniques such as BSMOTE or SMOTE, as the IR of these datasets was not that high. BSMOTE works well with the LBTF, yeast1, segment0 datasets, while the safe level-smote performed worst while considering these mentioned datasets. The density-oriented fuzzy clustering approach, considering the outliers already eliminated from the original dataset, presents an enhanced framework for clustering data with unique attributes while addressing the issue of class imbalance, as each cluster can better recognise local information with a balanced class distribution. Hence, it can be concluded from the above observation that the proposed method performed well with any dataset ranging from high to low category.

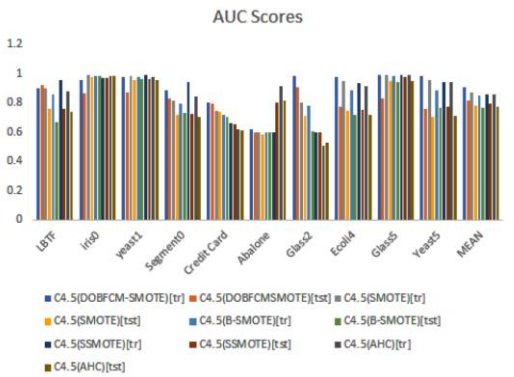


Figure 5
AUC Scores for given datasets

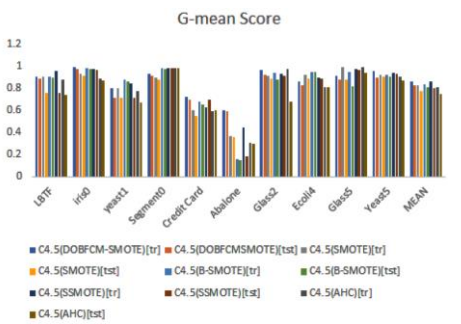


Figure 6
G-mean Scores for given datasets

Table 3

The AUC of the datasets from Decision tree classifier

Dataset	IR	Proposed [tr]	Proposed [tst]	SMOTE [tr]	SMOTE [tst]	BSMOTE [tr]	BSMOTE [tst]	SSMOTE [tr]	SSMOTE [tst]	AHC [tr]	AHC [tst]
LBTF	2.43	**0.903**	0.887	**0.903**	0.757	**0.904**	0.896	**0.956**	0.762	0.881	0.739
iris0	2.46	**0.984**	0.978	0.933	0.918	**0.986**	0.976	0.978	0.969	0.889	0.872
yeast1	2.46	0.805	0.711	0.805	0.711	**0.878**	**0.865**	**0.846**	0.715	0.778	0.674
Segment0	6.01	0.928	0.912	0.898	0.882	**0.988**	0.978	0.988	0.987	0.983	0.982
Credit Card	118.12	**0.723**	0.701	0.598	0.549	0.683	0.649	0.629	0.695	0.596	0.599
Abalone	128.87	**0.604**	0.591	0.366	0.356	0.159	0.147	0.448	0.183	0.307	0.296
Glass2	11.59	**0.963**	0.921	0.912	0.890	0.940	0.881	0.928	0.918	0.977	0.675
Ecoli4	15.8	0.858	0.824	0.927	0.889	**0.951**	0.946	0.897	0.885	0.814	0.808
Glass5	22.78	0.917	0.882	**0.997**	0.882	0.947	0.823	0.975	0.965	0.997	0.942
Yeast5	32.78	**0.957**	0.893	0.927	0.903	0.921	0.910	0.938	0.931	0.906	0.872
MEAN		0.865	0.830	0.827	0.774	0.836	0.807	0.858	0.801	0.813	0.746

