ASN-SMOTE: a synthetic minority oversampling method with adaptive qualified synthesizer selection

Xinkai Yi¹ · Yingying Xu¹ · Qian Hu¹ · Sujatha Krishnamoorthy² · Wei Li¹ · Zhenzhou Tang¹

Received: 10 July 2021 / Accepted: 23 December 2021 / Published online: 21 January 2022
© The Author(s) 2022

Abstract

Oversampling is a promising preprocessing technique for imbalanced datasets which generates new minority instances to balance the dataset. However, improper generated minority instances, i.e., noise instances, may interfere the learning of the classifier and impact it negatively. Given this, in this paper, we propose a simple and effective oversampling approach known as ASN-SMOTE based on the k -nearest neighbors and the synthetic minority oversampling technology (SMOTE). ASN-SMOTE first filters noise in the minority class by determining whether the nearest neighbor of each minority instance belongs to the minority or majority class. After that, ASN-SMOTE uses the nearest majority instance of each minority instance to effectively perceive the decision boundary, inside which the qualified minority instances are selected adaptively for each minority instance by the proposed adaptive neighbor selection scheme to synthesize new minority instance. To substantiate the effectiveness, ASN-SMOTE has been applied to three different classifiers and comprehensive experiments have been conducted on 24 imbalanced benchmark datasets. ASN-SMOTE is also extensively compared with nine notable oversampling algorithms. The results show that ASN-SMOTE achieves the best results in the majority of datasets. The ASN-SMOTE implementation is available at: <https://www.github.com/yixinkai123/ASN-SMOTE/>.

Keywords Imbalanced dataset · Oversampling · Synthetic minority oversampling · KNN · SMOTE · Noise filtering

Introduction

The problem of class-imbalanced data in machine learning may occur often, which means that the class distribution of data in binary or multi-class classification problems has a significant slant. For instance, within the binary classification problem, the majority class involves a large number of instances, while only a few instances in the minority class [42]. Such problems often appear in practical applications in bank fraudulent transaction detection methods [36], credit risk assessment [34], text classification [41], biomedical diagnosis [2,59] and firewall intrusion detection [5].

Learning from class-imbalanced data poses a tough challenge to conventional classifiers in the field of supervised

learning and data mining. Specifically, when dealing with imbalanced datasets, traditional classifiers usually favor majority class since they were originally designed to process balanced datasets [53]. As a result, the prediction ability of the classifier will be greatly affected by the imbalanced distribution of data, and the minority instances cannot be correctly classified. Especially in cases of extreme training data imbalances, the classifier will be highly influenced by the majority class and achieve high classification accuracy. However, the minority class may even go entirely undetected [14]. For example, for the application of firewall intrusion detection, the malicious attack information found in training data is only 0.1%, so the accuracy rate of classification of normal classifier to recognize all sample information as normal information is 99.9%. This means that even a well-trained classifier cannot be applied to the actual

Due to the universal existence of imbalanced datasets in practical applications and the difficulty for traditional classifiers to deal with them, learning from class-imbalanced data has attracted the attention of many prominent researchers over the last 20 years [31]. Many preprocessing methods have been put forward to deal with class imbalance of datasets.

Xinkai Yi and Yingying Xu contributed equally to this work.

✉ Zhenzhou Tang
mr.tangzz@gmail.com

¹ College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou 325035, China

² Department of Computer Science, Wenzhou-Kean University, Wenzhou 325060, China

These methods can be basically divided into three wide-ranging categories: algorithm level methods, cost-sensitive methods AND data-level methods [23]. Algorithm-level approaches modify or develop algorithms for classifiers to enhance the learning of minority classes [11,27,60]. And the essence of cost-sensitive methods is to minimize the overall cost of incorrect classifications [24,37]. Apart from the above, data-level methods preprocess imbalanced datasets by resampling of data to balance the class distribution [13,28,30,43,52]. Compared with algorithm level and cost-sensitive methods, data-level methods are more conventional ways to tackle imbalanced datasets for they can be applied to classifiers without modifying them [45].

Among the large number of oversampling methods, the synthetic minority oversampling technology (SMOTE) is one of the most well-known methods which generates artificial minority instances by linear interpolation [13]. However, SMOTE and some other existing methods mainly focus on dealing with the imbalance between classes, but fail to effectively solve the issues of within-class imbalances and minor disjuncts [3,13,30]. This may result in some improper synthesized minority instances, such as noise and overlapping instances, which lead to performance degradation instead of improvement. Although there have been some methods proposed that use minority instances around the decision boundary to synthesize instances and can largely address the above issues [16,21,28,43,44,52], however, this makes the decision boundary even more indistinct and blurred, and may cause the problem of the new synthesized samples being shown in the majority class region. This amplifies the noise and degrades the performance of classifiers [56].

In view of this, this paper proposes a simple while effective synthetic minority oversampling technology (SMOTE), known as ASN-SMOTE, which adaptively nominates qualified minority neighbors to synthesize minority instances. More specifically, for each instance of minority class, the Euclidean distances between it and all other instances are firstly calculated. After that, the minority instance whose nearest neighbor is a majority instance is considered to be noise and is filtered. On the other hand, if the minority instance's nearest neighbor is also a minority one, it adaptively determines the qualified synthesizer among its k -nearest neighbors (KNN) according to the Euclidean distances to them. Finally, these qualified neighbors are used to generate minority instances by the SMOTE algorithm [13]. The synthesized minority instances are combined with imbalanced dataset and fed into a traditional supervised classifier for training.

ASN-SMOTE can be applied to real-world datasets and improve the accuracy of the supervised classifiers without modifications.

The main advantages of ASN-SMOTE are threefold: (1) It can filter noise and improve the effectiveness of over-

sampling. (2) It makes full use of the information of the imbalanced dataset to adaptively select qualified synthesizers, which makes the synthesized minority instances more accurate to the practical data distribution. (3) It can be combined with any classification algorithm and there is no need to modify the classification algorithms.

To evaluate the performance of ASN-SMOTE, extensive comparative experiments were conducted. The proposed ASN-SMOTE was applied on 24 public benchmark datasets with the imbalance ratio ranging from 1.82 to 41.4. Three classifiers, namely the k -nearest neighbor (KNN) [18], SVM [17] and random forest (RF) [8], were used for classification. ASN-SMOTE was then compared with seven other notable oversampling techniques, namely (1) Random oversampling, (2) SMOTE [13], (3) ADASYN [30], (4) borderline1-SMOTE [28], (5) borderline2-SMOTE [28], (6) k -means SMOTE [21], (7) SVM-SMOTE [44], (8) SWIM-RBF [7], and (9) LoRAS [6]. The results show that ASN-SMOTE achieves the best results on the majority of datasets in terms of three different evaluation indicators.

The remainder of the paper is organized as follows. In “Related work”, prior research alongside other existing sampling methods are reviewed, especially in regard to oversampling methods, in addition to problems that exist in practical application. In “Oversampling algorithm for adaptively selecting neighbor synthesis instances”, the working principle of the proposed oversampling method are introduced in detail. And in “Experiments and results”, the experimental design, results, and discussion are described. Finally, the conclusion is presented in “Conclusion”.

Related work

In this section, data-level preprocessing methods are reviewed and summarized. Generally, these preprocessing methods can be categorized into three types, namely the oversampling, undersampling and hybrid sampling methods [20]. Oversampling refers to the use of sampling methods to generate new minority instances to balance the dataset [13,28,30,43,52]. In contrast, undersampling is employed to tackle the problem of imbalanced data sets by deleting some majority instances [29,38,39,54,55,57]. Table 1 summarizes the existing oversampling, undersampling and hybrid sampling methods that have been successfully applied to deal with imbalanced data problems.

Undersampling

Random undersampling is a common method which balances the class distribution of imbalanced datasets by randomly removing some majority instances. However, this method may cause the loss of some valuable classification infor-

Table 1 Summary of existing oversampling (O), undersampling (U) and hybrid sampling (H) techniques

Type	Sampling technique	Pros	Cons
O	SMOTE [13]	It synthesizes minority samples	May introduce noise and cause over-fitting
O	Borderline-SMOTE [28]	It improves the class distribution	May cause the over-lapping of majority minority samples
O	ADASYN [30]	It gives different weights to different minority samples	Can not find all minority boundary instances
O	Cluster-SMOTE [16]	It addresses the within-class imbalances	Its effectiveness depends on the number of clusters
O	Safe-Level-SMOTE [9]	It can avoids synthesizing more overlapping instances	It may cause over-fitting
O	SVM-SMOTE [44]	It expands minority class area	Not suitable for datasets with overlapping class conditional distributions
O	Gaussian-SMOTE [40]	It fully utilizes majority information to generate new samples in minority areas	Highly complex
O	DBSMOTE [10]	It avoids synthetic instances appearing in majority classes	The importance of the minority instances close to the decision boundary is ignored
O	MWMOTE [3]	It avoids synthesizing minority instances that overlap with the majority classes	It ignores small clusters of minority classes far away from majority classes
O	A-SUWO [43]	It can avoid the minority sub-clusters being ignored	It may not obtain wider and more effective instances
O	K-means SMOTE [21]	It can avoid noise interference problems to a certain extent and synthesize a large number of relatively safe instances	It ignores the importance of the minority instances close to the decision boundaries
O	SWIM-RBF [7]	It uses the density of the well-sampled majority class to guide the synthesis	It has high computational complexity
O	LoRAS [6]	It provides a better estimate for the mean of the underlying local data distribution of the minority class data space	It involves a large number of parameters
U	CNN [29]	It considers the distribution of samples	Noisy samples may lead to CNN running slowly
U	ENN [57]	It can optimally eliminate outliers and possible overlap among classes from a given training set	The accuracy of the ENN classifier converges to Bayes error as the number of instances approaches infinity
U	TomekLink [54]	It explicitly seeks to find boundary points	It is not suitable for small datasets
U	NCL [39]	NCL cleans neighborhoods that misclassify examples belonging to the class of interest	Samples with important information may be deleted
U	OSS [38]	It uses instance-based methods to reduce the larger class	The CNN rule is extremely sensitive to noise
U	EUS [26]	The majority selection mechanism helps to obtain more accurate subsets of instances than the global selection mechanism	The global selection mechanism is necessary to achieve the highest reduction rates
U	IHT [49]	Incorporating instance hardness into the learning process significantly increase classification accuracy.	Calculating instance hardness is computationally expensive
U	ACO-Sampling [58]	It addresses imbalanced DNA microarray data classification problem	The sampling procedure is time-consuming
U	IRUS [51]	The accuracy of multi label classification is improved	Deleting majority samples may result in the loss of important information
H	SMOTE-RSB [48]	It deletes noise and inappropriate data	It does not accurately represent the problem space

Table 1 continued

Type	Sampling technique	Pros	Cons
H	SMOTE-IPF [50]	Overcome the problems produced by noisy and borderline examples in imbalanced datasets	It has too many parameters to choose
H	SMOTE + RBFN [25]	It can overcome the problem of high imbalance ratio	It Needs large storage space construct network
H	NBBAG [35]	Compared with the same bagging oversampling methods, it achieves better results	High cost is required
H	RUS-IPF [15]	It reduces the noise and makes the class boundaries more regular	Sample size affects performance
H	CUSBoost [47]	It can effectively process the data with high imbalance ratio	
H	CBIS [19]	It determines and removes unrepresented samples in data	High complexity

mation during the training process of a classifier. To overcome this shortcoming, researchers have proposed distinctive undersampling approaches. The condensed nearest neighbor rule (CNN) [29] and Tomek link [54] are some examples. In the CNN method, some of the majority instances around the decision boundary are removed and the remaining instances are used for training. When the Tomek Link method is utilized, the Tomek Link pair is calculated, which is the nearest neighbors of two instances of different classes between each other, resulting in a deletion of the majority instance of the Tomek Link pair [4]. In general, when the dataset is minute or extremely imbalanced, these undersampling methods cannot overcome the class-imbalance problem. This results in the removal of a large number of majority instances and the classifier cannot achieve an effective out-of-sample performance [21,53].

