https://doi.org/10.1007/s42979-023-02249-3

ORIGINAL RESEARCH

A Diversity-Based Synthetic Oversampling Using Clustering for Handling Extreme Imbalance

Yuxuan Yang[1] · Hadi Akbarzadeh Khorshidi[1] · Uwe Aickelin[1]

Received: 3 April 2023 / Accepted: 12 August 2023

Abstract

Imbalanced data are typically observed in many real-life classification problems. However, mainstream machine learning algorithms are mostly designed with the underlying assumption of a relatively well-balanced distribution of classes. The mismatch between reality and algorithm assumption results in a deterioration of classification performance. One form of approach to address this problem is through re-sampling methods, although its effectiveness is limited; most re-sampling methods fail to consider the distribution of minority and majority instances and the diversity within synthetically generated data. Diversity becomes increasingly important when minority data becomes more sparse, as each data point becomes more valuable. They should all be considered during the generation process instead of being regarded as noise. In this paper, we propose a cluster-based diversity re-sampling method, combined with NOAH algorithm. Neighbourhood-based Clustering Diversity Over-sampling (NBCDO) is introduced with the aim to complement our previous cluster-based diversity algorithm Density-based Clustering Diversity Over-sampling (DBCDO). It first uses a neighbourhood-based clustering algorithm to consider the distribution of both minority and majority class instances, before applying NOAH algorithm to encourage diversity optimisation during the generation of synthetic instances. We demonstrate the implementation of both cluster-based diversity methods by conducting experiments over 10 real-life datasets with $\leq 5\%$ imbalance ratio and show that our proposed cluster-based diversity algorithm (NBCDO, DBCDO) brings performance improvements over its comparable methods (DB-SMOTE, MAHAKIL, KMEANS-SMOTE, MC-SMOTE).

Keywords Over-sampling · Diversity optimisation · Genetic algorithm · Imbalanced data · Clustering

Introduction

Imbalanced data are the situation whereby the proportion of "Negative" (majority class) instances is disproportionately larger to the number of instances marked as "Positive" (minority class). Without any treatment, this can negatively impact the performance of learning classifiers trained upon it as these classifiers would most likely interpret these minority instances as an outlier or anomaly [1]. Mainstream classification algorithms are typically designed with the goal of maximising predictive accuracy or minimising classification error, with the underlying assumption that distribution of instances between classes is relatively balanced. Therefore, this leads to a strong tendency for these classifiers to generate prediction of majority class, resulting in False Negative Rates (FNR) [2, 3]).

The treatment and handling of imbalanced data in current literature can be grouped into three main approaches, namely, cost-sensitive learning, ensemble-based methods,

This article is part of the topical collection "Advances on Computational Intelligence 2022" guest edited by Joaquim Filipe, Kevin Warwick, Janusz Kacprzyk, Thomas Bäck, Bas van Stein, Christian Wagner, Jonathan Garibaldi, H. K. Lam, Marie Cottrell and Faiyaz Doctor.

✉ Yuxuan Yang
yuyang@student.unimelb.edu.au

Hadi Akbarzadeh Khorshidi
hadi.khorshidi@unimelb.edu.au

Uwe Aickelin
uwe.aickelin@unimelb.edu.au

[1] School of Computing and Information Systems, The University of Melbourne, Grattan Street, Parkville, VIC, Australia

Published online: 08 November 2023

and re-sampling techniques. The target outcome for re-sampling techniques is the creation of a more balanced dataset for training and learning purposes by either random or synthetic measures. One variation of re-sampling is the application of over-sampling, where the most basic approach is Random Over-sampling. Throughout many years of research, there is also a wide variety of synthetic over-sampling approaches. Synthetic instances, as the name suggests, are not exact replications of the original minority instances. They aim to broaden the decision region compared to random over-sampling by reducing the likelihood of model overfitting. This leads to an improved False Negative Rate and helps enhanced performance of learning classifiers [4]. However, these algorithms have recently been proven to have a reduced effectiveness as they typically generate minority instances on a linear path on the "feature level" [5], and therefore prevent learning classifiers from obtaining a holistic view of the entire decision region of the minority class [6].

In our previous work, we illustrated how performance improvements can be obtained by introducing a strategy which optimises for diversity while protecting the integrity of the distribution of minority data space. We denote this algorithm as Density-based Clustering Diversity Over-sampling (DBCDO, previously named as CDO) [7]. The algorithm generates robust synthetic minority instances by taking a 2-step approach. A density-based clustering method is first applied to analyse and identify density distribution of minority instances. Synthetic data generation for each cluster is performed as the next step through NOAH algorithm to encourage diversity.

In this paper, we aim to extend the cluster-based diversity algorithm by incorporating a neighbourhood-based clustering algorithm. The proposed method is named as Neighbourhood-based Clustering Diversity Over-sampling (NBCDO). This paper also enhances the parameter selection for the original DADO and DIWO algorithms to achieve a more optimal result. Additionally, a comparison with alternative over-sampling approaches is conducted. With the two proposed cluster-based diversity algorithms, we have demonstrated significant improvement in handling extreme imbalanced scenario compared to the existing methods in the literature.

Related Work

SMOTE is a well-known synthetic over-sampling technique in literature [5]. Synthetic minority class instances are generated via a random selection of specified k-nearest neighbours of a minority sample, and apply a multiplier based on a uniform random distribution (0,1). This results in a "synthetic" instance which sits between the two minority points. SMOTE improved the performance of classifiers trained on imbalance dataset by expanding decision regions housing nearby minority instances as compared to basic random over-sampling which enhanced and narrowed decision regions with contrasting effects [5].

Recent studies have identified that traditional over-sampling methods (i.e. SMOTE) are limited to its casual tendency to generate synthetic instances, which sometimes extends into the input region of the majority class instances. This negatively impacts the performance of the subsequent learning classifier built [6, 8]. These studies have recognised the dual importance of maintaining the integrity of the minority sample region, in addition to boosting the diversity of minority class data. Subsequent studies have been conducted to address the above challenge.

ECO-ensemble is a cluster-based synthetic oversampling ensemble method [9]. Its underlying concept originates from the identification of suitable oversampling cluster regions with Evolutionary Algorithm (EA) to obtain the optimised ensemble. The SMOTE-Simple Genetic Algorithm (SMOTE-SGA) method is proposed to enhance diversity within the generated dataset [10]. The over-generalisation problem in SMOTE is addressed by the algorithm which determines instances to be generated and the number of synthetic instances created from the selected instance (sampling rate).

MAHAKIL is proposed with the purpose of generating more diverse synthetic instances [6]. It achieves this by pairing minority instances with previously generated synthetic instances to create instances inspired by the Chromosomal Theory of Inheritance. Its measure of diversity is based on Mahalanobis Distance and utilises the core concept of inheritance and genetic algorithm. The underlying idea is to create unique synthetic minority instances using two relatively distant parent instances which are different to their parents (i.e. existing minority class).

SMOM is proposed as a k-NN-based synthetic minority oversampling algorithm [11]. Its advantage over traditional k-NN based oversampling algorithms is that it minimises the minority data region from being over-generalised by considering both minority and majority data space density. This is achieved by computing selection weights to quantify the adverse impacts for all other classes if synthetic instances are generated along a particular neighbourhood direction. Low weights are assigned to neighbourhood directions which will result in over-generalisation. The key steps involve the usage of neighbourhood-based clustering (NBDOS) to identify outstanding and trapped instances, computation of selection weights for trapped instances (outstanding instances have equal weights in all directions), and generation of synthetic instances based on selection weights.

In 2018, Sampling With the Majority (SWIM) was proposed [8]. Synthetic minority instances are generated based on the distribution of majority class instances which are effective against extremely imbalanced data. In 2021, a diversity-based sampling method with a drop-in functionality was proposed to evaluate diversity. It was achieved via a greedy algorithm that is used to identify and discard subsets that share the most similarity [12].

KMEANS-SMOTE [13] is a data-level oversampling method that was introduced in 2018 which combines k-means clustering algorithm with SMOTE. It seeks to address the shortcomings of SMOTE by aiming for safe areas which would benefit from the generation of synthetic instances. It achieves this by oversampling safe regions within the decision boundary. The author also commented that the attractiveness of the proposed K-MEANS SMOTE comes from the universal availability and proven effectiveness of both underlying algorithms.