Table 4

The G-mean of the datasets from Decision tree classifier

Dataset	IR	Proposed [tr]	Proposed [tst]	SMOTE [tr]	SMOTE [tst]	BSMOTE [tr]	BSMOTE [tst]	SSMOTE [tr]	SSMOTE [tst]	AHC [tr]	AHC [tst]
LBTF	2.43	0.903	0.925	0.903	0.757	0.860	0.666	0.956	0.762	0.881	0.739
iris0	2.46	0.960	0.865	0.992	0.976	0.986	0.983	0.974	0.972	0.986	0.983
yeast1	2.46	0.980	0.87	0.983	0.954	0.976	0.964	0.989	0.966	0.980	0.957
Segment0	6.01	0.886	0.833	0.815	0.714	0.797	0.731	0.945	0.723	0.846	0.700
Credit Card	118.12	0.801	0.798	0.746	0.739	0.719	0.701	0.659	0.652	0.621	0.609
Abalone	128.87	0.616	0.601	0.598	0.584	0.601	0.598	0.600	0.799	0.917	0.8136
Glass2	11.59	0.987	0.905	0.805	0.711	0.778	0.605	0.599	0.595	0.506	0.527
Ecoli4	15.8	0.975	0.775	0.948	0.746	0.888	0.716	0.938	0.751	0.9151	0.720
Glass5	22.78	0.941	0.831	0.969	0.947	0.984	0.939	0.965	0.975	0.978	0.953
Yeast5	32.78	0.985	0.762	0.954	0.701	0.889	0.767	0.939	0.771	0.9453	0.708
MEAN		0.908	0.816	0.873	0.783	0.848	0.767	0.859	0.796	0.859	0.771

7. Conclusion

Data impurities like outliers are subjected to deteriorate the classifiers' performance, evenly shown by this comparative study. This paper proposed a hybrid approach, DOFCM-SMOTE, using oversampling approach and clustering. The data was divided into minority and majority sections using a fuzzy clustering technique after local density information was taken into account to find outliers. The data distribution is then balanced by increasing the amount of the minority cluster using an oversampling method, i.e, SMOTE algorithm. The classification performance of 10 real-world benchmark datasets from publically available repositories, is evaluated using AUC and G-mean scores as shown in Figures 5, 6. It is evident from our observations of the training and testing datasets that the suggested method performs well on the datasets with high IRs like the abalone,yeast5, and credit card dataset. We analyze the proposed approach with traditional techniques such as; SMOTE, BSMOTE, SMOTE, and AHC. The results indicate that the proposed method surpasses existing approaches across most available datasets, outperforming conventional techniques, especially considering a high IR(often the case in real-world data). Furthermore, by creating more minority training examples, the proposed approach expands the minority class border and enhances the classifier performance on imbalanced datasets.the proposed method improves the classifier performance on imbalanced datasets by broadening the minority class boundary by generating additional minority training instances. In subsequent work, we intend to optimize oversampling techniques through exploring the integration of more precise conditional input to generate minority instances based on specific class labels or attribute values.

Statements and Declarations

Conflicts of interests The authors confirm that none of the conclusions in this study could have been influenced by any personal or financial conflicts of interest. We affirm that this research is free from conflicts of interest.

Data availability The data supporting this study's conclusions is publicly accessible in a repository, WEBSPAM-UK2007[12] and KEEL dataset repository[1].

Financial or non-financial interests The authors declare that they possess no relevant financial or non-financial interests to disclose. Furthermore, they affirm that there are no conflicts of interest related to the content of this article.

References

[1] J. Alcalá-Fdez, L. Sánchez, S. García, M. J. del Jesus, S. Ventura, J. M. Benítez, and F. Herrera, "KEEL: A software tool to assess evolutionary algorithms for data mining problems," *Soft Computing*, vol. 13, no. 3, pp. 307–318 (2009).

[2] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic & Soft Computing*, vol. 17, pp. 255–287 (2011).

[3] R. Barandela, J. S. Sánchez, V. García, and E. Rangel, "Strategies for learning in class imbalance problems," *Pattern Recognition*, vol. 36, no. 3, pp. 849–851 (2003).

[4] S. Barua, M. M. Islam, K. Murase, and X. Yao, "A novel synthetic minority oversampling technique for imbalanced data set learning," in Proc. Int. Conf. Neural Information Processing, Springer, pp. 735–744 (2011).

[5] S. Barua, M. M. Islam, K. Murase, and X. Yao, "ProWSyn: Proximity weighted synthetic oversampling technique for imbalanced data set learning," in Proc. Pacific-Asia Conf. Knowledge Discovery and Data Mining, Springer, pp. 317–328 (2013).

[6] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-Level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem," in Proc. Pacific-Asia Conf. Knowledge Discovery and Data Mining, Springer, pp. 475–482 (2009).

[7] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "DB-SMOTE: Density-based synthetic minority over-sampling technique," *Applied Intelligence*, vol. 36, no. 3, pp. 664–684 (2012).

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357 (2002).

[9] G. Cohen, M. Hilario, H. Sax, A. Hugonnet, and V. François, "Learning from imbalanced data in surveillance of nosocomial infection," *Artificial Intelligence in Medicine*, vol. 37, no. 1, pp. 7–15 (2006).