Oversampling

Random oversampling is the simplest oversampling method. It randomly selects minority instances and copies them until the class distribution of imbalanced dataset becomes balanced. Nevertheless, this method generates samples similar to the original data resulting in over-fitting of the classification algorithm [4,14].

To avoid the problem of over-fitting caused by random oversampling, Chawla et al. proposed the use of the SMOTE algorithm [13] in 2002. Contrary to merely copying the existing instances, this algorithm generates artificial instances by linear interpolation. As shown in Fig. 1, SMOTE algorithm generates minority instances by interpolating between the minority instance and the k nearest minority class neighbors. Specifically, the SMOTE algorithm generates minority instances in three steps. First, it chooses a minority instance $\vec{x}$, then randomly selects one $\vec{y}$ of the K nearest minority

Fig. 1 The selected minority instance and its $k = 5$ nearest neighbors are used to generate five new samples by SMOTE linearly interpolation

neighbors of $\vec{x}$. Eventually, a new instance $\vec{s}$ is synthesized, which is the random interpolation between these two instances calculated by formula $\vec{s} = \vec{x} + \lambda \times (\vec{y} - \vec{x})$, where λ is a random weight in $[0,1]$. However, SMOTE treats all minority instances alike and ignores the locations of the minority instances selected for synthesizing new instances, which results in the synthesized minority instances located in the majority class region, causing noise and deteriorating the classification performance, as shown in Fig. 2.

Given the above problems, a variety of improved oversampling methods have been proposed. Adaptive synthesis sampling (ADASYN) [30] and its variant KernelADASYN [52] adaptively adjust sampling weights according to the proportion of majority instances in k -nearest neighbors. In this way, minority instances that are similar to the majority instances are more likely to be sampled. Similar to ADASYN, Borderline-SMOTE [28] has two sampling strategies. Borderline1-SMOTE1 focuses attention to the decision boundary. First, it identifies the minority instances around the decision boundary, and then synthesizes instances with these minority instances, so as to balance the class distribution of

Fig. 2 Due to the noise in the imbalance dataset, the new samples synthesized by SMOTE are likely to be located in the majority class region

the dataset. On the other hand, Borderline2-SMOTE extends the previous method allowing minority instances to interpolate with only one majority class neighbors, and setting the interpolation weight to less than 0.5. The aim of this is to improve the similarity between the minority instance and the synthetic instance, so as to enhance the ability of classifier to distinguish the instance around the decision boundary.

Additionally, another kind of sampling methods are clustering-based oversampling algorithms. Cluster-SMOTE initially uses the k -means clustering algorithm to divide the minority instances into several clusters and applies SMOTE in each cluster [16]. In Ref. [43], an adaptive semi-supervised weighted over-sampling (A-SUMO) approach was presented. A-SUMO first utilizes a semi-supervised hierarchical clustering algorithm to cluster minority instances and adaptively determines the oversample size to each sub-cluster. Recently, a heuristic oversampling method based on k -means and SMOTE was proposed by Douzas et al. [21]. This method clusters the entire sample set, and calculates the imbalance rate (IR) of each cluster. According to the size of IR, SMOTE is used within the cluster. The main idea of SVM-SMOTE (Support Vector Machine SMOTE) [44] is to use an SVM classifier to find the support vector, and then synthesize new instances based on it. Similar to Borderline-SMOTE, SVM-SMOTE also determines the type of instances (safe, danger, noise) according to the attributes of k -nearest neighbors, and then train SVM according to the danger type [44].

The above methods suggested that the minority instances around the decision boundary can be used to synthesize instances and can largely address the issues of within-class imbalances and minor disjuncts. However, this blurs the decision boundary and may generate minority instances in the majority instances region (see Fig. 3), which amplifies the noise and degrades the performance of classifiers [56].

Fig. 3 The new instances generated by clustering are located in the majority class region

Unlike SMOTE and its variants that rely on the positions and distances between minority instances, SWIM-RBF [7] uses the density of minority instances relative to the distribution of majority class to determine where to generate synthetic instances. LoRAS [6] provides a method to choose neighborhoods of a minority class instance point by performing prior manifold learning over the minority class using t-Stochastic Neighborhood Embedding.

To address more effectively the issue of how to determine the proper minority instances involved in the synthesis to avoid the generated minority instances falling into the majority class region, in this paper, we present ASN-SMOTE which is also an improved variant of SMOTE. Similar to Borderline-SMOTE [28] and some other variants of SMOTE [21,44], ASN-SMOTE performs noise filtering before synthesizing minority instances. However, in ASN-SMOTE, a minority instance whose nearest neighbor is a majority instance is considered to be noise and is filtered. It is different from other SMOTE-based techniques that determine whether an instance is noise according to the proportion of majority instances in its k -nearest neighbors. Another difference between ASN-SMOTE and other oversampling techniques is that ASN-SMOTE only selects minority neighbors that are closer than nearest majority instance for synthetic interpolation. These two mechanisms help ASN-SMOTE selecting proper minority instances to synthesize minority instances which will not appear in majority class regions.

Oversampling algorithm for adaptively selecting neighbor synthesis instances

Previously, we have reviewed some related works and presented the existing issues. Motivated by this, we present the novel synthetic minority oversampling method with adaptive qualified synthesizer selection, known as ASN-SMOTE, in this section. The fundamental idea behind the ASN-SMOTE algorithm is to make full use of all the information of the k

Fig. 4 Identify the noise and the minority instances around decision boundary by judging the nearest instance of each minority instance in the whole dataset

nearest neighbor (KNN) instances to synthesize the appropriate minority instances. The ASN-SMOTE involves the following three steps: (1) noise filtering, (2) adaptively selecting neighbor instances, and (3) synthesizing instances.

Noise filtering

Filtering noise is an essential process in the training stage of machine learning because noise is a kind of interference for sampling algorithms and classifiers [12]. It not only invalidates the sampling algorithm, but also reduces the overall performance of the classifier. In this work, a KNN-based noise filtering algorithm is proposed to reduce the generation of synthetic minority instances located in the majority class regions. The proposed algorithm can accurately identify noise and the minority instances surrounding the decision boundary by judging the nearest instance of each minority instance, and filter them.

Specifically, denote the majority class set, the minority class set and the complete dataset as $\mathbf{N}$, $\mathbf{P}$ and $\mathbf{T}$, respectively. N_p and N_n are the numbers of minority instances and majority instances respectively. For each instance $p_i \in \mathbf{P}$, ($i = 1, 2, \dots, N_p$), calculate the Euclidean distance to each instance in the $\mathbf{T}$, and find the nearest instance $D(p_i)$:

$$D(p_i) = \arg \min_{t \in \mathbf{T}, t \neq p_i} \|p_i - t\|_2, \quad (1)$$

where $\|x - y\|_2$ returns the Euclidean distance between x and y .

If $D(p_i) \in \mathbf{N}$, p_i is regarded as noise or a minority instance around the decision boundary and is recorded as an unqualified instance, $\mathbf{M}_u = \mathbf{M}_u \cup p_i$, where $\mathbf{M}_u$ is the set of unqualified minority instances of p_i . Otherwise, $\mathbf{M}_q = \mathbf{M}_q \cup p_i$ where $\mathbf{M}_q$ is the set of qualified minority instances of p_i (see Fig. 4).

The pseudo-code of the noise filtering method is presented in Algorithm 1.

Algorithm 1 Noise filtering

```

1: Initialization:  $\mathbf{M}_u = \emptyset$ ,  $\mathbf{M}_q = \emptyset$ .
2: for each  $p_i \in \mathbf{P}$ , ( $i = 1, 2, \dots, N_p$ ) do
3:    $D(p_i) = \arg \min_{t \in \mathbf{T}, t \neq p_i} \|p_i - t\|_2$ .
4:   if  $D(p_i) \in \mathbf{N}$  then
5:      $\mathbf{M}_u = \mathbf{M}_u \cup p_i$ .
6:   else
7:      $\mathbf{M}_q = \mathbf{M}_q \cup p_i$ .
8:   end if
9: end for

```

Adaptive selection of neighbor instances

Neighbor instance selection greatly affects the quality of the generated instances, where the challenge is to select the appropriate nearest neighbors that qualify for a minority class instance. In the original SMOTE algorithm, minority class instances are randomly selected to synthesize new instances. This may result in that the synthesized instances locate in the majority class region [37]. Using these synthetic instances as training data reduces the performance of the classifier. Given this, an adaptive neighbor selection strategy is proposed in this paper.

Specifically,

- (1) For each instance in $\mathbf{M}_q$, denoted as q_j , $j = 1, 2, \dots, N_q$, find the k nearest neighbor instances of q_j in $\mathbf{T}$ by Euclidean distance calculation, denoted as $q_j^1, q_j^2, \dots, q_j^k$.
- (2) If all the k nearest neighbor instances, i.e., $q_j^1, q_j^2, \dots, q_j^k$ are minority instances, all of them are added to the set $\mathbf{Q}_j$ where $\mathbf{Q}_j$ represents the set of qualified neighbors of the qualified minority instance q_j . Otherwise, go to step 3.
- (3) If there is at least one majority instance in the k nearest neighbor instances, find the nearest majority instance, denoted as q_j^{near} , among the k nearest neighbor instances of q_j .
- (4) Each neighbor in the minority class whose distance to q_j is smaller than $\|q_j - q_j^{\text{near}}\|_2$ is added to the set $\mathbf{Q}_j$.

The pseudo-code of the adaptive neighbor instances selection method is presented in Algorithm 2.

Synthesizing instances

Synthesizing the minority class instances is the last step of the oversampling technique for imbalanced data. The instance synthesis method in this work follows the original linear interpolation for it is easy to calculate and does not cause the synthesized instance to be located in the majority class regions due to improper weight allocations.

Specifically, for the qualified minority instance q_i , we first determine the number of synthesis instances (N_s) for each

Algorithm 2 Adaptive neighbor instances selection

```

1: for each  $q_j \in \mathbf{M}_q$ , ( $j = 1, 2, \dots, N_q$ ) do
2:    $\mathbf{Q}_j = \emptyset$ .
3:    $q_j^{\text{near}} \leftarrow \text{null}$ .
4: end for
5: for each  $q_j \in \mathbf{M}_q$ , ( $j = 1, 2, \dots, N_q$ ) do
6:    $\mathbf{T}' \leftarrow \mathbf{T}$ .
7:   for  $m = 1 \rightarrow k$  do
8:      $q_j^m \leftarrow \arg \min_{t \in \mathbf{T}', t \neq q_j} \|q_j - t\|_2$ .
9:     if  $q_j^m \in \mathbf{N}$  and  $q_j^{\text{near}}$  is null then
10:       $q_j^{\text{near}} \leftarrow q_j^m$ .
11:     end if
12:      $\mathbf{T}' \leftarrow \mathbf{T}' - q_j^m$ .
13:   end for
14:   if  $q_j^{\text{near}}$  is null then
15:      $\mathbf{Q}_j \leftarrow \{q_j^1, q_j^2, \dots, q_j^k\}$ .
16:   else
17:     for  $m = 1 \rightarrow k$  do
18:       if  $\|q_j - q_j^m\|_2 \leq \|q_j - q_j^{\text{near}}\|_2$  then
19:          $\mathbf{Q}_j \leftarrow \mathbf{Q}_j \cup q_j^m$ .
20:       end if
21:     end for
22:   end if
23: end for

```

Fig. 5 ASN-SMOTE adaptively selects the qualified neighbors through noise filtering to synthesize instances ($k = 4$)

qualified minority instance as follows:

$$N_s = \text{round} \left[\frac{N_n - N_p}{N_q} \right]. \quad (2)$$

After that, randomly select a qualified neighbor from $\mathbf{Q}_i$, denoted as q'_i , and perform linear interpolation to synthesize an instance (s) as follows:

$$s = q_i + w \times (q'_i - q_i), \quad (3)$$

where $w \in (0, 1)$ is the weight. This random selection and linear interpolation is executed for N_s times for each qualified minority instance.