In 2020, Minority Clustering SMOTE (MC-SMOTE) [14] is then introduced. It aims to soften the occurrence of sample-intensive and sample-sparse regions after the synthetic data generation process. It incorporates an element of K-means algorithm at minority datapoints. The algorithm aims to populate synthetic instances between clusters. The authors experimented MC-SMOTE to determine wind-turbine fault and concluded that MC-SMOTE outperformed SMOTE.

In recent years, the challenges of imbalanced data classification are also reflected in imaging data. Generative Adversarial Neutral Networks (GANs) have attracted much focus from researchers due to their ability to model complex datasets across many different fields. A recent survey[15] was conducted by Sampath, et al., where it categorises existing GANs based techniques into 3 main groups (image level, object level, and pixel level imbalances). This study also enables readers to gain an understanding of how GANs are used to address the issue of imbalanced datasets.

Most recently, Diversity-based Average Distance Oversampling (DADO) and Diversity-based Instance-Wise Oversampling (DIWO) have been proposed to promote diversity [16]. The objective of the two techniques is to generate well-diverse synthetic instances close to minority class instances. DADO aims to ensure diversity in the region among minority class instances, when minority instances are compact, and performs better when the immediate surrounding area is located within the minority space. In the case of DIWO, the contrasting approach is taken to ensure synthetic instances are clustered as closely to the actual minority class instances when minority instances are widely distributed, and the surrounding area does not sit within the minority space.

In our recent paper, we proposed a synthetic sampling method, namely Density-based Clustering Diversity Oversampling (DBCDO) [7]. Our proposed method combined the advantages of both DADO and DIWO by analysing density distribution of the minority instances using DBSCAN, a density-based clustering approach and maximising diversity optimisation.

In this paper, we aim to expand our proposed clustering algorithm with an alternative; neighbourhood-based clustering algorithm (NBDOS). The workings of NBDOS focuses on both minority and majority density, instead of only minority density. It is focused on the identification of clusters of outstanding instances, which allows us to classify all remaining minority instances outside of these outstanding clusters as trapped instances. Once these instances are promptly identified, DADO is applied onto clusters of outstanding instances and DIWO is applied onto trapped instances.

Methodology

Cluster-Based Diversity Over-Sampling (CDO)

In this section, we describe our synthetic data generation method, Cluster-based Diversity Over-sampling (CDO). CDO aims to enhance and improve robustness of synthetic data generation by integrating and leveraging the advantages of both DADO and DIWO. This is predicated on the outcome of the density distribution of the minority instances, where DADO is applied onto narrow clusters and DIWO is applied to disperse clusters. We will start with providing a brief overview of our proposed implementation of CDO using DBSCAN (identified as DBCDO), followed by NBDOS (identified as NBCDO).

DBSCAN was originally chosen as our preferred clustering method as it is more efficient when the problem involved identifying arbitrary shaped clusters in comparison to partition-based or hierarchical-based clustering methods [17]. DBSCAN was first introduced in 1996 [17] and is a non-parametric density-based clustering algorithm. It simultaneously performs two functions, first by strengthening the grouping of instances which are within proximity to each other, and secondly by identifying points which are placed in low-density areas (points whose nearest neighbours are relatively far away). This implies that DBSCAN is robust against outlier detection. Another notable advantage of DBSCAN is its ability to allow for selecting desired levels of similarity through hyper-parameter selection.

Neighbourhood-based clustering (or NBDOS) which discovers the clusters of outstanding instances is introduced as part of SWOM [11]. It aims to distinguish outstanding instances (minority instances which are clustered closely) and trapped instances (minority instances which are dispersed, isolated and sometimes located within other majority regions). In our situation, the advantage of NBDOS lies in the fact that it is conducted on the entire data on both minority and majority data space to uncover the minority instances lying in dense clusters or spread dispersedly. This differs from DBSCAN, where the algorithm is applied solely on minority data to detect instances above a certain density threshold. This in turn causes NBDOS to be more sensitive towards the hyper-parameters selection. This will consequently impact the identification of "soft core", "outstanding" and "trapped" instances, especially in the use case on a small number of minority instances. In extremely imbalanced data space, this can result in situations where all instances are classified as "trapped".

The algorithm of CDO is shown in Algorithm 1. Clustering algorithm is applied in Step 2. The choice of clustering algorithms (DBSCAN and NBDOS) are denoted in Algorithm 2 & 3 respectively. It is worthwhile to observe that DBSCAN only requires minority instances as input, whereas NBDOS requires the entire data space (Algorithm 3, line 3 & 4). After clusters are obtained, CDO applies the following synthetic sampling process: if minority instances do not belong to any cluster, then apply DIWO; if minority instances belong to a cluster, then all the instances within the cluster perform DADO (Algorithm 1, lines 9 and 12). The algorithm of DBSCAN and NBDOS is shown in Algorithm 2 & 3, respectively.

Diversity Optimisation

The proposed algorithm for diversity optimisation and generation of synthetic instances is the extended form of NOAH's algorithm [18], as shown in Algorithm 4. Algorithm 4 contains 3 stages and requires the following input parameters: population size (n), number of generations to optimise objective function (g), number of instances remaining in the population after bound adaptation (r), percentage improvement of bound (v) and finally, the stopping criterion diversity maximisation (c). The above implies that if the population diversity does not improve for *c* generations, convergence of the diversity maximisation is achieved. The whole algorithm terminates if the bound does not improve for *c* generation. To further optimise the objective function, Algorithm 4 has incorporated the usage of Genetic Algorithm (GA), as it is the most popular evolutionary algorithm. Mutation and crossover concepts are utilised to create new instances. Instances which objective functions are better than bound value (b) are kept (Algorithm 4, lines 5 and 14). For DADO, the objective function (f) is the average of distance from all instances in the minority class. For DIWO, the objective function (f) is the distance to each instance.

We also made the following update to the DADO and DIWO algorithm, with the aim to further promote diversity within synthetic data. For DADO, the population size was initially set to oversampling size + 50 this has since been updated to oversampling size + 1. The intuition behind this modification is that lower population size encourages a more diverse synthetic sample generation process. The DIWO boundary was initially set to the minimum and maximum of the "isolated" minority instances data space, it has since been updated with the minimum and maximum of the entire minority instances data space. The broader generation region promotes more diverse synthetic samples.

Diversity-Based Selection

The preferred measure of diversity is Solow-Polasky measure. There are 3 main properties which are required of a diversity measure, which are: (1) monotonicity in variety; (2) monotonicity in distance; (3) twinning. The first property implies that the diversity measure will increase or at least be non-decreasing when an individual element currently not present in the dataset is added. The second property requires that the diversity between a particular set *S* (i.e. instances) should not be smaller than another set *S'* if all pairs in S are as remote as all pairs in *S'*. The third property ensures the diversity measure remains the same when an additional element, already in the set, is added. Solow-Polasky measure can be expressed in the following Eq. (1), where *M* represents the distance matrix. The Euclidean distance between elements of set *S* is denoted as $d(s_i, s_j)$. Thereafter, our diversity measure is derived and computed by the summation of all inverse matrix of ($M^{-1} = \left[m_{ij}\right]^{-1}$).

$$D(S) = \sum M^{-1} = \sum_{i} \sum_{j} e^{-d(s_i, s_j)} \tag{1}$$

To obtain the best diversity amongst all the instances, the ideal scenario would be to generate all possible permutation of subsets. However, this cannot be achieved as it would be computationally infeasible and expensive. As an alternative methodology, we propose the use of a greedy approach which would filter out instances which have the least contribution to the diversity of our dataset. Our definition of contribution is defined as the difference in diversity for our dataset with and without the instance. As proven in this study [18], the difference can be expressed in the following formula:

$$\sum M^{-1} - \sum A^{-1} = \frac{1}{\bar{c}}(\sum \bar{b} + \bar{c}) \tag{2}$$

where A is the distance matrix of the set without that particular instance, $M = \begin{bmatrix} A & b \\ b^T & c \end{bmatrix}$, $M^{-1} = \begin{bmatrix} \bar{A} & \bar{b} \\ \bar{b}^T & \bar{c} \end{bmatrix}$, c and $\bar{c}$ are single elements, b and $\bar{b}$ are vectors and b^T and $\bar{b}^T$ are their transpose.