[10] R. N. Dave, "Robust fuzzy clustering algorithms," in Proc. 2nd IEEE Int. Conf. Fuzzy Systems, IEEE, pp. 1281–1286 (1993).

[11] R. N. Davé and R. Krishnapuram, "Robust clustering methods: A unified view," *IEEE Transactions on Fuzzy Systems*, vol. 5, no. 2, pp. 270–293 (1997).

[12] C. Denoyer, "Web spam challenge 2007," [Online]. Available: http://www.corpusetud.univ-mlv.fr/experiments/webspam/

[13] G. Douzas and F. Bacao, "Self-organizing map oversampling (SOMO) for imbalanced data set learning," *Expert Systems with Applications*, vol. 82, pp. 40–52 (2017).

[14] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining (KDD), pp. 226–231 (1996).

[15] A. Fernández, S. Garcia, F. Herrera, and N. Chawla, "SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary," *J. Artif. Intell. Res.*, vol. 61, pp. 863–905 (2018).

[16] A. Ghazikhani, H. S. Yazdi, and R. Monsefi, "Class imbalance handling using wrapper-based random oversampling," in Proc. 20th Iranian Conf. Electrical Engineering (ICEE), IEEE, pp. 611–616 (2012).

[17] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in Int. Conf. Intelligent Computing, Springer, pp. 878–887 (2005).

[18] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in Proc. 2008 IEEE Int. Joint Conf. Neural Networks (IEEE World Congress on Computational Intelligence), pp. 1322–1328 (2008).

[19] X. Huang, C.-Z. Zhang, and J. Yuan, "Predicting extreme financial risks on imbalanced dataset: A combined kernel FCM and kernel SMOTE based SVM classifier," *Comput. Econ.*, vol. 56, no. 1, pp. 187–216 (2020).

[20] P. Kaur and A. Gosain, "An intelligent undersampling technique based upon intuitionistic fuzzy sets to alleviate class imbalance problem of classification with noisy environment," *Int. J. Intell. Eng. Inform.*, vol. 6, no. 5, pp. 417–433 (2018).

[21] P. Kaur and A. Gosain, "FF-SMOTE: A metaheuristic approach to combat class imbalance in binary classification," *Appl. Artif. Intell.*, vol. 33, no. 5, pp. 420–439 (2019).

[22] R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.,* vol. 1, no. 2, pp. 98–110 (1993).

[23] F. Last, G. Douzas, and F. Bacao, "Oversampling for imbalanced learning based on k-means and SMOTE," arXiv preprint arXiv:1711.00837 (2017).

[24] N. Lunardon, G. Menardi, and N. Torelli, "ROSE: A package for binary imbalanced learning," R J., vol. 6, no. 1 (2014).

[25] R. Malhotra and S. Kamal, "An empirical study to investigate oversampling methods for improving software defect prediction using imbalanced data," *Neurocomputing,* vol. 343, pp. 120–140 (2019).

[26] I. Nekooeimehr and S. K. Lai-Yuen, "Adaptive semi-unsupervised weighted oversampling (A-SUWO) for imbalanced datasets," *Expert Syst. Appl.,* vol. 46, pp. 405–416 (2016).

[27] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, "A possibilistic fuzzy c-means clustering algorithm," *IEEE Trans. Fuzzy Syst.,* vol. 13, no. 4, pp. 517–530 (2005).

[28] P. Kaur, I. Lamba, and G. Anjana, "DOFCM: A robust clustering technique based upon density," *Int. J. Eng. Technol.,* vol. 3, no. 3, p. 297 (2011).

[29] S. Sharma, A. Gosain, and S. Jain, "A review of the oversampling techniques in class imbalance problem," in Int. Conf. Innovative Computing and Communications, Springer, pp. 459–472 (2022).

[30] B. Tang and H. He, "KernelADASYN: Kernel based adaptive synthetic data generation for imbalanced learning," in 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 664–671 (2015).

[31] S. Vats and B. Sagar, "Performance evaluation of k-means clustering on Hadoop infrastructure," *J. Discrete Math. Sci. Cryptogr.,* vol. 22, no. 8, pp. 1349–1363 (2019).

[32] Y. Xu, C. Wu, K. Zheng, L. Yu, and G. Zhang, "Fuzzy–Synthetic Minority Oversampling Technique: Oversampling based on fuzzy set theory for Android malware detection in imbalanced datasets," *Int. J. Distrib. Sensor Netw.,* vol. 13, no. 4, Article ID 1550147717703116 (2017).

Received August, 2024