Algorithm 3 Procedure of ASN-SMOTE

Input: The original imbalanced dataset $\mathbf{T}$ and the number of nearest neighbors to be found for each qualified minority instance k .

Output: The oversampling dataset $\mathbf{S}$.

```

1: Divide  $\mathbf{T}$  into two subset:  $\mathbf{N}$  and  $\mathbf{P}$ .
2: Perform noise filtering and obtain  $\mathbf{M}_q$  by Algorithm 1.
3: for each  $q_i \in \mathbf{M}_q$ , ( $i = 1, 2, \dots, N_q$ ) do
4:   Perform adaptive neighbor selection and obtain  $\mathbf{Q}_i$  by Algorithm 2.
5: end for
6: Calculate  $N_s$  by Eq.(2).
7: for each  $q_i \in \mathbf{M}_q$ , ( $i = 1, 2, \dots, N_q$ ) do
8:   Synthesize  $N_s$  instances by Eq.(3).
9: end for

```

As shown in Fig. 5, the noise filtered minority class instances adaptively select qualified minority class neighbor instances to synthesize new instances. The synthesized minority instances are generated from the original minority instances and do not reside in the region of the majority class.

The procedure of the proposed ASN-SMOTE is summarized as follows.

Experiments and results

This section focuses on the validation and evaluation of the proposed oversampling method. The ASN-SMOTE is evaluated with different classifiers, datasets in terms of various performance measures.

Experimental datasets

To evaluate the performance of ASN-SMOTE, experiments were conducted on 24 public datasets that were selected from the KEEL repository [1] and the UCI Machine Learning Repository [22]. Considering that the datasets of real-world applications are quite complex, in the case of that a dataset contains more than two categories, one of the smaller categories is selected as the minority class. The rest of the data is used as the majority class. Additionally, to evaluate the adaptability of ASN-SMOTE under extreme conditions, two large datasets with high imbalance rate were selected as well. The detailed information of the experimental datasets is shown in Table 2.

Evaluation measures

In this work, the geometric mean (G -mean) of accuracy, F -measure and area under the curve (AUC) were selected as the evaluation measures, which are, respectively, defined as follows.

Table 2 Information description of the experimental data set

Dataset	Num. of attributes	Category	Size. of minority class	Size. of majority class	Num. of instances	IR
Haberman	3	2:1	81	225	306	2.78
Pima	8	1:0	268	500	768	1.87
Car	6	'v-good':others	65	1663	1728	25.58
Iris	4	2:others	50	100	150	2
Ecoli	7	'pp':others	52	284	336	5.46
Seeds	7	1:others	70	140	210	2
Wine	13	1:others	48	130	178	2.71
Dermatology-6	34	6:others	20	338	358	16.9
Newthyroid2	5	2:others	35	180	215	5.14
Pageblock	10	others:1	560	4913	5473	8.77
Glass0	9	0:others	70	144	214	2.06
Glass1	9	1:others	76	138	214	1.82
Glass4	9	4:others	13	201	214	15.47
Glass5	9	5:others	9	205	214	22.78
Glass6	9	6:others	29	185	214	6.38
Yeast	8	2:others	429	1055	1484	2.46
Yeast1	8	3:others	244	1240	1484	5.08
Yeast2	8	4:others	163	1321	1484	8.1
Yeast3	8	5:1	51	463	514	9.07
Yeast4	8	7:2	35	429	464	12.25
Yeast5	8	9:1	20	463	483	23.15
Yeast6	8	5:others	51	1433	1484	28.09
Yeast7	8	6:others	44	1440	1484	32.72
Yeast8	8	7:others	35	1449	1484	41.4

Geometric evaluation accuracy:

$$G\text{-mean} = \sqrt{\text{Sensitivity} \times \text{Specificity}}.$$

The harmonic mean of Sensitivity and Precision:

$$F\text{-measure} = \frac{2 \times \text{Sensitivity} \times \text{Precision}}{\text{Sensitivity} + \text{Precision}}.$$

AUC is the area under the receiver operating characteristics (ROC) curve. The ROC curve is a two-dimensional coordinate graph in which the X -axis represents the false positive rate (FPR) and Y -axis represent the true positive rate (TPR). It is a commonly used performance measure to evaluate the imbalance problem and can be calculated as:

$$\text{AUC} = \frac{1 + \text{TPR} - \text{FPR}}{2},$$

where $\text{TPR} = \frac{\text{TP}}{N_p}$ and $\text{FPR} = \frac{\text{FP}}{N_n}$, TP represents the proportion of positive (minority) class instances correctly classified, and FP represents the proportion of negative (majority) class instances that is incorrectly classified as positive class.

Experimental configurations

To conserve the between-class ratios of the original datasets, a fivefold stratified cross validation was used to ensure that the original class distribution is preserved in each fold. Furthermore, each experiment was conducted three times to obtain the average metrics value as to avoid the impact of randomness. To evaluate the performance of the proposed method, it was compared with nine other oversampling techniques: (1) random over-sampling, (2) SMOTE, (3) ADASYN, (4) Borderline1-SMOTE, (5) Borderline2-SMOTE, (6) k -means SMOTE, (7) SVM-SMOTE, (8) SWIM-RBF, and (9) LoRAS. Three classifiers, namely KNN, SVM and Random Forests (RF) were used to make classifications. These algorithms were implemented using python on a 64-bit operating system with 8 GB available RAM using 2.30 GHz i5-8300 CPU.

For SMOTE and its variants, namely Borderlin1-SMOTE, Borderlin2-SMOTE, k -means SMOTE, and ASN-SMOTE, the number of nearest neighbors were selected among the values (3, 5, 7, 9) and the number of clusters in k -means SMOTE was selected from (3, 5, 7, 9, 10, 15, 20), subject

to detail dataset. For SWIM-RBF, epsilon was selected from (0.25, 0.5, 1, 2, 3, 5). And for LoRAS, the number of shadows was selected from 2 to the number of features and the number of nearest neighbors were selected among the values (3, 5, 7, 9). SVM-SMOTE and ADASYN were implemented in the Python library scikit-learn [46] with default parameters.

For the KNN classifier [18], the number of nearest neighbors was selected from the values (3, 5, 7, 9). For the SVM classifier [17], the cost parameter was selected from the values (0.001, 0.01, 0.1, 1, 10). The parameter kernel width γ was selected from the values (0.1, 0.3, 0.5, 0.7, 1, 2, 5). For the Random Forests (RF) classifier [8], the parameter $n_estimators$, i.e. the number of the decision trees of the estimators, was selected from the values (20, 30, 50, 80, 100). The parameter max_depth , which is the maximum depth of the decision tree, was selected from the values (5, 10, 13, 15, 20). And the Gini index was selected as the node partition standard. The parameter $max_features$, i.e., the number of features to consider when looking for the best split, was the square radial of the number of features in the dataset.

The parameters of three classifiers and eight over-sampling methods were optimized using fivefold stratified cross-validation in the 24 public datasets based on G -means, F -measure and AUC.

Results and discussion

The average results and the standard deviations of the G -means, F -measures and AUCs obtained by three classifiers are shown in Tables 3, 4 and 5, respectively. The best experimental results on each dataset are highlighted in bold. In addition, Fig. 6 summarizes the scores and the comprehensive ranking of these oversampling techniques in 24 datasets in terms of convergence accuracy. The three performance measures obtained from fivefold cross validation are used as the ranking basis. For each dataset, the first rank is allocated to the optimum oversampling technique and the eighth rank is assigned to the worst performing technique. Consequently, the average ranking of the three performance measures of each oversampling techniques in all 24 datasets is calculated. The mean ranking of each method ranged between [1.0, 10.0].

From the results, it is shown that ASN-SMOTE can obtain the best performance in at least one performance measure in 18 out of 24 public datasets when SVM, KNN, and RF were used. Figure 6 also indicates that ASN-SMOTE ranks first overall no matter which classifier and performance indicator is adopted. This is because there are always noise minority instances in practical datasets and exist the issues of within-class imbalance and small disjuncts. And the noise filtering and adaptive neighbor selection proposed in ASN-SMOTE are promising mechanisms to deal with them by avoiding synthesizing new minority instances in the majority class region. We can also observe that ASN-SMOTE shows high perfor-

mance as well on some datasets that have high imbalance ratios and sparse minority instances. This is possibly because such datasets have more severe within-class issues and hence noise instances account for a larger proportion of minority instances. As a result, a larger proportion of synthetic minority instances are likely to be located in the majority class region, which makes the advantages of ASN-SMOTE to be fully exploited.

On the other hand, we find that the random oversampling method performs the most poorly on all three measures when KNN was used as the classifier. When SVM and RF were used as the classifier, Borderlin2-SMOTE method performs the worst on all three measures. Compared to ADASYN, k -means SMOTE, SVM-SMOTE, Borderlin2-SMOTE, SMOTE performs better in term of AUC, and Borderline1-SMOTE and SMOTE perform better in terms of G -mean and F -measure. ADASYN, K -means SMOTE and SVM-SMOTE perform moderately according to all three classifiers and three measures. The two recently proposed algorithms, i.e., SWIM-RBF and LoRAS perform well on all three measures when KNN and SVM were used as the classifier.

To have a deep insight into the effects of the noise filtering and adaptive neighbor selection mechanisms, we applied SMOTE, SMOTE/NF (SMOTE with noise filtering) and ASN-SMOTE to three two-dimensional datasets, namely the toy dataset, moon dataset and circle dataset. The distributions of synthetic instances are shown in Fig. 7. It can be observed that SMOTE, without noise filtering and adaptive neighbor selection, generates the largest number of minority instances in the majority class region. By contrast, SMOTE/NF reduces the number of minority instances located in the majority class region since the minority instances that are closer to the majority instances are rejected to actively generate instances with other minority instances. However, there are still some synthesis minority instances in the majority class region. The reasons are twofold. One is that in SMOTE/NF, there may be a majority class region between two minority instances used to synthesize instances. The other is that noise minority instances may be selected to generate minority instances. Both situations may cause the generated minority instance to be located in the majority class region. The adaptive neighbor selection mechanism leveraged in ASN-SMOTE can avoid such situations. As we can see from the figures, all instances generated by ASN-SMOTE locate in the minority class region, owing to the noise filtering and adaptive neighbor selection mechanisms.