Algorithm 1: Cluster-based diversity over-sampling algorithm (CDO)

```
/* Step 1: Clustering minority instances */
for each point $M$ in minority class do:
apply clustering algorithm
if M belongs to a cluster:
label($M$) = $C$
if M does not belong to any cluster:
label($M$) = isolated
end for
/* Step 2: Perform diversity algorithm for each cluster and isolated instances*/
for each cluster $C$ do:
$P_{DADO}^{C}$ ← NOAH(n, g, r, c, v, f, m)
end for
for each minority instances $M$ that is isolated do:
$P_{DIWO}^{M}$ ← $NOAH(n, g, r, c, v, f, m)$
end for
/* Step 3: Combine generated datasets */
$P = P_{DADO}^{C}\ U\ P_{DIWO}^{M}$
```

Algorithm 2: DBSCAN

```
/* input */
$p:$ binary parameter if border point is assigned in clusters; $eps$ Epsilon
/* output*/
$M_{label}$: labelled minority instances
initialise the cluster labels of minority class to 0, $C = 0$
for each instance $M$ in minority class do:
  if $M$ is labelled then next
  if $M$ is not labelled then
    NeighborPts ← return all points within $eps$ neighbourhood of $M$ (incl. $M$)
  if size of NeighborPts = 1 then $M_{label}$ ← isolated next
     $C = C + 1$
     $M_{label}$ ← $C$
     for each $M'$ in NeighborPts do:
      if $M'_{label}$ = isolated and $p$ = $True$ then $M'_{label}$ = $C$ next
      if $M'$ is labelled: next
        $M'_{label}$ ← $C$
      neighborPts ' ← return all points within $eps$ neighbourhood of $M'$ (incl. $M'$)
      if size(neighborPts ') > 1 then NeighborPts ← NeighborPts U NeighborPts'
end for
end for
```

Algorithm 3: NBDOS

```
/* input */
$k1$: Number of nearest neighbours used to generate the synthetic instances; $k2$: number of nearest neighbours used to determine soft core instances; $nTh$: the minimum number of points required per cluster; $rTh$: the minimal proportion of minority instances which should be achieved for the soft core instances in their k-nearest neighbours

/* output */
$M_{label}$: labelled minority instances

/* Identify nearest neighbours */
for each instance $M$ in minority class:
find the nearest $k3$ instances from $M.N_{k3}$, in all minority class region, where $N$ is the minority class region; $k3 = \max(k1, k2)$; record the distance between $M$ and its $k1$th nearest instance as $M.r_{k1}$. and obtain the nearest $k1$ instance from $M.N_{k1}$
find the nearest $k_3$ instances from $M.P_{k3}$, in all majority class region, where $P$ is the majority class region; record instances $T$ which distances of they to $M$ are no larger than $M.r_{k1}$, then construct $M.Fs \leftarrow T \cup M.N_{k1}$
  find the $k_2$-nearest neighbours set of $M$, $M.N_{k2}$, in the union of $M.N_{k3}$ and $M.P_{k3}$
/* Cluster */
initialise the soft core instances set of minority class to be empty, $sfS \leftarrow \emptyset$
initialise the cluster labels of minority class to 0, $C \leftarrow 0$
for each minority instances $M$ do:
  $Tem \leftarrow$ all minority class instances in $x.N_{k2}$
  if $round\left(\frac{|Tem|}{k_2}\right) \geq rTh$ then
    $sfS \leftarrow sfS \cup \{M\}$
    $H_{k2}(M) \leftarrow Tem$
  end if
end for
for all $M$ in $sfS$ do
    $H_{k2}(M) \leftarrow H_{k2}(M) \cup (Tem \cap sfS)$
end for
for each minority instance $M$ do:
  if $M$ is labelled then next
  if $M$ is not labelled then
    $C = C + 1$
    $M_{label} \leftarrow expendCluster(sfS, M, C, M_{label})$
  end if
end for
for each cluster $C$ do
  if size of cluster $C < nTh$ then $M_{label} \leftarrow 0$
  end if
end for

function $expendCluster(sfS, M, C, M_{label})$
  $sfC \leftarrow \{M\}$
  $M_{label} \leftarrow C$
  while $sfC \neq \emptyset$ do
    for all $M^i \in H_{k2}(M^j)$ do
      if $M^i_{label} = 0$ then
        $M^i_{label} \leftarrow C$
        if $M^i \in sfS$ then $sfC \leftarrow sfC \cup \{M^i\}$
        end if
    end for

    $sfC \leftarrow sfC \setminus \{M^j\}$
  end while
  return $M_{label}$

end function
```

Algorithm 4: Diversity optimisation algorithm (NOAH)

```
\* Input *\
$n$: population size; $g$: number of generations to optimise objective fuction; $r$: number of instances remaining in the population after bound adaption; $v$: percentage improvement of bound; $c$: the stopping criterion diversity maximisation; $m$: binary variable of DIWO and DADO
\* Output *\
a diverse set of instances $S$
initialise $S \leftarrow \emptyset$; b $\leftarrow \infty$; $i = 0$
while $i < c$ do
/* Step 1: Optimising the objective function */
    $P \leftarrow$ Generate a population with $n$ instances
    for $g$ generations do
        $P' \leftarrow$ generate new $n$ instances via mutation and crossover from $P$ with objective values better than $b$
        $P \leftarrow$ Select $n$ best instances from $P \cup P'$
    end for
/* Step 2: Bound adaptation */
    $P \leftarrow$ select $r$ best instances from $P \cup S$
    $b' \leftarrow$ put the objective value of rth best instance in $P \cup S$
    if $m = DIWO$ and $b - b' < v \times b$ then $i \leftarrow i + 1$ else $i \leftarrow 0$
    $b \leftarrow b'$
/* Step 3: Diversity maximisation */
        $P'' \leftarrow$ generate new $r$ instances via mutation and crossover from $P$ with objective values better than $b$
        $P^{\therefore} \leftarrow$ select r best diverse instances from $P'' \cup S$
    end while
    if diversity of $P^{\therefore}$ is more than S then $S \leftarrow P^{\therefore}$ else $j \leftarrow j + 1$
end while
```

Validation of Synthetic Dataset

Evaluation Method

The learning classifiers used to evaluate the generated data are, Naïve Bayes (NB), Decision Tree (DT), k-Nearest Neighbour (KNN), Support Vector Machine (SVM), and Random Forest (RF). We chose KNN and RF as they are sensitive to imbalanced data based on their model assumptions [19]. DT is selected based on development of decision regions which are influenced by re-sampling methods [20]. SVM with radial kernel is effective to classify classes which are not separable linearly.

We measure the performance of the classifiers on test data using AUC, F1-score, G-means, and PR-AUC as classification accuracy is not an appropriate measure for imbalanced data.

To calculate F1-score (5), we need to measure recall and precision shown as (3) and (4). Recall is the proportion of correctly predicted positive instances to all instances in the positive class. Precision is the proportion of correctly predicted positive instances to all predicted positive instances.

$$\mathrm{R}ecall = \frac{TP}{TP + FN} \tag{3}$$

$$\mathrm{P}recision = \frac{TP}{TP + FP} \tag{4}$$

$$\mathrm{F1} = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{5}$$

The Receiver Operating Characteristic (ROC) curve is a technique to summarise the performance of a classifier over trade-offs between recall and False Positive Rate (FPR) as (6).

Table 1 Synthetic datasets characteristics

Dataset	Number of clusters	Variance of minority data space	Data points	Imbalance ratio (%)
DS1 (Example 3)	2	Medium	315	10
DS2 (Example 4)	2	High	315	10
DS3 (Example 6)	1	Low	300	5
DS4 (Example 7)	1	Medium	300	5

Table 2 Performance results of mean and standard error for each measure across synthetic datasets

DS1			DS3		
	NBCDO	DBCDO		NBCDO	DBCDO
AUC	**0.9665 ± 0.027**	0.9600 ± 0.031	AUC	**0.9986 ± 0.003**	0.9982 ± 0.003
F1	0.7099 ± 0.103	**0.7168 ± 0.122**	F1	**0.8960 ± 0.126**	0.8799 ± 0.123
G-means	**0.9394 ± 0.036**	0.9367 ± 0.042	G-means	**0.9620 ± 0.179**	0.9606 ± 0.178
PR-AUC	**0.9887 ± 0.016**	0.9883 ± 0.017	PR-AUC	0.9981 ± 0.008	0.9981 ± 0.008
DS2			DS4		
	NBCDO	DBCDO		NBCDO	DBCDO
AUC	0.8764 ± 0.050	**0.8859 ± 0.059**	AUC	**0.9869 ± 0.012**	0.9867 ± 0.011
F1	0.4429 ± 0.118	**0.4457 ± 0.128**	F1	0.7590 ± 0.167	**0.7646 ± 0.173**
G-means	**0.8423 ± 0.073**	0.8389 ± 0.056	G-means	**0.9535 ± 0.174**	0.9526 ± 0.174
PR-AUC	0.9841 ± 0.013	0.9841 ± 0.013	PR-AUC	**0.9978 ± 0.008**	0.9979 ± 0.008

Bold numbers indicate the mean of method performance is the best among all comparable methods

$$FPR = \frac{FP}{FP + TN} \tag{6}$$

where FP stands for False Positive that is the number of instances from positive class predicted incorrectly.