Besides the performance comparison, Friedman's test [32] followed by Holm's test [33] were performed to verify the statistical significance of the proposed method compared to the other oversampling methods. The Friedman's test is a non-parametric statistical test. Similar to the repeated measures ANOVA (analysis of variance), it is utilized to distinguish

Table 3 Results obtained by KNN on datasets oversampled by different techniques

Method	G-mean	F-measure	AUC	G-mean	F-measure	AUC	G-mean	F-measure	AUC
Dataset	Haberman			Pima			Car		
Random	0.530 ± 0.063	0.362 ± 0.068	0.628 ± 0.040	0.691 ± 0.057	0.609 ± 0.065	0.769 ± 0.051	0.913 ± 0.071	0.566 ± 0.322	0.965 ± 0.030
SMOTE	0.599 ± 0.124	0.442 ± 0.133	0.639 ± 0.069	0.724 ± 0.038	0.648 ± 0.044	0.779 ± 0.053	0.915 ± 0.064	0.566 ± 0.321	0.971 ± 0.025
ADASYN	0.574 ± 0.060	0.418 ± 0.065	0.609 ± 0.056	0.695 ± 0.048	0.618 ± 0.053	0.765 ± 0.062	0.924 ± 0.059	0.545 ± 0.306	0.954 ± 0.047
Borderline1-SMOTE	0.605 ± 0.056	0.451 ± 0.059	0.647 ± 0.043	0.706 ± 0.048	0.632 ± 0.053	0.760 ± 0.053	0.849 ± 0.135	0.498 ± 0.303	0.961 ± 0.034
Borderline2-SMOTE	0.623 ± 0.068	0.471 ± 0.074	0.648 ± 0.064	0.718 ± 0.042	0.645 ± 0.046	0.764 ± 0.053	0.785 ± 0.177	0.289 ± 0.215	0.894 ± 0.082
K-means SMOTE	0.560 ± 0.084	0.420 ± 0.106	0.663 ± 0.067	0.695 ± 0.044	0.612 ± 0.053	0.779 ± 0.050	0.920 ± 0.059	0.571 ± 0.317	0.969 ± 0.027
SVMSMOTE	0.573 ± 0.096	0.411 ± 0.105	0.635 ± 0.073	0.715 ± 0.039	0.638 ± 0.044	0.772 ± 0.055	0.916 ± 0.077	0.539 ± 0.353	0.967 ± 0.027
SWIM-RBF	0.599 ± 0.090	0.441 ± 0.100	0.660 ± 0.077	0.700 ± 0.029	0.618 ± 0.034	0.778 ± 0.052	0.924 ± 0.032	0.584 ± 0.294	0.978 ± 0.034
LoRAS	0.628 ± 0.072	0.475 ± 0.078	0.681 ± 0.062	0.722 ± 0.042	0.645 ± 0.049	0.777 ± 0.049	0.919 ± 0.025	0.585 ± 0.276	0.978 ± 0.031
ASN-SMOTE	0.661 ± 0.055	0.511 ± 0.059	0.708 ± 0.061	0.727 ± 0.045	0.652 ± 0.052	0.788 ± 0.057	0.944 ± 0.044	0.590 ± 0.224	0.976 ± 0.036
Dataset	Iris			Ecoli			Seeds		
Random	0.939 ± 0.057	0.910 ± 0.083	0.986 ± 0.018	0.866 ± 0.115	0.678 ± 0.171	0.898 ± 0.109	0.906 ± 0.061	0.868 ± 0.079	0.932 ± 0.042
SMOTE	0.955 ± 0.040	0.933 ± 0.057	0.988 ± 0.015	0.881 ± 0.123	0.738 ± 0.195	0.901 ± 0.110	0.909 ± 0.055	0.876 ± 0.078	0.932 ± 0.053
ADASYN	0.930 ± 0.068	0.896 ± 0.097	0.982 ± 0.017	0.835 ± 0.102	0.603 ± 0.140	0.885 ± 0.110	0.902 ± 0.063	0.857 ± 0.086	0.940 ± 0.056
Borderline1-SMOTE	0.936 ± 0.071	0.910 ± 0.095	0.986 ± 0.014	0.872 ± 0.127	0.719 ± 0.218	0.892 ± 0.120	0.903 ± 0.070	0.860 ± 0.092	0.942 ± 0.055
Borderline2-SMOTE	0.942 ± 0.060	0.916 ± 0.085	0.985 ± 0.014	0.882 ± 0.131	0.748 ± 0.209	0.907 ± 0.110	0.894 ± 0.061	0.849 ± 0.081	0.930 ± 0.051
K-means SMOTE	0.960 ± 0.034	0.942 ± 0.048	0.987 ± 0.017	0.891 ± 0.130	0.777 ± 0.216	0.905 ± 0.104	0.909 ± 0.050	0.876 ± 0.067	0.942 ± 0.045
SVMSMOTE	0.927 ± 0.071	0.892 ± 0.101	0.980 ± 0.016	0.867 ± 0.133	0.708 ± 0.213	0.897 ± 0.107	0.899 ± 0.064	0.857 ± 0.084	0.948 ± 0.039
SWIM-RBF	0.950 ± 0.048	0.925 ± 0.069	0.986 ± 0.018	0.878 ± 0.112	0.715 ± 0.178	0.901 ± 0.104	0.906 ± 0.044	0.867 ± 0.060	0.938 ± 0.049
LoRAS	0.949 ± 0.042	0.925 ± 0.063	0.986 ± 0.017	0.884 ± 0.126	0.748 ± 0.202	0.904 ± 0.108	0.909 ± 0.052	0.870 ± 0.069	0.937 ± 0.046
ASN-SMOTE	0.955 ± 0.040	0.933 ± 0.057	0.988 ± 0.015	0.889 ± 0.126	0.765 ± 0.211	0.904 ± 0.102	0.912 ± 0.048	0.879 ± 0.066	0.937 ± 0.049
Dataset	Wine			Dermatology-6			Newthyroid2		
Random	0.856 ± 0.048	0.811 ± 0.074	0.973 ± 0.033	0.972 ± 0.053	0.949 ± 0.063	0.973 ± 0.050	0.926 ± 0.048	0.831 ± 0.025	0.963 ± 0.059
SMOTE	0.906 ± 0.037	0.864 ± 0.064	0.978 ± 0.026	0.969 ± 0.055	0.906 ± 0.092	0.972 ± 0.049	0.919 ± 0.066	0.847 ± 0.055	0.990 ± 0.007
ADASYN	0.902 ± 0.059	0.859 ± 0.088	0.964 ± 0.042	0.970 ± 0.052	0.927 ± 0.061	0.972 ± 0.049	0.935 ± 0.046	0.827 ± 0.031	0.977 ± 0.032
Borderline1-SMOTE	0.921 ± 0.057	0.878 ± 0.086	0.970 ± 0.035	0.969 ± 0.055	0.906 ± 0.092	0.972 ± 0.052	0.944 ± 0.057	0.858 ± 0.078	0.992 ± 0.008
Borderline2-SMOTE	0.912 ± 0.059	0.866 ± 0.087	0.979 ± 0.022	0.964 ± 0.053	0.855 ± 0.120	0.972 ± 0.052	0.921 ± 0.064	0.785 ± 0.119	0.976 ± 0.029
K-means SMOTE	0.904 ± 0.052	0.874 ± 0.059	0.978 ± 0.023	0.969 ± 0.055	0.906 ± 0.092	0.973 ± 0.050	0.859 ± 0.123	0.801 ± 0.124	0.974 ± 0.029
SVMSMOTE	0.922 ± 0.056	0.881 ± 0.084	0.969 ± 0.030	0.967 ± 0.054	0.888 ± 0.102	0.973 ± 0.050	0.913 ± 0.061	0.826 ± 0.046	0.975 ± 0.033
SWIM-RBF	0.905 ± 0.045	0.866 ± 0.063	0.961 ± 0.045	0.972 ± 0.053	0.949 ± 0.063	0.973 ± 0.050	0.951 ± 0.026	0.881 ± 0.022	0.996 ± 0.004
LoRAS	0.926 ± 0.042	0.889 ± 0.070	0.978 ± 0.020	0.973 ± 0.054	0.971 ± 0.057	0.975 ± 0.050	0.949 ± 0.053	0.880 ± 0.047	0.996 ± 0.003

Table 3 continued

Method	G-mean	F-measure	AUC	G-mean	F-measure	AUC	G-mean	F-measure	AUC
ASN-SMOTE	0.923 ± 0.028	0.895 ± 0.047	0.978 ± 0.023	0.972 ± 0.053	0.949 ± 0.063	0.973 ± 0.050	0.958 ± 0.063	0.916 ± 0.087	0.996 ± 0.006
Dataset	page-blocks0			Glass0			Glass1		
Random	0.897 ± 0.026	0.736 ± 0.054	0.925 ± 0.021	0.765 ± 0.040	0.692 ± 0.047	0.893 ± 0.032	0.760 ± 0.106	0.716 ± 0.076	0.841 ± 0.116
SMOTE	0.894 ± 0.030	0.728 ± 0.057	0.942 ± 0.013	0.772 ± 0.036	0.702 ± 0.042	0.881 ± 0.039	0.781 ± 0.075	0.730 ± 0.072	0.854 ± 0.107
ADASYN	0.900 ± 0.019	0.709 ± 0.073	0.939 ± 0.014	0.760 ± 0.035	0.696 ± 0.026	0.902 ± 0.025	0.794 ± 0.131	0.756 ± 0.111	0.853 ± 0.125
Borderline1-SMOTE	0.896 ± 0.025	0.723 ± 0.065	0.938 ± 0.013	0.753 ± 0.064	0.694 ± 0.054	0.895 ± 0.035	0.779 ± 0.107	0.734 ± 0.088	0.844 ± 0.138
Borderline2-SMOTE	0.899 ± 0.018	0.695 ± 0.068	0.940 ± 0.015	0.750 ± 0.063	0.688 ± 0.064	0.893 ± 0.038	0.760 ± 0.144	0.718 ± 0.124	0.838 ± 0.155
K-means SMOTE	0.815 ± 0.083	0.724 ± 0.057	0.933 ± 0.029	0.770 ± 0.044	0.695 ± 0.042	0.904 ± 0.041	0.752 ± 0.055	0.691 ± 0.056	0.849 ± 0.097
SVM-SMOTE	0.896 ± 0.030	0.742 ± 0.063	0.940 ± 0.015	0.747 ± 0.056	0.681 ± 0.058	0.891 ± 0.044	0.778 ± 0.122	0.738 ± 0.100	0.845 ± 0.143
SWIM-RBF	0.839 ± 0.059	0.735 ± 0.023	0.941 ± 0.018	0.766 ± 0.040	0.690 ± 0.055	0.895 ± 0.040	0.796 ± 0.104	0.700 ± 0.082	0.841 ± 0.093
LoRAS	0.893 ± 0.030	0.735 ± 0.061	0.936 ± 0.012	0.780 ± 0.040	0.709 ± 0.048	0.893 ± 0.040	0.793 ± 0.092	0.747 ± 0.085	0.864 ± 0.104
ASN-SMOTE	0.896 ± 0.034	0.735 ± 0.069	0.952 ± 0.010	0.780 ± 0.051	0.710 ± 0.065	0.888 ± 0.050	0.791 ± 0.131	0.752 ± 0.110	0.815 ± 0.160
Dataset	Glass4			Glass5			Glass6		
RandomOver	0.851 ± 0.167	0.598 ± 0.287	0.929 ± 0.071	0.860 ± 0.162	0.602 ± 0.342	0.824 ± 0.231	0.887 ± 0.111	0.782 ± 0.205	0.920 ± 0.068
SMOTE	0.848 ± 0.165	0.582 ± 0.280	0.900 ± 0.110	0.910 ± 0.143	0.622 ± 0.310	0.878 ± 0.232	0.875 ± 0.156	0.793 ± 0.220	0.949 ± 0.044
ADASYN	0.853 ± 0.157	0.583 ± 0.279	0.896 ± 0.113	0.910 ± 0.143	0.622 ± 0.310	0.937 ± 0.115	0.917 ± 0.062	0.782 ± 0.188	0.962 ± 0.037
Borderline1-SMOTE	0.861 ± 0.142	0.584 ± 0.276	0.883 ± 0.141	0.854 ± 0.165	0.575 ± 0.331	0.824 ± 0.231	0.859 ± 0.130	0.735 ± 0.241	0.935 ± 0.056
Borderline2-SMOTE	0.850 ± 0.156	0.573 ± 0.281	0.928 ± 0.099	0.793 ± 0.152	0.481 ± 0.241	0.868 ± 0.234	0.886 ± 0.109	0.791 ± 0.202	0.951 ± 0.037
K-means SMOTE	0.749 ± 0.226	0.532 ± 0.345	0.868 ± 0.157	0.466 ± 0.398	nan ± nan	0.822 ± 0.228	0.842 ± 0.146	0.764 ± 0.218	0.928 ± 0.069
SVM-SMOTE	0.857 ± 0.149	0.583 ± 0.277	0.882 ± 0.146	0.795 ± 0.155	0.508 ± 0.266	0.822 ± 0.236	0.851 ± 0.158	0.755 ± 0.234	0.937 ± 0.055
SWIM-RBF	0.853 ± 0.157	0.583 ± 0.279	0.890 ± 0.133	0.902 ± 0.159	0.621 ± 0.311	0.867 ± 0.239	0.898 ± 0.108	0.820 ± 0.191	0.956 ± 0.039
LoRAS	0.853 ± 0.157	0.583 ± 0.279	0.897 ± 0.106	0.910 ± 0.143	0.622 ± 0.310	0.880 ± 0.227	0.936 ± 0.056	0.864 ± 0.186	0.946 ± 0.044
ASN-SMOTE	0.855 ± 0.220	0.605 ± 0.296	0.930 ± 0.106	0.860 ± 0.152	0.602 ± 0.269	0.943 ± 0.112	0.882 ± 0.112	0.798 ± 0.227	0.957 ± 0.052
Dataset	Yeast			Yeast1			Yeast2		
Random	0.682 ± 0.045	0.555 ± 0.050	0.740 ± 0.059	0.744 ± 0.013	0.495 ± 0.012	0.802 ± 0.016	0.871 ± 0.024	0.640 ± 0.039	0.909 ± 0.038
SMOTE	0.685 ± 0.040	0.558 ± 0.045	0.749 ± 0.045	0.737 ± 0.017	0.487 ± 0.027	0.804 ± 0.012	0.877 ± 0.026	0.668 ± 0.015	0.925 ± 0.039
ADASYN	0.671 ± 0.058	0.547 ± 0.062	0.740 ± 0.056	0.721 ± 0.018	0.458 ± 0.022	0.788 ± 0.007	0.872 ± 0.025	0.628 ± 0.032	0.917 ± 0.031
Borderline1-SMOTE	0.683 ± 0.037	0.556 ± 0.042	0.742 ± 0.049	0.746 ± 0.022	0.498 ± 0.032	0.798 ± 0.014	0.884 ± 0.038	0.647 ± 0.046	0.930 ± 0.023
Borderline2-SMOTE	0.689 ± 0.044	0.564 ± 0.047	0.749 ± 0.042	0.741 ± 0.021	0.485 ± 0.023	0.793 ± 0.020	0.885 ± 0.023	0.611 ± 0.027	0.924 ± 0.030
K-means SMOTE	0.628 ± 0.065	0.510 ± 0.082	0.760 ± 0.061	0.735 ± 0.049	0.597 ± 0.059	0.828 ± 0.026	0.852 ± 0.024	0.737 ± 0.042	0.921 ± 0.033
SVM-SMOTE	0.683 ± 0.062	0.555 ± 0.070	0.751 ± 0.058	0.754 ± 0.031	0.517 ± 0.035	0.814 ± 0.020	0.881 ± 0.027	0.673 ± 0.032	0.924 ± 0.029
SWIM-RBF	0.686 ± 0.051	0.560 ± 0.056	0.757 ± 0.061	0.740 ± 0.023	0.498 ± 0.024	0.815 ± 0.022	0.887 ± 0.022	0.661 ± 0.038	0.934 ± 0.017

Table 3 continued

Method	G-mean	F-measure	AUC	G-mean	F-measure	AUC	G-mean	F-measure	AUC
LoRAS	0.692 ± 0.046	0.566 ± 0.052	0.753 ± 0.043	0.739 ± 0.012	0.495 ± 0.017	0.803 ± 0.021	0.885 ± 0.032	0.697 ± 0.020	0.929 ± 0.035
ASN-SMOTE	0.692 ± 0.038	0.568 ± 0.040	0.766 ± 0.049	0.764 ± 0.034	0.547 ± 0.037	0.835 ± 0.020	0.886 ± 0.019	0.699 ± 0.011	0.940 ± 0.024
Dataset	Yeast3			Yeast4			Yeast5		
Random	0.838 ± 0.095	0.687 ± 0.126	0.868 ± 0.089	0.919 ± 0.099	0.774 ± 0.185	0.935 ± 0.093	0.746 ± 0.167	0.528 ± 0.182	0.808 ± 0.129
SMOTE	0.890 ± 0.051	0.728 ± 0.085	0.934 ± 0.043	0.906 ± 0.093	0.683 ± 0.160	0.926 ± 0.091	0.751 ± 0.160	0.341 ± 0.105	0.810 ± 0.125
ADASYN	0.860 ± 0.037	0.670 ± 0.104	0.914 ± 0.037	0.876 ± 0.092	0.553 ± 0.146	0.915 ± 0.097	0.735 ± 0.155	0.277 ± 0.091	0.795 ± 0.125
Borderline1-SMOTE	0.915 ± 0.044	0.733 ± 0.105	0.935 ± 0.031	0.910 ± 0.095	0.795 ± 0.165	0.935 ± 0.094	0.747 ± 0.154	0.332 ± 0.098	0.806 ± 0.120
Borderline2-SMOTE	0.861 ± 0.060	0.599 ± 0.103	0.904 ± 0.049	0.878 ± 0.094	0.659 ± 0.183	0.925 ± 0.100	0.744 ± 0.157	0.316 ± 0.111	0.819 ± 0.087
K-means SMOTE	0.849 ± 0.100	0.757 ± 0.142	0.882 ± 0.093	0.931 ± 0.099	0.872 ± 0.159	0.938 ± 0.092	0.771 ± 0.161	0.517 ± 0.165	0.826 ± 0.121
SVM-SMOTE	0.856 ± 0.068	0.684 ± 0.114	0.900 ± 0.060	0.925 ± 0.100	0.810 ± 0.168	0.935 ± 0.093	0.696 ± 0.149	0.329 ± 0.099	0.787 ± 0.129
SWIM-RBF	0.878 ± 0.028	0.682 ± 0.098	0.927 ± 0.049	0.911 ± 0.096	0.711 ± 0.161	0.930 ± 0.099	0.804 ± 0.163	0.530 ± 0.133	0.827 ± 0.128
LoRAS	0.893 ± 0.041	0.722 ± 0.137	0.928 ± 0.045	0.922 ± 0.096	0.783 ± 0.133	0.936 ± 0.092	0.768 ± 0.113	0.428 ± 0.154	0.833 ± 0.087
ASN-SMOTE	0.909 ± 0.028	0.728 ± 0.073	0.926 ± 0.047	0.918 ± 0.097	0.747 ± 0.137	0.944 ± 0.073	0.785 ± 0.167	0.702 ± 0.168	0.840 ± 0.126
Dataset	Yeast6			Yeast7			Yeast8		
Random	0.745 ± 0.099	0.333 ± 0.073	0.824 ± 0.063	0.867 ± 0.066	0.638 ± 0.123	0.954 ± 0.030	0.740 ± 0.082	0.414 ± 0.054	0.825 ± 0.053
SMOTE	0.823 ± 0.051	0.332 ± 0.074	0.900 ± 0.024	0.929 ± 0.049	0.665 ± 0.120	0.977 ± 0.017	0.825 ± 0.087	0.317 ± 0.035	0.843 ± 0.085
ADASYN	0.811 ± 0.067	0.316 ± 0.062	0.882 ± 0.027	0.938 ± 0.052	0.649 ± 0.093	0.977 ± 0.017	0.785 ± 0.114	0.273 ± 0.037	0.838 ± 0.085
Borderline1-SMOTE	0.817 ± 0.056	0.385 ± 0.061	0.906 ± 0.026	0.938 ± 0.053	0.650 ± 0.098	0.966 ± 0.024	0.840 ± 0.095	0.451 ± 0.101	0.867 ± 0.077
Borderline2-SMOTE	0.818 ± 0.072	0.331 ± 0.042	0.913 ± 0.030	0.921 ± 0.054	0.455 ± 0.054	0.962 ± 0.020	0.856 ± 0.089	0.341 ± 0.130	0.866 ± 0.083
K-means SMOTE	0.437 ± 0.124	0.160 ± 0.071	0.815 ± 0.031	0.930 ± 0.050	0.679 ± 0.119	0.967 ± 0.024	0.775 ± 0.088	0.433 ± 0.062	0.840 ± 0.066
SVM-SMOTE	0.776 ± 0.076	0.378 ± 0.038	0.877 ± 0.042	0.927 ± 0.048	0.637 ± 0.125	0.976 ± 0.018	0.792 ± 0.092	0.438 ± 0.105	0.826 ± 0.057
SWIM-RBF	0.804 ± 0.075	0.315 ± 0.033	0.887 ± 0.045	0.949 ± 0.027	0.621 ± 0.098	0.974 ± 0.019	0.833 ± 0.078	0.365 ± 0.046	0.873 ± 0.074
LoRAS	0.820 ± 0.085	0.326 ± 0.037	0.906 ± 0.034	0.940 ± 0.052	0.669 ± 0.085	0.965 ± 0.021	0.840 ± 0.063	0.309 ± 0.057	0.867 ± 0.073
ASN-SMOTE	0.824 ± 0.084	0.332 ± 0.022	0.862 ± 0.071	0.951 ± 0.027	0.671 ± 0.144	0.977 ± 0.017	0.829 ± 0.072	0.482 ± 0.057	0.841 ± 0.065