AUC, the area under the ROC curve, is a suitable measure for classifiers' performance, especially in the situation of imbalanced data, and is independent of the decision boundary [5, 21]. PR-AUC denotes the area under the Precision Recall curve.

The G-means (7) is the geometric mean of True Positive Rate (TPR), which is the same as Recall in (3) and true negative rate (TNR), which is $1 - FPR$.

$$G - means = \sqrt{TPR \times TNR} \tag{7}$$

Synthetic Dataset

To examine our proposed methods under different scenario, 4 2-dimensional datasets are created. Each of these four datasets are eventually split into half, with Imbalance Ratio (IR) of 10% and 5%, respectively. These datasets are used in our initial experiments to assist in hyper-parameters selection. Table 1 provides a summary of these datasets (DS1-4). There is a varying amount of cluster within each dataset, ranging from 0 (randomly distributed data points) in DS3 to 5 in DS1. For each of the four synthetic datasets, instances are randomly divided into training and test datasets with a 75:25 split. DBCDO and NBCDO are utilised to balance our training datasets. Learning classifiers are applied onto the balanced training datasets. Performance of these constructed learning classifiers is then assessed using the test datasets. Performance measures (AUC, F1, G-Means, and PR-AUC)

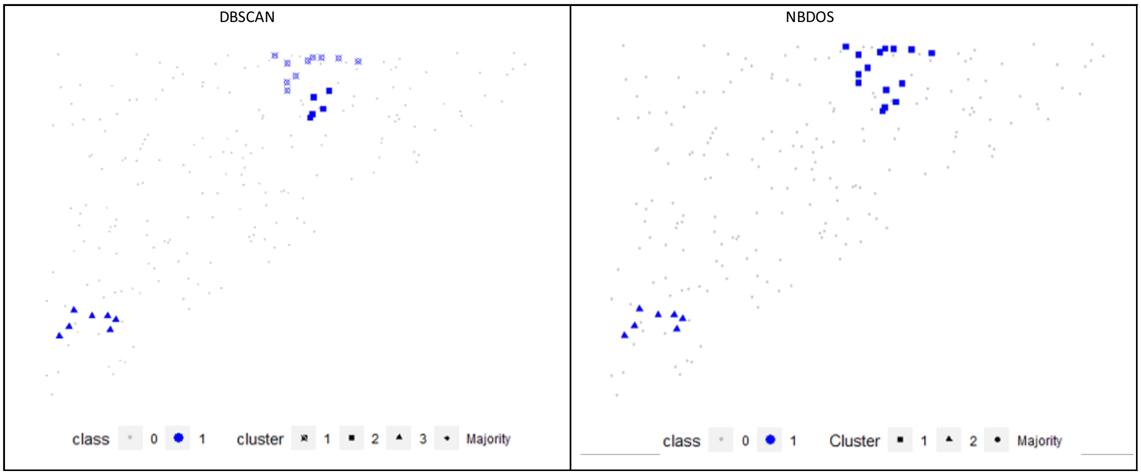

Fig. 1 Plots for clustering methods on minority data points

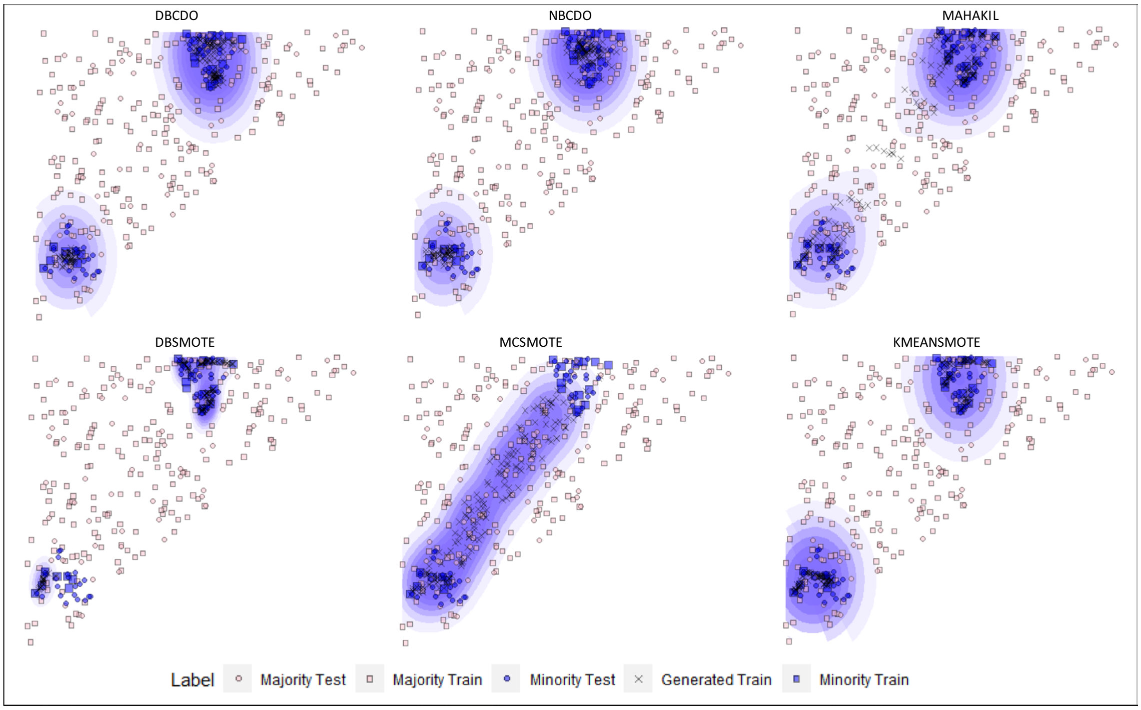

Fig. 2 Plots for synthetic datasets generation region

are computed for the best performing classifier. The above process is repeated 30 times.

Parameter Selection

The distance measures chosen for both objective function and diversity measure are the optimal distance measure based on experimental results [16]. Euclidean distance measure (D_{Eu}) is chosen for DADO, and Canberra (D_c) is chosen for DIWO.

$$D_{Eu}(x, y) = \sqrt{\sum_i (x_i - y_i)^2} \tag{8}$$

$$D_c(x, y) = \sum_i \frac{|x_i - y_i|}{|x_i| + |y_i|} \tag{9}$$

Our next step is to determine the optimal hyper-parameters for DBSCAN and NDBOS. Based on previous work ([7], the optimal parameter configuration for Epsilon (eps) is 0.05 and Border Point (p) set as “T” / “True”. For NBDOS, there is a total of 4 hyper-parameters which require configuration. *nTh* (“minimum points per cluster”) is set as 5 given our goal is to target extremely imbalanced datasets. *rTh* is set as 0.5 to relax the stringent condition of selecting soft care instances. For *k1* and *k2*, extensive study on synthetic datasets has been conducted on a series of combinations where $k1 \in (4, 6, 8)$ and $k2 \in (5, 6, 7, 8)$. Results from our study of synthetic datasets suggest that the optimal value for *k1* and *k2* is 8 and 6, respectively. Nevertheless, it is important to call out that these parameter selections are sensitive to the distribution of majority and minority instances in varying datasets, which could impact its capability.