bold values indicate the best performances

Table 4 Results obtained by SVM on datasets oversampled by different techniques

Method	G-mean	F-measure	AUC	G-mean	F-measure	AUC	G-mean	F-measure	AUC
Dataset	Haberman			Pima			Car		
Random	0.574 ± 0.106	0.443 ± 0.130	0.632 ± 0.076	0.737 ± 0.037	0.663 ± 0.043	0.740 ± 0.036	0.933 ± 0.072	0.643 ± 0.313	0.936 ± 0.068
SMOTE	0.587 ± 0.097	0.451 ± 0.120	0.634 ± 0.072	0.740 ± 0.046	0.666 ± 0.055	0.742 ± 0.045	0.923 ± 0.083	0.631 ± 0.319	0.927 ± 0.079
ADASYN	0.594 ± 0.094	0.459 ± 0.113	0.640 ± 0.065	0.740 ± 0.042	0.667 ± 0.047	0.742 ± 0.041	0.901 ± 0.107	0.610 ± 0.318	0.908 ± 0.098
Borderline1-SMOTE	0.590 ± 0.108	0.459 ± 0.131	0.643 ± 0.074	0.743 ± 0.040	0.672 ± 0.044	0.744 ± 0.039	0.915 ± 0.098	0.629 ± 0.324	0.920 ± 0.091
Borderline2-SMOTE	0.589 ± 0.099	0.455 ± 0.124	0.636 ± 0.074	0.739 ± 0.026	0.669 ± 0.026	0.742 ± 0.025	0.888 ± 0.085	0.449 ± 0.290	0.893 ± 0.082
K-means SMOTE	0.591 ± 0.115	0.463 ± 0.139	0.645 ± 0.081	0.725 ± 0.042	0.649 ± 0.051	0.730 ± 0.040	0.923 ± 0.083	0.631 ± 0.319	0.927 ± 0.079
SVMSMOTE	0.571 ± 0.086	0.433 ± 0.105	0.623 ± 0.063	0.744 ± 0.027	0.672 ± 0.030	0.746 ± 0.027	0.946 ± 0.066	0.626 ± 0.322	0.949 ± 0.060
SWIM-RBF	0.583 ± 0.090	0.450 ± 0.109	0.634 ± 0.065	0.747 ± 0.047	0.675 ± 0.056	0.749 ± 0.045	0.959 ± 0.047	0.653 ± 0.299	0.961 ± 0.044
LoRAS	0.616 ± 0.062	0.459 ± 0.072	0.650 ± 0.059	0.745 ± 0.050	0.671 ± 0.059	0.746 ± 0.049	0.936 ± 0.068	0.658 ± 0.303	0.939 ± 0.065
ASN-SMOTE	0.615 ± 0.076	0.466 ± 0.085	0.637 ± 0.057	0.748 ± 0.049	0.678 ± 0.053	0.750 ± 0.048	0.974 ± 0.023	0.707 ± 0.170	0.974 ± 0.023
Dataset	Iris			Ecoli			Seeds		
Random	0.929 ± 0.079	0.899 ± 0.108	0.935 ± 0.072	0.931 ± 0.057	0.835 ± 0.124	0.932 ± 0.057	0.835 ± 0.070	0.780 ± 0.087	0.843 ± 0.063
SMOTE	0.935 ± 0.070	0.904 ± 0.100	0.940 ± 0.064	0.931 ± 0.057	0.835 ± 0.124	0.932 ± 0.057	0.844 ± 0.073	0.790 ± 0.091	0.850 ± 0.069
ADASYN	0.900 ± 0.104	0.868 ± 0.129	0.910 ± 0.092	0.890 ± 0.054	0.703 ± 0.119	0.891 ± 0.055	0.820 ± 0.128	0.774 ± 0.134	0.832 ± 0.114
Borderline1-SMOTE	0.905 ± 0.101	0.874 ± 0.125	0.915 ± 0.089	0.910 ± 0.093	0.798 ± 0.184	0.910 ± 0.093	0.799 ± 0.151	0.758 ± 0.139	0.818 ± 0.128
Borderline2-SMOTE	0.912 ± 0.091	0.879 ± 0.118	0.920 ± 0.081	0.894 ± 0.089	0.745 ± 0.178	0.894 ± 0.089	0.818 ± 0.100	0.762 ± 0.116	0.829 ± 0.094
K-means SMOTE	0.935 ± 0.070	0.904 ± 0.100	0.940 ± 0.064	0.925 ± 0.075	0.840 ± 0.140	0.926 ± 0.074	0.872 ± 0.069	0.820 ± 0.092	0.875 ± 0.069
SVMSMOTE	0.899 ± 0.106	0.868 ± 0.130	0.910 ± 0.093	0.899 ± 0.100	0.769 ± 0.189	0.900 ± 0.101	0.824 ± 0.084	0.771 ± 0.101	0.832 ± 0.077
SWIM-RBF	0.929 ± 0.079	0.899 ± 0.108	0.935 ± 0.072	0.922 ± 0.069	0.812 ± 0.145	0.923 ± 0.069	0.857 ± 0.054	0.802 ± 0.074	0.861 ± 0.052
LoRAS	0.935 ± 0.070	0.904 ± 0.100	0.940 ± 0.064	0.929 ± 0.064	0.834 ± 0.137	0.930 ± 0.064	0.860 ± 0.050	0.806 ± 0.070	0.864 ± 0.047
ASN-SMOTE	0.935 ± 0.070	0.904 ± 0.100	0.940 ± 0.064	0.933 ± 0.062	0.846 ± 0.132	0.933 ± 0.062	0.865 ± 0.052	0.811 ± 0.071	0.868 ± 0.051
Dataset	Wine			Dermatology-6			Newthyroid2		
Random	0.970 ± 0.011	0.958 ± 0.002	0.971 ± 0.010	0.962 ± 0.058	0.825 ± 0.156	0.963 ± 0.056	0.982 ± 0.029	0.971 ± 0.035	0.983 ± 0.028
SMOTE	0.970 ± 0.017	0.951 ± 0.030	0.971 ± 0.017	0.962 ± 0.058	0.825 ± 0.156	0.963 ± 0.056	0.982 ± 0.029	0.971 ± 0.035	0.983 ± 0.028
ADASYN	0.979 ± 0.019	0.960 ± 0.036	0.979 ± 0.019	0.963 ± 0.059	0.840 ± 0.150	0.965 ± 0.056	0.982 ± 0.029	0.971 ± 0.035	0.983 ± 0.028
Borderline1-SMOTE	0.975 ± 0.016	0.959 ± 0.024	0.975 ± 0.016	0.937 ± 0.115	0.871 ± 0.127	0.946 ± 0.098	0.982 ± 0.029	0.971 ± 0.035	0.983 ± 0.028
Borderline2-SMOTE	0.979 ± 0.027	0.962 ± 0.047	0.979 ± 0.026	0.934 ± 0.114	0.844 ± 0.151	0.943 ± 0.097	0.977 ± 0.017	0.903 ± 0.069	0.978 ± 0.017
K-means SMOTE	0.974 ± 0.021	0.967 ± 0.029	0.975 ± 0.020	0.953 ± 0.056	0.757 ± 0.199	0.954 ± 0.054	0.982 ± 0.029	0.971 ± 0.035	0.983 ± 0.028
SVMSMOTE	0.975 ± 0.016	0.959 ± 0.024	0.975 ± 0.016	0.963 ± 0.059	0.840 ± 0.150	0.965 ± 0.056	0.982 ± 0.029	0.971 ± 0.035	0.983 ± 0.028
SWIM-RBF	0.983 ± 0.016	0.975 ± 0.020	0.983 ± 0.016	0.965 ± 0.059	0.858 ± 0.149	0.966 ± 0.057	0.982 ± 0.029	0.971 ± 0.035	0.983 ± 0.028
LoRAS	0.979 ± 0.019	0.968 ± 0.029	0.979 ± 0.019	0.962 ± 0.058	0.825 ± 0.156	0.963 ± 0.056	0.982 ± 0.029	0.971 ± 0.035	0.983 ± 0.028
ASN-SMOTE	0.974 ± 0.021	0.967 ± 0.029	0.975 ± 0.020	0.965 ± 0.059	0.858 ± 0.149	0.966 ± 0.057	0.989 ± 0.017	0.951 ± 0.069	0.989 ± 0.016

Table 4 continued

Method	G-mean	F-measure	AUC	G-mean	F-measure	AUC	G-mean	F-measure	AUC
Dataset	page-blocks0			Glass0			Glass1		
Random	0.601 ± 0.182	0.279 ± 0.080	0.664 ± 0.134	0.684 ± 0.172	0.675 ± 0.135	0.736 ± 0.128	0.492 ± 0.186	0.562 ± 0.096	0.586 ± 0.117
SMOTE	0.628 ± 0.175	0.293 ± 0.085	0.685 ± 0.131	0.739 ± 0.127	0.697 ± 0.129	0.764 ± 0.109	0.490 ± 0.241	0.578 ± 0.160	0.592 ± 0.163
ADASYN	0.684 ± 0.128	0.318 ± 0.077	0.723 ± 0.100	0.690 ± 0.193	0.693 ± 0.155	0.751 ± 0.144	0.443 ± 0.282	0.578 ± 0.130	0.594 ± 0.141
Borderline1-SMOTE	0.711 ± 0.112	0.335 ± 0.080	0.741 ± 0.093	0.684 ± 0.193	0.690 ± 0.156	0.748 ± 0.143	0.429 ± 0.204	0.536 ± 0.072	0.555 ± 0.101
Borderline2-SMOTE	0.718 ± 0.156	0.384 ± 0.171	0.752 ± 0.127	0.659 ± 0.213	0.681 ± 0.163	0.734 ± 0.155	0.397 ± 0.348	0.590 ± 0.143	0.600 ± 0.157
K-means SMOTE	0.410 ± 0.252	0.187 ± 0.147	0.592 ± 0.128	0.692 ± 0.028	0.617 ± 0.043	0.712 ± 0.039	0.510 ± 0.261	0.586 ± 0.168	0.605 ± 0.178
SVM-SMOTE	0.627 ± 0.170	0.292 ± 0.082	0.685 ± 0.125	0.701 ± 0.150	0.687 ± 0.128	0.751 ± 0.112	0.474 ± 0.197	0.558 ± 0.100	0.579 ± 0.121
SWIM-RBF	0.301 ± 0.107	0.144 ± 0.070	0.539 ± 0.023	0.716 ± 0.147	0.697 ± 0.129	0.761 ± 0.111	0.513 ± 0.203	0.583 ± 0.122	0.606 ± 0.131
LoRAS	0.623 ± 0.173	0.290 ± 0.081	0.681 ± 0.127	0.745 ± 0.138	0.699 ± 0.143	0.767 ± 0.125	0.505 ± 0.216	0.589 ± 0.142	0.604 ± 0.149
ASN-SMOTE	0.606 ± 0.219	0.353 ± 0.155	0.685 ± 0.163	0.734 ± 0.132	0.701 ± 0.125	0.768 ± 0.105	0.516 ± 0.221	0.584 ± 0.147	0.605 ± 0.156
Dataset	Glass4			Glass5			Glass6		
Random	0.859 ± 0.149	0.602 ± 0.279	0.879 ± 0.118	0.900 ± 0.138	0.518 ± 0.239	0.915 ± 0.110	0.915 ± 0.063	0.813 ± 0.149	0.920 ± 0.057
SMOTE	0.874 ± 0.129	0.616 ± 0.273	0.889 ± 0.107	0.900 ± 0.138	0.518 ± 0.239	0.915 ± 0.110	0.915 ± 0.065	0.809 ± 0.144	0.920 ± 0.060
ADASYN	0.874 ± 0.129	0.616 ± 0.273	0.889 ± 0.107	0.897 ± 0.157	0.555 ± 0.254	0.915 ± 0.122	0.903 ± 0.068	0.777 ± 0.172	0.909 ± 0.063
Borderline1-SMOTE	0.881 ± 0.116	0.619 ± 0.269	0.894 ± 0.099	0.871 ± 0.177	0.468 ± 0.221	0.895 ± 0.131	0.887 ± 0.099	0.787 ± 0.219	0.899 ± 0.086
Borderline2-SMOTE	0.845 ± 0.161	0.577 ± 0.280	0.869 ± 0.124	0.848 ± 0.165	0.354 ± 0.153	0.873 ± 0.120	0.918 ± 0.067	0.816 ± 0.144	0.923 ± 0.062
K-means SMOTE	0.833 ± 0.211	0.614 ± 0.301	0.837 ± 0.209	0.842 ± 0.168	0.521 ± 0.248	0.867 ± 0.135	0.864 ± 0.146	0.782 ± 0.170	0.884 ± 0.112
SVM-SMOTE	0.853 ± 0.166	0.612 ± 0.281	0.877 ± 0.128	0.850 ± 0.156	0.522 ± 0.246	0.872 ± 0.128	0.872 ± 0.154	0.778 ± 0.207	0.892 ± 0.120
SWIM-RBF	0.859 ± 0.148	0.600 ± 0.276	0.879 ± 0.118	0.895 ± 0.135	0.477 ± 0.208	0.910 ± 0.108	0.915 ± 0.065	0.809 ± 0.144	0.920 ± 0.060
LoRAS	0.874 ± 0.129	0.616 ± 0.273	0.889 ± 0.107	0.905 ± 0.140	0.556 ± 0.252	0.920 ± 0.112	0.915 ± 0.068	0.810 ± 0.155	0.920 ± 0.063
ASN-SMOTE	0.848 ± 0.240	0.613 ± 0.322	0.853 ± 0.234	0.904 ± 0.130	0.518 ± 0.238	0.917 ± 0.105	0.917 ± 0.065	0.824 ± 0.150	0.923 ± 0.059
Dataset	Yeast			Yeast1			yeast2		
Random	0.706 ± 0.041	0.583 ± 0.046	0.707 ± 0.042	0.781 ± 0.037	0.586 ± 0.049	0.785 ± 0.034	0.924 ± 0.010	0.747 ± 0.016	0.924 ± 0.010
SMOTE	0.710 ± 0.027	0.588 ± 0.030	0.712 ± 0.028	0.780 ± 0.037	0.578 ± 0.048	0.783 ± 0.035	0.921 ± 0.022	0.759 ± 0.018	0.922 ± 0.021
ADASYN	0.697 ± 0.020	0.584 ± 0.019	0.709 ± 0.020	0.765 ± 0.047	0.508 ± 0.063	0.768 ± 0.047	0.924 ± 0.019	0.726 ± 0.018	0.925 ± 0.018
Borderline1-SMOTE	0.699 ± 0.023	0.587 ± 0.022	0.713 ± 0.022	0.772 ± 0.045	0.515 ± 0.057	0.775 ± 0.046	0.925 ± 0.016	0.729 ± 0.034	0.925 ± 0.016
Borderline2-SMOTE	0.693 ± 0.023	0.586 ± 0.016	0.710 ± 0.016	0.771 ± 0.031	0.513 ± 0.043	0.774 ± 0.032	0.912 ± 0.012	0.657 ± 0.039	0.912 ± 0.012
K-means SMOTE	0.595 ± 0.065	0.485 ± 0.077	0.654 ± 0.043	0.763 ± 0.031	0.629 ± 0.027	0.779 ± 0.023	0.868 ± 0.026	0.768 ± 0.035	0.874 ± 0.023