Synthetic Experiment Results

Comparison between DBCDO and NBCDO was conducted using 4 synthetic datasets and the performance of each of

Table 3 Real-word data description

Dataset	Name	Dim	Size	Dataset	Name	Dim	Size
D1	Wisconsin	9	683	D6	Glass (0, 1, 2, 3 vs 4, 5, 6)	9	214
D2	Diabetes	8	768	D7	Haberman	3	306
D3	Ecoli (0,1 vs 5)	6	240	D8	New Thyroid	5	215
D4	Ecoli 2	7	336	D9	Pima	8	768
D5	Ecoli 3	7	336	D10	Wine Red Low vs High	11	280

Table 4 Performance results of mean and standard error across datasets with 5% imbalance levels

Dataset	NBCDO	DBCDO	DBSMOTE	MAHAKIL	MCSMOTE	KMEANSMOTE
F1						
D1	0.9520 ± 0.022	**0.9526 ± 0.022**	0.9432 ± 0.024	0.9489 ± 0.023	0.9524 ± 0.022	0.9471 ± 0.023
D2	0.5558 ± 0.081	0.5635 ± 0.074	0.5446 ± 0.111	**0.6307 ± 0.040**	0.6281 ± 0.049	0.4900 ± 0.127
D3	0.8356 ± 0.098	0.8260 ± 0.102	**0.8460 ± 0.088**	0.8177 ± 0.095	0.8051 ± 0.119	0.8114 ± 0.112
D4	0.8426 ± 0.095	0.8619 ± 0.096	0.8406 ± 0.080	0.8289 ± 0.110	0.8402 ± 0.091	**0.8761 ± 0.062**
D5	0.6527 ± 0.087	0.6647 ± 0.094	**0.6913 ± 0.079**	0.6531 ± 0.101	0.6708 ± 0.086	0.6757 ± 0.091
D6	0.8468 ± 0.082	**0.8616 ± 0.058**	0.8238 ± 0.058	0.8499 ± 0.065	0.8505 ± 0.063	0.8126 ± 0.067
D7	**0.4684 ± 0.115**	0.4602 ± 0.106	0.3183 ± 0.155	0.4390 ± 0.133	0.4373 ± 0.126	0.2482 ± 0.114
D8	**0.9742 ± 0.042**	0.9718 ± 0.035	0.9653 ± 0.034	0.9630 ± 0.037	0.9689 ± 0.037	0.9661 ± 0.040
D9	0.5650 ± 0.073	0.5625 ± 0.069	0.5423 ± 0.114	**0.6307 ± 0.040**	0.6281 ± 0.049	0.4900 ± 0.127
D10	0.7005 ± 0.094	**0.7179 ± 0.104**	0.6924 ± 0.148	0.6974 ± 0.082	0.6940 ± 0.122	0.6688 ± 0.147
G-Means						
D1	0.9658 ± 0.017	**0.9680 ± 0.015**	0.9605 ± 0.020	0.9664 ± 0.016	0.9671 ± 0.016	0.9632 ± 0.017
D2	0.6457 ± 0.065	0.6521 ± 0.061	0.6319 ± 0.090	**0.7087 ± 0.031**	0.7058 ± 0.040	0.5882 ± 0.109
D3	**0.9095 ± 0.082**	0.9004 ± 0.093	0.9006 ± 0.086	0.9090 ± 0.068	0.9094 ± 0.096	0.9088 ± 0.093
D4	0.9220 ± 0.041	**0.9350 ± 0.036**	0.9090 ± 0.051	0.9198 ± 0.044	0.9206 ± 0.047	0.9331 ± 0.037
D5	0.8908 ± 0.046	0.8999 ± 0.046	0.9095 ± 0.052	0.8977 ± 0.041	0.9112 ± 0.040	**0.9143 ± 0.037**
D6	0.8909 ± 0.079	**0.9039 ± 0.059**	0.8636 ± 0.062	0.8943 ± 0.067	0.8998 ± 0.074	0.8576 ± 0.069
D7	0.6130 ± 0.084	0.6058 ± 0.076	0.4751 ± 0.151	0.6080 ± 0.104	**0.6177 ± 0.085**	0.4065 ± 0.114
D8	**0.9909 ± 0.019**	0.9888 ± 0.021	0.9779 ± 0.031	0.9848 ± 0.021	0.9867 ± 0.023	0.9866 ± 0.025
D9	0.6534 ± 0.059	0.6511 ± 0.056	0.6301 ± 0.092	**0.7087 ± 0.031**	0.7058 ± 0.040	0.5882 ± 0.109
D10	0.7854 ± 0.084	**0.7924 ± 0.090**	0.7644 ± 0.121	0.7726 ± 0.087	0.7725 ± 0.100	0.7455 ± 0.120
PR-AUC						
D1	0.9821 ± 0.056	0.9838 ± 0.056	0.9796 ± 0.061	0.9835 ± 0.055	**0.9840 ± 0.056**	0.9816 ± 0.062
D2	0.8278 ± 0.065	0.8316 ± 0.062	**0.8441 ± 0.064**	0.8151 ± 0.066	0.8272 ± 0.063	0.8073 ± 0.066
D3	**0.9837 ± 0.030**	0.9832 ± 0.031	0.9823 ± 0.030	0.9808 ± 0.034	0.9835 ± 0.028	0.9827 ± 0.030
D4	**0.9807 ± 0.037**	0.9806 ± 0.038	0.9777 ± 0.038	0.9798 ± 0.038	0.9806 ± 0.036	0.9797 ± 0.037
D5	0.9934 ± 0.004	**0.9935 ± 0.004**	0.9930 ± 0.005	0.9925 ± 0.006	0.9930 ± 0.005	0.9927 ± 0.005
D6	0.9788 ± 0.055	0.9796 ± 0.052	**0.9830 ± 0.030**	0.9775 ± 0.056	0.9767 ± 0.062	0.9822 ± 0.041
D7	**0.8375 ± 0.056**	0.8331 ± 0.060	0.8187 ± 0.062	0.8198 ± 0.068	0.8259 ± 0.059	0.8145 ± 0.056
D8	**0.9991 ± 0.001**	0.9990 ± 0.001	0.9988 ± 0.002	0.9989 ± 0.001	**0.9991 ± 0.001**	0.9989 ± 0.002
D9	0.8290 ± 0.063	0.8290 ± 0.065	**0.845 ± 0.064**	0.8151 ± 0.066	0.8272 ± 0.063	0.8073 ± 0.066
D10	0.9658 ± 0.022	0.9721 ± 0.011	0.9641 ± 0.022	**0.9726 ± 0.012**	0.9719 ± 0.012	0.9633 ± 0.017
AUC						
D1	**0.9916 ± 0.006**	0.9904 ± 0.007	0.9892 ± 0.007	0.9887 ± 0.008	0.9893 ± 0.007	0.9904 ± 0.007
D2	0.7793 ± 0.037	0.7743 ± 0.037	0.7920 ± 0.038	0.7956 ± 0.031	**0.7978 ± 0.036**	0.7915 ± 0.043
D3	0.9913 ± 0.008	0.9900 ± 0.008	0.9915 ± 0.008	**0.9917 ± 0.008**	0.9908 ± 0.007	0.9915 ± 0.007
D4	0.9640 ± 0.022	0.9654 ± 0.023	0.9647 ± 0.024	0.9627 ± 0.022	0.9669 ± 0.020	**0.9684 ± 0.024**
D5	**0.9397 ± 0.033**	0.9370 ± 0.034	0.9355 ± 0.034	0.9348 ± 0.037	0.9377 ± 0.035	0.9329 ± 0.039
D6	0.9757 ± 0.022	0.9750 ± 0.023	**0.9790 ± 0.020**	0.9773 ± 0.019	0.9775 ± 0.021	0.9780 ± 0.021
D7	0.6993 ± 0.072	**0.7020 ± 0.069**	0.6537 ± 0.079	0.6741 ± 0.084	0.6861 ± 0.086	0.6733 ± 0.075
D8	0.9983 ± 0.004	**0.9984 ± 0.004**	0.9980 ± 0.004	0.9978 ± 0.005	**0.9984 ± 0.004**	0.9982 ± 0.004
D9	0.7772 ± 0.038	0.7767 ± 0.036	0.7937 ± 0.037	0.7958 ± 0.031	**0.7978 ± 0.036**	0.7915 ± 0.043
D10	0.9334 ± 0.027	0.9273 ± 0.031	**0.9413 ± 0.03**	0.9298 ± 0.032	0.9286 ± 0.022	0.9363 ± 0.028

Bold numbers indicate the mean of method performance is the best among all comparable methods the algorithms are evaluated. In total, there are 16 different results, and they are reflected in Table 2. NBCDO performs better than DBCDO in 10 out of 16 cases across evaluation metrics. This is especially so for DS1, 3, 4 where there is low to medium variance in minority space. NBCDO does not outperform DBCDO in DS2 where there is high variance in minority space. The likely explanation for this is when the minority data space has high variance, it is more likely to be surrounded by majority data points. This may have caused NBCDO to cluster and categorise the entire data space (which also houses majority class instances) as isolated instances, thereby impacting the process of learning.

Graphical Representation

To provide graphical representations, we created a synthetic dataset with two clusters, 10% imbalanced ratio in training data with a balanced ratio in testing data. A comparison between the two clustering algorithm, DBCAN and NBDOS is demonstrated in Fig. 1. DBSCAN identified 3 clusters compared to NBDOS, which accurately identified 2 clusters. This can be attributed to the additional information about majority distribution utilised during the clustering process. NBDOS will end up with a more representative data generation region once DADO is applied due to the more accurate identification of actual clusters by NBDOS when compared to DBSCAN method.

In Fig. 2, synthetic datasets are generated by DBCDO and NBCDO (two cluster-based diversity methods) and its 3 comparable methods (DB-SMOTE, MAHAKIL, KMEANS-SMOTE and MC-SMOTE). We observe that the regions of synthetic generated instances for cluster-based diversity algorithms, MAHAKIL and KMEANS-SMTOE are relatively similar. However, cluster-based diversity algorithms stand out for its ability to cover all the data points of the minority test data with the narrowest region. MAHAKIL created synthetic data points between the two clusters, which occupies a larger region and could result in over-generalisation and higher False Positive Rate. The region for DB-SMOTE does not cover all of minority test data points, which could result in higher false negative rate. The region for MC-SMOTE also has many synthetic data generated outside of the clusters, which is how this algorithm works.

Validation of Real-Life Dataset

We validate the proposed CDO algorithm against an assortment of 10 imbalanced datasets, with varying dimensions. The datasets and their characteristics are described in Table 3, and "Ratio" is used to indicate the original proportion of majority to minority instances. To replicate the scenarios with low and extremely low imbalanced ratio, we reduce the imbalanced ratio to 5% and 10 absolute count of minority instances.

The data within each of the real-world datasets are randomly divided into train and test datasets using a 75:25 split, respectively. This process is repeated for 30 iterations, resulting in 30 unique variations of training datasets and accompanying experimental datasets for each of the 10 real-world datasets. After the initialisation step, we apply our proposed methods (NBCDO and CDO) alongside with existing methods in the literature, namely DB-SMOTE, MAHAKIL, MC-SMOTE and KMEANS-SMOTE to evaluate algorithm performance. Six learning classifiers (GLM, NB. DT, KNN, SVM, NN) are then constructed on each of the training datasets ($n = 30$). Subsequently, the trained classifiers are applied onto test datasets.

For each real-world datasets, the best performing classifier is selected before computing the mean and standard error of the performance measures as F1, AUC, PR-AUC and G-mean. Additionally, we examine the statistical significance of differences for the performance measures obtained from all comparable methods using a non-parametric statistical test, Mann–Whitney test.

Experimental Results

The mean and standard error (stated in parenthesis) of our proposed method (DBCDO, NBCDO) and its comparable methods (DB-SMOTE, MAHAKIL, KMEANS-SMOTE and MC-SMOTE) are presented in Table 4 (5% imbalanced ratio) and Table 5 (10 minority instances).

By looking at the performance metrics for 5% imbalanced ratio (Table 4), across all evaluation metric and datasets, both DBCDO and NBCDO have the highest mean 10 times, followed by DB-SMOTE 7 times, MAHAKIL and MC-SMOTE 6 times each, and KMEANS-SMOTE 3 times. In total, cluster-based diversity algorithm outperformed its comparable algorithms 20 out of 40 times.

By looking at the performance metrics for 10 minority instances (Table 5), across all evaluation metric and datasets, DBCDO has the highest mean 10 times, followed by NBCDO 9 times, MAHAKIL 7 times, DB-SMOTE and MC-SMOTE 6 times, and KMEANS-SMOTE 3 times. In total, cluster-based diversity algorithm outperformed its comparable algorithms 19 out of 40 times.

The Mann–Whitney test is performed for each pairing of all 6 comparable algorithms. Tables 6 and 7 display the results from the test, where each figure represents the frequency that the specified method is statistically better than its comparable methods.

Table 6 reports results on 5% imbalanced datasets. MC-SMOTE statistically outperforms its comparable algorithms 48 times across all datasets and evaluation metrics, followed

Table 5 Performance results of mean and standard error across datasets with 10 minority instances

Datase	NBCDO	DBCDO	DBSMOTE	MAHAKIL	MCSMOTE	KMEANSMOTE
F1						
D1	0.9448 ± 0.022	**0.9479 ± 0.021**	0.9316 ± 0.033	0.9433 ± 0.022	0.9466 ± 0.020	0.9450 ± 0.022
D2	0.5398 ± 0.082	0.5456 ± 0.082	0.4049 ± 0.141	**0.5832 ± 0.069**	0.5811 ± 0.057	0.4348 ± 0.140
D3	0.8344 ± 0.101	0.8260 ± 0.102	**0.8460 ± 0.088**	0.8177 ± 0.095	0.8051 ± 0.119	0.8114 ± 0.112
D4	0.8418 ± 0.102	0.8468 ± 0.109	0.8358 ± 0.080	0.8308 ± 0.098	0.8296 ± 0.115	**0.8481 ± 0.111**
D5	0.6525 ± 0.090	0.6611 ± 0.087	**0.6871 ± 0.092**	0.6497 ± 0.095	0.6611 ± 0.083	0.6623 ± 0.093
D6	0.8579 ± 0.079	0.8743 ± 0.057	0.8369 ± 0.074	**0.8745 ± 0.049**	0.8701 ± 0.055	0.8460 ± 0.063
D7	**0.4689 ± 0.094**	0.4605 ± 0.110	0.3668 ± 0.144	0.4362 ± 0.115	0.4552 ± 0.098	0.3082 ± 0.127
D8	0.9694 ± 0.048	**0.9831 ± 0.028**	0.9482 ± 0.061	0.9785 ± 0.031	0.9752 ± 0.033	0.9674 ± 0.039
D9	0.5343 ± 0.079	0.5510 ± 0.081	0.4047 ± 0.141	**0.5832 ± 0.069**	0.5811 ± 0.057	0.4348 ± 0.140
D10	**0.7198 ± 0.048**	0.7144 ± 0.084	0.6772 ± 0.141	0.6800 ± 0.072	0.7096 ± 0.074	0.6672 ± 0.114
G-Means						
D1	0.9591 ± 0.020	**0.9635 ± 0.016**	0.9484 ± 0.032	0.9610 ± 0.018	0.9623 ± 0.016	0.9619 ± 0.017
D2	0.6342 ± 0.062	0.6371 ± 0.067	0.5113 ± 0.130	**0.6746 ± 0.050**	0.6679 ± 0.048	0.5522 ± 0.104
D3	0.8997 ± 0.094	0.9007 ± 0.093	0.9006 ± 0.086	0.9090 ± 0.068	**0.9094 ± 0.096**	0.9088 ± 0.093
D4	0.9158 ± 0.045	**0.9309 ± 0.039**	0.9100 ± 0.048	0.9192 ± 0.039	0.9248 ± 0.040	0.9243 ± 0.054
D5	0.8917 ± 0.061	0.8984 ± 0.049	0.8919 ± 0.070	0.8977 ± 0.062	**0.9125 ± 0.035**	0.9113 ± 0.038
D6	0.8900 ± 0.071	**0.9136 ± 0.053**	0.8758 ± 0.066	0.9104 ± 0.044	0.9091 ± 0.054	0.8843 ± 0.064
D7	0.6137 ± 0.079	0.6036 ± 0.081	0.5201 ± 0.129	0.6087 ± 0.092	**0.6299 ± 0.079**	0.4445 ± 0.148
D8	0.9850 ± 0.032	**0.9961 ± 0.007**	0.9700 ± 0.054	0.9920 ± 0.017	0.9913 ± 0.018	0.9849 ± 0.024
D9	0.6298 ± 0.060	0.6415 ± 0.066	0.5111 ± 0.129	**0.6746 ± 0.050**	0.6679 ± 0.048	0.5522 ± 0.104
D10	**0.8053 ± 0.047**	0.7865 ± 0.082	0.7407 ± 0.125	0.7604 ± 0.085	0.7764 ± 0.080	0.7383 ± 0.115
PR-AUC						
D1	0.9885 ± 0.014	**0.9923 ± 0.007**	0.9858 ± 0.015	0.9902 ± 0.010	0.9896 ± 0.013	0.9890 ± 0.011
D2	**0.8183 ± 0.058**	0.8169 ± 0.059	0.7994 ± 0.071	0.8090 ± 0.063	0.8111 ± 0.063	0.8019 ± 0.065
D3	**0.9839 ± 0.031**	0.9831 ± 0.031	0.9823 ± 0.030	0.9808 ± 0.034	0.9835 ± 0.028	0.9827 ± 0.030
D4	0.9783 ± 0.037	0.9795 ± 0.039	0.9765 ± 0.038	0.9790 ± 0.038	**0.9796 ± 0.036**	**0.9796 ± 0.037**
D5	0.9932 ± 0.004	**0.9935 ± 0.004**	0.9920 ± 0.008	0.9923 ± 0.005	0.9928 ± 0.006	0.9926 ± 0.006
D6	**0.9574 ± 0.097**	0.9561 ± 0.099	0.9572 ± 0.087	0.9564 ± 0.095	0.9551 ± 0.101	0.9558 ± 0.096
D7	0.8391 ± 0.058	**0.8448 ± 0.055**	0.8323 ± 0.059	0.8336 ± 0.055	0.8285 ± 0.053	0.8185 ± 0.057
D8	0.9948 ± 0.025	0.9948 ± 0.025	**0.9952 ± 0.022**	0.9949 ± 0.026	0.9948 ± 0.026	0.9946 ± 0.026
D9	**0.8185 ± 0.058**	0.8165 ± 0.059	0.7994 ± 0.070	0.8090 ± 0.063	0.8111 ± 0.063	0.8019 ± 0.065
D10	0.9431 ± 0.065	0.9462 ± 0.065	0.9314 ± 0.077	**0.9479 ± 0.065**	0.9447 ± 0.066	0.9422 ± 0.063
AUC						
D1	**0.9907 ± 0.007**	0.9895 ± 0.009	0.9883 ± 0.009	0.9872 ± 0.009	0.9873 ± 0.008	0.9875 ± 0.009
D2	0.7645 ± 0.036	0.7637 ± 0.036	**0.7804 ± 0.034**	0.7634 ± 0.045	0.7674 ± 0.037	0.7648 ± 0.038
D3	0.9911 ± 0.008	0.9910 ± 0.008	0.9914 ± 0.008	**0.9912 ± 0.008**	0.9908 ± 0.007	0.9915 ± 0.007
D4	0.9594 ± 0.031	0.9620 ± 0.028	0.9616 ± 0.025	0.9592 ± 0.028	0.9625 ± 0.026	**0.9640 ± 0.028**
D5	**0.9404 ± 0.031**	0.9372 ± 0.034	0.9306 ± 0.046	0.9362 ± 0.034	0.9351 ± 0.038	0.9343 ± 0.038
D6	0.9758 ± 0.023	0.9804 ± 0.018	0.9793 ± 0.020	0.9787 ± 0.018	**0.9815 ± 0.016**	0.9798 ± 0.018
D7	**0.7096 ± 0.071**	0.7054 ± 0.070	0.6683 ± 0.072	0.6901 ± 0.073	0.6922 ± 0.078	0.6827 ± 0.075
D8	0.9989 ± 0.003	0.9991 ± 0.002	0.9985 ± 0.003	0.9991 ± 0.002	**0.9992 ± 0.002**	0.9987 ± 0.004
D9	0.7639 ± 0.035	0.7638 ± 0.036	**0.7812 ± 0.034**	0.7631 ± 0.045	0.7674 ± 0.037	0.7648 ± 0.038
D10	0.9533 ± 0.016	0.9459 ± 0.016	**0.9535 ± 0.016**	0.9514 ± 0.023	0.9449 ± 0.019	0.9463 ± 0.019