Table 4 continued

Method	G-mean	F-measure	AUC	G-mean	F-measure	AUC	G-mean	F-measure	AUC
SVM-SMOTE	0.697 ± 0.020	0.584 ± 0.019	0.709 ± 0.020	0.785 ± 0.046	0.575 ± 0.062	0.788 ± 0.044	0.925 ± 0.017	0.751 ± 0.022	0.925 ± 0.017
SWIM-RBF	0.711 ± 0.045	0.588 ± 0.051	0.712 ± 0.045	0.784 ± 0.031	0.586 ± 0.042	0.787 ± 0.028	0.926 ± 0.018	0.746 ± 0.005	0.927 ± 0.017
LoRAS	0.709 ± 0.032	0.586 ± 0.035	0.710 ± 0.032	0.785 ± 0.045	0.592 ± 0.054	0.789 ± 0.040	0.931 ± 0.017	0.770 ± 0.014	0.931 ± 0.016
ASN-SMOTE	0.710 ± 0.022	0.590 ± 0.022	0.714 ± 0.022	0.776 ± 0.044	0.597 ± 0.053	0.783 ± 0.038	0.916 ± 0.019	0.775 ± 0.038	0.917 ± 0.019
Dataset	Yeast3			Yeast4			Yeast5		
Random	0.885 ± 0.083	0.694 ± 0.097	0.891 ± 0.072	0.918 ± 0.097	0.747 ± 0.139	0.923 ± 0.089	0.713 ± 0.157	0.516 ± 0.192	0.760 ± 0.122
SMOTE	0.889 ± 0.084	0.710 ± 0.079	0.895 ± 0.072	0.923 ± 0.099	0.791 ± 0.165	0.928 ± 0.090	0.716 ± 0.155	0.528 ± 0.140	0.764 ± 0.118
ADASYN	0.846 ± 0.091	0.593 ± 0.106	0.853 ± 0.083	0.846 ± 0.070	0.447 ± 0.072	0.854 ± 0.064	0.704 ± 0.077	0.202 ± 0.033	0.720 ± 0.073
Borderline1-SMOTE	0.883 ± 0.087	0.650 ± 0.099	0.889 ± 0.077	0.922 ± 0.099	0.776 ± 0.151	0.927 ± 0.090	0.721 ± 0.162	0.339 ± 0.161	0.755 ± 0.131
Borderline2-SMOTE	0.873 ± 0.048	0.563 ± 0.087	0.876 ± 0.047	0.907 ± 0.102	0.680 ± 0.172	0.911 ± 0.095	0.696 ± 0.157	0.220 ± 0.086	0.725 ± 0.136
K-means SMOTE	0.870 ± 0.103	0.765 ± 0.111	0.881 ± 0.084	0.915 ± 0.096	0.838 ± 0.172	0.920 ± 0.087	0.723 ± 0.158	0.658 ± 0.155	0.774 ± 0.121
SVM-SMOTE	0.885 ± 0.082	0.693 ± 0.090	0.891 ± 0.071	0.919 ± 0.097	0.759 ± 0.149	0.924 ± 0.089	0.708 ± 0.157	0.465 ± 0.195	0.753 ± 0.123
SWIM-RBF	0.884 ± 0.081	0.693 ± 0.083	0.891 ± 0.070	0.922 ± 0.099	0.779 ± 0.159	0.927 ± 0.090	0.717 ± 0.157	0.538 ± 0.170	0.765 ± 0.121
LoRAS	0.888 ± 0.085	0.705 ± 0.087	0.894 ± 0.074	0.921 ± 0.098	0.777 ± 0.149	0.927 ± 0.089	0.721 ± 0.156	0.621 ± 0.132	0.772 ± 0.119
ASN-SMOTE	0.889 ± 0.085	0.711 ± 0.087	0.895 ± 0.074	0.925 ± 0.099	0.808 ± 0.162	0.930 ± 0.090	0.723 ± 0.158	0.658 ± 0.155	0.774 ± 0.121
Dataset	Yeast6			Yeast7			Yeast8		
Random	0.809 ± 0.051	0.291 ± 0.059	0.816 ± 0.041	0.966 ± 0.009	0.488 ± 0.091	0.966 ± 0.009	0.848 ± 0.075	0.285 ± 0.042	0.853 ± 0.068
SMOTE	0.828 ± 0.063	0.325 ± 0.067	0.835 ± 0.053	0.966 ± 0.010	0.491 ± 0.090	0.967 ± 0.010	0.857 ± 0.078	0.332 ± 0.051	0.863 ± 0.070
ADASYN	0.817 ± 0.056	0.305 ± 0.079	0.825 ± 0.046	0.967 ± 0.010	0.500 ± 0.097	0.968 ± 0.009	0.833 ± 0.066	0.236 ± 0.023	0.839 ± 0.060
Borderline1-SMOTE	0.828 ± 0.062	0.390 ± 0.054	0.836 ± 0.052	0.966 ± 0.009	0.493 ± 0.089	0.967 ± 0.009	0.884 ± 0.080	0.441 ± 0.103	0.890 ± 0.072
Borderline2-SMOTE	0.798 ± 0.058	0.331 ± 0.054	0.807 ± 0.048	0.951 ± 0.015	0.406 ± 0.098	0.952 ± 0.014	0.883 ± 0.089	0.345 ± 0.115	0.888 ± 0.082
K-means SMOTE	0.518 ± 0.158	0.153 ± 0.083	0.613 ± 0.096	0.967 ± 0.010	0.500 ± 0.097	0.968 ± 0.010	0.869 ± 0.069	0.426 ± 0.109	0.875 ± 0.060
SVM-SMOTE	0.800 ± 0.039	0.398 ± 0.048	0.812 ± 0.029	0.967 ± 0.009	0.495 ± 0.087	0.967 ± 0.009	0.855 ± 0.070	0.426 ± 0.108	0.862 ± 0.062
SWIM-RBF	0.827 ± 0.059	0.295 ± 0.057	0.834 ± 0.050	0.964 ± 0.011	0.475 ± 0.097	0.964 ± 0.010	0.872 ± 0.085	0.283 ± 0.062	0.877 ± 0.079
LoRAS	0.823 ± 0.059	0.315 ± 0.073	0.830 ± 0.049	0.972 ± 0.008	0.536 ± 0.092	0.972 ± 0.008	0.878 ± 0.086	0.306 ± 0.064	0.882 ± 0.079
ASN-SMOTE	0.824 ± 0.061	0.361 ± 0.057	0.830 ± 0.054	0.973 ± 0.007	0.539 ± 0.079	0.973 ± 0.007	0.882 ± 0.080	0.427 ± 0.116	0.888 ± 0.072

bold values indicate the best performances

Table 5 Results obtained by RF on datasets oversampled by different techniques

Method	G-mean	F-measure	AUC	G-mean	F-measure	AUC	G-mean	F-measure	AUC
Dataset	Haberman			Pima			Car		
Random	0.626 ± 0.066	0.482 ± 0.085	0.720 ± 0.055	0.729 ± 0.040	0.654 ± 0.050	0.825 ± 0.036	0.593 ± 0.484	0.475 ± 0.406	0.968 ± 0.064
SMOTE	0.598 ± 0.108	0.444 ± 0.121	0.702 ± 0.070	0.739 ± 0.026	0.666 ± 0.032	0.823 ± 0.041	0.506 ± 0.292	0.313 ± 0.225	0.858 ± 0.074
ADASYN	0.605 ± 0.072	0.446 ± 0.079	0.690 ± 0.049	0.738 ± 0.034	0.664 ± 0.043	0.823 ± 0.040	0.792 ± 0.396	0.667 ± 0.368	0.961 ± 0.057
Borderline1-SMOTE	0.626 ± 0.043	0.475 ± 0.047	0.680 ± 0.056	0.748 ± 0.046	0.677 ± 0.056	0.820 ± 0.038	0.792 ± 0.396	0.667 ± 0.368	0.970 ± 0.043
Borderline2-SMOTE	0.630 ± 0.095	0.473 ± 0.104	0.710 ± 0.037	0.749 ± 0.043	0.677 ± 0.053	0.823 ± 0.036	0.779 ± 0.242	0.448 ± 0.361	0.897 ± 0.095
K-means SMOTE	0.574 ± 0.104	0.434 ± 0.127	0.729 ± 0.082	0.731 ± 0.042	0.658 ± 0.053	0.823 ± 0.034	0.637 ± 0.337	0.456 ± 0.366	0.993 ± 0.010
SVMSMOTE	0.619 ± 0.088	0.472 ± 0.104	0.705 ± 0.072	0.735 ± 0.037	0.662 ± 0.044	0.826 ± 0.046	0.962 ± 0.033	0.661 ± 0.277	0.974 ± 0.032
SWIM-RBF	0.639 ± 0.068	0.491 ± 0.081	0.723 ± 0.076	0.750 ± 0.039	0.679 ± 0.048	0.828 ± 0.034	0.569 ± 0.466	0.456 ± 0.378	0.979 ± 0.042
LoRAS	0.603 ± 0.097	0.457 ± 0.113	0.729 ± 0.063	0.745 ± 0.034	0.674 ± 0.044	0.829 ± 0.038	0.468 ± 0.438	0.304 ± 0.287	0.981 ± 0.036
ASN-SMOTE	0.644 ± 0.089	0.498 ± 0.101	0.737 ± 0.061	0.752 ± 0.038	0.681 ± 0.045	0.820 ± 0.039	0.687 ± 0.370	0.603 ± 0.334	0.980 ± 0.035
Dataset	Iris			Ecoli			Seeds		
RandomOver	0.933 ± 0.068	0.910 ± 0.092	0.988 ± 0.015	0.883 ± 0.096	0.791 ± 0.187	0.934 ± 0.102	0.869 ± 0.084	0.830 ± 0.115	0.961 ± 0.039
SMOTE	0.954 ± 0.042	0.940 ± 0.050	0.990 ± 0.013	0.890 ± 0.109	0.804 ± 0.202	0.946 ± 0.082	0.865 ± 0.097	0.825 ± 0.131	0.966 ± 0.038
ADASYN	0.960 ± 0.034	0.942 ± 0.048	0.989 ± 0.014	0.858 ± 0.082	0.722 ± 0.176	0.942 ± 0.063	0.865 ± 0.108	0.822 ± 0.142	0.961 ± 0.046
Borderline1-SMOTE	0.954 ± 0.042	0.940 ± 0.050	0.990 ± 0.013	0.821 ± 0.154	0.726 ± 0.240	0.923 ± 0.121	0.884 ± 0.103	0.845 ± 0.138	0.967 ± 0.041
Borderline2-SMOTE	0.960 ± 0.034	0.942 ± 0.048	0.990 ± 0.013	0.824 ± 0.161	0.713 ± 0.247	0.912 ± 0.118	0.877 ± 0.096	0.832 ± 0.125	0.965 ± 0.039
K-means SMOTE	0.958 ± 0.039	0.949 ± 0.046	0.990 ± 0.013	0.873 ± 0.087	0.809 ± 0.179	0.949 ± 0.082	0.881 ± 0.077	0.841 ± 0.106	0.965 ± 0.035
SVMSMOTE	0.939 ± 0.066	0.912 ± 0.092	0.990 ± 0.013	0.872 ± 0.120	0.762 ± 0.218	0.932 ± 0.101	0.858 ± 0.091	0.813 ± 0.122	0.966 ± 0.039
SWIM-RBF	0.949 ± 0.048	0.930 ± 0.061	0.991 ± 0.011	0.908 ± 0.103	0.802 ± 0.198	0.943 ± 0.079	0.883 ± 0.100	0.845 ± 0.132	0.970 ± 0.035
LoRAS	0.954 ± 0.042	0.940 ± 0.060	0.990 ± 0.013	0.887 ± 0.106	0.796 ± 0.198	0.934 ± 0.108	0.877 ± 0.082	0.838 ± 0.114	0.967 ± 0.032
ASN-SMOTE	0.960 ± 0.034	0.942 ± 0.048	0.990 ± 0.013	0.886 ± 0.089	0.809 ± 0.185	0.938 ± 0.082	0.869 ± 0.084	0.828 ± 0.115	0.968 ± 0.036
Dataset	Wine			Dermatology6			Newthyroid2		
Random	0.987 ± 0.017	0.983 ± 0.020	0.999 ± 0.003	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	0.966 ± 0.061	0.953 ± 0.065	0.998 ± 0.003
SMOTE	0.987 ± 0.017	0.983 ± 0.020	0.999 ± 0.001	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	0.970 ± 0.036	0.969 ± 0.038	1.000 ± 0.000
ADASYN	0.979 ± 0.019	0.975 ± 0.021	0.999 ± 0.001	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	0.949 ± 0.058	0.925 ± 0.053	0.998 ± 0.003
Borderline1-SMOTE	0.987 ± 0.017	0.983 ± 0.020	0.999 ± 0.003	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	0.954 ± 0.062	0.951 ± 0.066	0.998 ± 0.003
Borderline2-SMOTE	0.991 ± 0.017	0.985 ± 0.031	0.999 ± 0.003	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	0.983 ± 0.017	0.926 ± 0.069	0.998 ± 0.005
K-means SMOTE	0.987 ± 0.017	0.983 ± 0.020	0.999 ± 0.003	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	0.969 ± 0.062	0.967 ± 0.067	1.000 ± 0.000
SVMSMOTE	0.987 ± 0.017	0.983 ± 0.020	0.999 ± 0.003	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	0.980 ± 0.029	0.960 ± 0.052	0.999 ± 0.002
SWIM-RBF	0.996 ± 0.008	0.992 ± 0.016	0.999 ± 0.003	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	0.994 ± 0.007	0.973 ± 0.033	0.999 ± 0.002
LoRAS	0.987 ± 0.017	0.983 ± 0.020	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	0.985 ± 0.030	0.985 ± 0.031	1.000 ± 0.000
ASN-SMOTE	0.992 ± 0.010	0.984 ± 0.020	0.999 ± 0.001	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	0.992 ± 0.011	0.962 ± 0.050	0.999 ± 0.002