Bold numbers indicate the mean of method performance is the best among all comparable methods by NBCDO 40 times, DBCDO 39 times, MAHAKIL 32 times, and both DB-SMOTE 27 and KMEANS-SMOTE 25 times. Specifically, NBCDO has the best statistical performance across AUC and PR-AUC, whereas MC-SMOTE has the best statistical performance across F1 and G-means.

Table 7 reports results on datasets with 10 minority instances. DBCDO statistically outperforms its comparable algorithms 61 times across all datasets and evaluation metrics, followed by NBCDO 42 times, MC-SMOTE 41 times, MAHAKIL 33 times, DBSMOTE 17 times and KMEANS-SMOTE 14 times. Specifically, DBCDO has the best statistical performance across F1, G-means and PR-AUC, whereas NBCDO has the best statistical performance across AUC.

Discussions

As shown in the results for mean comparison (Tables 4 and 5), both cluster-based diversity methods (NBCDO and DBCDO) outperformed its comparable methods.

Cluster-based diversity methods outperformed DB-SMOTE as they considered the data space distribution and generated diverse instances within the boundaries of the identified data generation region. In contrast, DB-SMOTE method created synthetic instances using linear interpolation. We also observed cluster-based diversity algorithms perform better compared to MAHAKIL in situations where minority instances are sparser (i.e. when dataset is reduced to 10 minority data points). This can be attributed to the nature of MAHAKIL algorithm such that it only performs well when minority data distribution is convex and in situations where there are sufficient number of minority instances [16].

Cluster-based diversity methods also outperform KMEANS-SMOTE. This could be explained by the limitation of KMEANS-SMOTE at high imbalance levels. KMEANS-SMOTE performs clustering at dataset level and it generates synthetic data within each cluster based on selected k-nearest neighbours. In the circumstances where there are only a handful of minority class instances within the cluster, synthetic data points generated by SMOTE will be of relative similarity, resulting in lowered diversity.

With a comparison of NBCDO and DBCDO using their mean performance (Table 4, 5), we can conclude NBCDO performs better when the dataset has few dimensions (e.g. DS 7). In contrast, when there is higher dimensionality within the dataset (e.g. DS 1, 6 and 10), DBCDO performs better. Since density-based clustering performs better when feature set is large, and distance-based clustering performs better when feature set is small, the observation can be explained by the knowledge that NBCDO is based on distance-based clustering, and DBCDO is based on density-based clustering.

An evaluation of the overall statistical performance (Table 6, 7) allows us to conclude that most of these results echo our findings in the comparison of mean performance. We discovered cluster-based diversity measures perform better at extremely imbalanced datasets through special and individualised treatment of isolated instances, relative to existing clustering methods which tends to group them into a specific cluster. It also validates our hypothesis that diversity is more important when minority instances are sparse.

Although most evaluation metrics indicate cluster-based diversity methods as the best-performing methods, there are two metrics, AUC and F1 which favours MC-SMOTE at a 5% imbalanced level. It is worth highlighting that as the imbalance level becomes more extreme (e.g. 10 minority instances), the performance edge of MC-SMOTE over cluster-based diversity methods dissipates. A possible explanation for this is that as the issue of imbalance dataset becomes more prominent, there is greater likelihood of a data point treated as "isolated" and thereby not grouped into a cluster with other minority instances. In contrast to MC-SMOTE, which has strong tendency to group minority instances into clusters, cluster-based diversity methods assess data points individually and help to ensure that isolated data points are correctly identified. This assists in minimising the likelihood of introducing noises (errors) at the commencement of the subsequent synthetic data generation process.

Conclusions

In this study, we propose a new cluster-based diversity re-sampling method named NBCDO, with the aim to complement our previously introduced density-based clustering diversity algorithm (DBCDO). In contrast to DBCDO which uses DBSCAN as an underlying clustering algorithm, clustering for NBCDO is performed based on a recent clustering algorithm NBDOS, which considers data distribution within both minority and majority data space when identifying clusters. NBCDO first utilises NBDOS to identify clusters and isolated instances. It then utilises this information to create synthetic samples while incorporating diversity optimisation to promote diversity within each generation region. Two cluster-based diversity methods, DBCDO (based on DBSCAN) and NBCDO (based NBDOS) are evaluated together with its comparable methods on 10 real-world datasets with $\leq 5\%$ imbalanced ratio and, in most cases, it has been found to have statistically superior performance to its comparable methods.

More importantly, this paper highlights the versatility of NOAH, our diversity optimisation algorithm. When it is paired with both clustering algorithm (DBSCAN and NBDOS), empirical results shows that it consistently outperforms comparable methods in most cases. We summarise

Table 6 Performance results of Mann–Whitney test across datasets with 5% imbalance levels

	NB CDO	DB CDO	DB SMOTE	MAHAKIL	MC SMOTE	KMEANS-MOTE		NB CDO	DB CDO	DB SMOTE	MAHAKIL	MC SMOTE	KMEANS-MOTE
F1													
D1	2	3	0	1	3	1	D6	1	2	0	1	2	0
D2	1	1	0	4	4	0	D7	3	2	1	2	2	0
D3	2	0	0	0	0	0	D8	0	0	0	0	0	0
D4	0	4	0	0	0	4	D9	1	1	0	4	4	0
D5	0	0	5	0	2	2	D10	0	0	0	0	0	0
G-Means													
D1	1	3	0	2	2	1	D6	2	2	0	2	2	0
D2	1	1	0	4	4	0	D7	2	2	1	2	2	0
D3	0	0	0	0	0	0	D8	2	1	0	0	1	1
D4	0	3	0	0	0	4	D9	1	1	0	4	4	0
D5	0	0	3	0	3	3	D10	0	0	0	0	0	0
PR-AUC													
D1	1	1	0	1	1	0	D6	0	0	0	0	0	0
D2	2	2	5	0	2	0	D7	5	2	0	0	0	0
D3	0	0	0	0	0	0	D8	0	0	0	0	0	0
D4	0	0	0	0	0	1	D9	2	2	5	0	2	0
D5	0	0	0	0	0	0	D10	0	0	0	0	0	0
AUC													
D1	5	3	0	0	1	3	D6	0	0	1	0	1	0
D2	1	0	2	2	2	2	D7	3	3	0	1	1	0
D3	0	0	0	0	0	0	D8	0	0	0	0	0	0
D4	0	0	0	0	0	2	D9	0	0	2	2	2	2
D5	2	0	0	0	1	0	D10	0	0	2	0	0	1

Each figure reports the frequency that the selected method is significantly better than its comparable methods within the same dataset ($p < 0.05$)

Table 7 Performance results of Mann–Whitney test across datasets with 10 minority instances

	NB CDO	DB CDO	DB SMOTE	MAHAKIL	MC SMOTE	KMEANS-MOTE		NB CDO	DB CDO	DBSMOTE	MAHAKIL	MC SMOTE	KMEANS-MOTE
F1													
D1	1	3	0	1	1	1	D6	1	2	0	2	2	0
D2	2	2	0	4	4	0	D7	3	2	1	2	2	0
D3	2	0	0	0	0	0	D8	1	1	0	1	1	1
D4	0	3	0	0	0	2	D9	2	3	0	4	4	0
D5	0	0	4	0	0	0	D10	1	1	0	0	2	0
G-Means													
D1	1	5	0	1	1	1	D6	0	3	0	2	3	0
D2	2	2	0	4	4	0	D7	3	2	1	2	3	0
D3	0	0	0	0	0	0	D8	1	4	0	2	2	0
D4	0	3	0	0	0	2	D9	2	3	0	4	4	0
D5	0	0	0	0	3	3	D10	2	2	0	0	1	0
PR-AUC													
D1	1	4	0	1	1	1	D6	0	0	0	0	0	0
D2	2	1	0	0	0	0	D7	1	5	0	0	0	0
D3	0	0	0	0	0	0	D8	0	0	0	1	0	0
D4	0	1	0	0	0	0	D9	2	1	0	0	0	0
D5	0	1	0	0	0	0	D10	0	0	0	0	0	0
AUC													
D1	5	4	0	0	0	0	D6	0	0	0	0	1	1
D2	0	0	5	0	0	0	D7	4	2	0	1	2	0
D3	0	0	0	0	0	0	D8	0	0	0	0	0	0
D4	0	1	0	0	0	2	D9	0	0	5	0	0	0
D5	2	0	0	0	0	0	D10	1	0	1	1	0	0

Each figure reports the frequency that the selected method is significantly better than its comparable methods within the same dataset ($p < 0.05$)

and attribute its superior performance to its ability to identify the minority space for synthetic data generation and its ability to obtain optimal spread of generated instances due to genetic algorithm.

For future work, we may consequently incorporate other typologies of clustering algorithms, such as centroid-based clustering (k-means) and distribution-based clustering (e.g. Gaussian) in conjunction with our diversity optimisation algorithm, NOAH. This would allow us to further test the validity of NOAH algorithm. Additionally, the implementation of NBCDO is based on a fixed hyper-parameters configuration derived from our synthetic experiments. This is a one-size-fits-all approach which is then applied onto each dataset, regardless of their characteristics. For future work, there is a consideration to pre-determine the optimal hyper-parameters configuration and tailored it specifically for the specific dataset. Additionally, due to the superior ability of NBDOS to draw accurate and specific decision boundaries for each minority instances, we would like to extend this algorithm to the multi-class classification problem as the class overlapping issue will be more severe and complex, thereby requiring more sophisticated clustering algorithm before the over-sampling process commences.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Data availability The data used is publically avaiable.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Ali A, Shamsuddin SM, Ralescu AL. Classification with class imbalance problem. Int J Adv Soft Comput Appl. 2013;5:176–204.
2. Y. Sasaki, "The truth of the f-measure. 2007," https://www.cs.odu.edu/mukka/cs795sum09dm/Lecturenotes/Day3/F-measure-YS-26Oct07.pdf [accessed 2021–05–26], 2007.
3. Thabtah F, Hammoud S, Kamalov F, Gonsalves A. Data imbalance in classification: experimental evaluation. Inf Sci. 2020;513:429–41.
4. Japkowicz N, Stephen S. The class imbalance problem: a systematic study. Intell Data Anal. 2002;6(5):429–49.
5. Chawla KW, Bowyer L, Hall O, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. J Artif Intell Res. 2002;16:321–57.
6. Bennin KE, Keung J, Phannachitta P, Monden A, Mensah S. Mahakil: diversity based oversampling approach to alleviate the class imbalance issue in software defect prediction. IEEE Trans Software Eng. 2017;44(6):534–50.
7. Y. Yang, H. A. Khorshidi, and U. Aickelin, "Cluster-based Diversity Over-sampling: A Density and Diversity Oriented Synthetic Over-sampling for Imbalanced Data," in *Proceedings of the 14th International Joint Conference on Computational Intelligence*, 2022, doi: https://doi.org/10.5220/0011381000003332.
8. S. Sharma, C. Bellinger, B. Krawczyk, O. Zaiane, and N. Japkowicz, "Synthetic oversampling with the majority class: A new perspective on handling extreme imbalance," in *2018 IEEE international conference on data mining (ICDM)*, 2018: IEEE, pp. 447–456.
9. Lim P, Goh CK, Tan KC. Evolutionary cluster-based synthetic oversampling ensemble (eco-ensemble) for imbalance learning. IEEE Trans Cybern. 2016;47(9):2850–61.
10. T. E. Tallo and A. Musdholifah, "The implementation of genetic algorithm in smote (synthetic minority oversampling technique) for handling imbalanced dataset problem," in *2018 4th international conference on science and technology (ICST)*, 2018: IEEE, pp. 1–4.
11. Zhu T, Lin Y, Liu Y. Synthetic minority oversampling technique for multiclass imbalance problems. Pattern Recogn. 2017;72:327–40.
12. Y. Y. Yang, H. Akbarzadeh HA Khorshidi, U. U. Aickelin, A. A. Nevgi, and E. E. Ekinci, "On the Importance of Diversity in Re-Sampling for Imbalanced Data and Rare Events in Mortality Risk Models," in *Australasian Computer Science Week Multiconference*, 2021, pp. 1–8.
13. Douzas G, Bacao F, Last F. Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. Inf Sci. 2018;465:1–20.
14. Yi H, Jiang Q, Yan X, Wang B. Imbalanced classification based on minority clustering synthetic minority oversampling technique with wind turbine fault detection application. IEEE Trans Industr Inf. 2020;17(9):5867–75.
15. Sampath V, Maurtua I, Aguilar Martin JJ, Gutierrez A. A survey on generative adversarial networks for imbalance problems in computer vision tasks. J Big Data. 2021;8:1–59.
16. Khorshidi HA, Aickelin U. Constructing classifiers for imbalanced data using diversity optimisation. Inf Sci. 2021;565:1–16.
17. Ester M, Kriegel H-P, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. KDD. 1996;96:226–31.
18. T. Ulrich and L. Thiele, "Maximizing population diversity in single-objective optimization," in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, 2011, pp. 641–648.
19. Muñoz MA, Villanova L, Baatar D, Smith-Miles K. Instance spaces for machine learning classification. Machine Learn. 2018;107:109–47. https://doi.org/10.1007/s10994-017-5629-5.
20. Chawla NV. Data Mining for Imbalanced Datasets: An Overview. In: Maimon O, Rokach L, editors. Data Mining and Knowledge Discovery Handbook. Boston, MA: Springer; 2010.
21. Wong DJN, Oliver CM, Moonesinghe SR. Predicting postoperative morbidity in adult elective surgical patients using the Surgical Outcome Risk Tool (SORT). BJA. 2017;119:95–105. https://doi.org/10.1093/bja/aex117.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.