Table 5 continued

Method	G-mean	F-measure	AUC	G-mean	F-measure	AUC	G-mean	F-measure	AUC
Dataset	Pageblocks0			Glass0			Glass1		
Random	0.906 ± 0.031	0.827 ± 0.032	0.987 ± 0.005	0.849 ± 0.030	0.794 ± 0.037	0.936 ± 0.027	0.754 ± 0.073	0.705 ± 0.057	0.846 ± 0.073
SMOTE	0.915 ± 0.032	0.823 ± 0.048	0.988 ± 0.007	0.850 ± 0.025	0.796 ± 0.044	0.935 ± 0.029	0.777 ± 0.093	0.736 ± 0.103	0.843 ± 0.073
ADASYN	0.929 ± 0.027	0.813 ± 0.063	0.986 ± 0.007	0.846 ± 0.047	0.789 ± 0.064	0.927 ± 0.031	0.752 ± 0.091	0.702 ± 0.098	0.845 ± 0.077
Borderline1-SMOTE	0.923 ± 0.033	0.823 ± 0.054	0.986 ± 0.006	0.856 ± 0.057	0.804 ± 0.075	0.933 ± 0.027	0.789 ± 0.072	0.744 ± 0.061	0.862 ± 0.083
Borderline2-SMOTE	0.918 ± 0.034	0.778 ± 0.083	0.982 ± 0.009	0.840 ± 0.029	0.782 ± 0.045	0.938 ± 0.031	0.771 ± 0.098	0.729 ± 0.097	0.835 ± 0.102
K-means SMOTE	0.884 ± 0.066	0.815 ± 0.069	0.988 ± 0.007	0.850 ± 0.032	0.805 ± 0.031	0.931 ± 0.032	0.777 ± 0.070	0.730 ± 0.078	0.828 ± 0.063
SVM-SMOTE	0.916 ± 0.037	0.801 ± 0.045	0.988 ± 0.004	0.848 ± 0.035	0.793 ± 0.053	0.934 ± 0.023	0.784 ± 0.087	0.739 ± 0.086	0.856 ± 0.075
SWIM-RBF	0.895 ± 0.060	0.806 ± 0.064	0.988 ± 0.007	0.837 ± 0.035	0.784 ± 0.054	0.924 ± 0.032	0.775 ± 0.071	0.730 ± 0.075	0.828 ± 0.087
LoRAS	0.921 ± 0.042	0.821 ± 0.060	0.987 ± 0.008	0.838 ± 0.046	0.779 ± 0.058	0.921 ± 0.040	0.772 ± 0.069	0.725 ± 0.075	0.840 ± 0.050
ASN-SMOTE	0.917 ± 0.044	0.765 ± 0.106	0.984 ± 0.010	0.864 ± 0.043	0.810 ± 0.053	0.928 ± 0.027	0.794 ± 0.085	0.751 ± 0.082	0.865 ± 0.068
Dataset	Glass4			Glass5			Glass6		
Random	0.811 ± 0.169	0.678 ± 0.263	0.907 ± 0.144	0.895 ± 0.126	0.716 ± 0.328	1.000 ± 0.000	0.910 ± 0.106	0.843 ± 0.129	0.984 ± 0.015
SMOTE	0.801 ± 0.177	0.640 ± 0.318	0.930 ± 0.105	0.754 ± 0.386	0.582 ± 0.438	0.944 ± 0.106	0.933 ± 0.070	0.866 ± 0.102	0.992 ± 0.009
ADASYN	0.784 ± 0.195	0.632 ± 0.330	0.929 ± 0.101	0.754 ± 0.386	0.582 ± 0.438	0.956 ± 0.082	0.925 ± 0.071	0.827 ± 0.113	0.985 ± 0.016
Borderline1-SMOTE	0.859 ± 0.159	0.691 ± 0.247	0.984 ± 0.022	0.695 ± 0.366	0.516 ± 0.392	0.993 ± 0.010	0.910 ± 0.106	0.843 ± 0.129	0.990 ± 0.010
Borderline2-SMOTE	0.920 ± 0.078	0.715 ± 0.256	0.973 ± 0.037	0.671 ± 0.363	0.509 ± 0.398	0.920 ± 0.155	0.898 ± 0.105	0.811 ± 0.170	0.979 ± 0.019
K-means SMOTE	0.730 ± 0.159	0.613 ± 0.211	0.915 ± 0.121	0.819 ± 0.173	0.407 ± 0.180	0.885 ± 0.200	0.904 ± 0.105	0.879 ± 0.112	0.994 ± 0.009
SVM-SMOTE	0.828 ± 0.157	0.670 ± 0.290	0.988 ± 0.014	0.695 ± 0.366	0.516 ± 0.392	0.939 ± 0.116	0.910 ± 0.106	0.843 ± 0.129	0.984 ± 0.015
SWIM-RBF	0.867 ± 0.162	0.691 ± 0.300	0.938 ± 0.113	0.954 ± 0.087	0.782 ± 0.344	0.961 ± 0.078	0.933 ± 0.072	0.870 ± 0.126	0.973 ± 0.042
LoRAS	0.834 ± 0.147	0.637 ± 0.298	0.958 ± 0.052	0.698 ± 0.368	0.556 ± 0.427	0.937 ± 0.127	0.935 ± 0.072	0.879 ± 0.092	0.988 ± 0.013
ASN-SMOTE	0.865 ± 0.161	0.669 ± 0.291	0.976 ± 0.022	0.668 ± 0.364	0.509 ± 0.399	0.995 ± 0.010	0.938 ± 0.067	0.893 ± 0.070	0.989 ± 0.013
Dataset	Yeast			Yeast1			Yeast2		
Random	0.713 ± 0.050	0.601 ± 0.063	0.781 ± 0.057	0.730 ± 0.055	0.591 ± 0.068	0.841 ± 0.044	0.906 ± 0.019	0.790 ± 0.025	0.967 ± 0.012
SMOTE	0.715 ± 0.049	0.602 ± 0.062	0.782 ± 0.055	0.744 ± 0.047	0.605 ± 0.048	0.849 ± 0.035	0.909 ± 0.021	0.791 ± 0.030	0.965 ± 0.016

Table 5 continued

Method	G-mean	F-measure	AUC	G-mean	F-measure	AUC	G-mean	F-measure	AUC
ADASYN	0.708 ± 0.051	0.591 ± 0.062	0.778 ± 0.059	0.731 ± 0.046	0.543 ± 0.052	0.838 ± 0.036	0.908 ± 0.028	0.775 ± 0.037	0.967 ± 0.014
Borderline1-SMOTE	0.711 ± 0.053	0.596 ± 0.065	0.776 ± 0.053	0.735 ± 0.042	0.566 ± 0.048	0.839 ± 0.039	0.919 ± 0.030	0.794 ± 0.050	0.967 ± 0.011
Borderline2-SMOTE	0.716 ± 0.045	0.597 ± 0.055	0.773 ± 0.051	0.755 ± 0.026	0.564 ± 0.033	0.833 ± 0.031	0.914 ± 0.025	0.735 ± 0.023	0.966 ± 0.007
K-means SMOTE	0.645 ± 0.044	0.541 ± 0.063	0.784 ± 0.055	0.710 ± 0.049	0.606 ± 0.064	0.843 ± 0.052	0.863 ± 0.028	0.755 ± 0.039	0.965 ± 0.006
SVMSMOTE	0.715 ± 0.045	0.604 ± 0.059	0.778 ± 0.062	0.743 ± 0.055	0.590 ± 0.056	0.845 ± 0.042	0.920 ± 0.018	0.798 ± 0.022	0.962 ± 0.013
SWIM-RBF	0.677 ± 0.054	0.572 ± 0.071	0.780 ± 0.067	0.699 ± 0.053	0.584 ± 0.063	0.841 ± 0.043	0.877 ± 0.033	0.769 ± 0.039	0.967 ± 0.013
LoRAS	0.659 ± 0.040	0.555 ± 0.061	0.780 ± 0.058	0.680 ± 0.051	0.576 ± 0.062	0.839 ± 0.051	0.857 ± 0.029	0.766 ± 0.036	0.967 ± 0.016
ASN-SMOTE	0.716 ± 0.049	0.604 ± 0.063	0.784 ± 0.054	0.757 ± 0.047	0.591 ± 0.045	0.855 ± 0.034	0.887 ± 0.039	0.779 ± 0.052	0.974 ± 0.008
Dataset	Yeast3			Yeast4			Yeast5		
Random	0.852 ± 0.114	0.725 ± 0.106	0.987 ± 0.006	0.887 ± 0.072	0.828 ± 0.137	0.941 ± 0.081	0.722 ± 0.156	0.640 ± 0.131	0.928 ± 0.036
SMOTE	0.884 ± 0.087	0.749 ± 0.063	0.986 ± 0.010	0.895 ± 0.081	0.776 ± 0.124	0.952 ± 0.066	0.749 ± 0.162	0.592 ± 0.117	0.836 ± 0.101
ADASYN	0.872 ± 0.102	0.742 ± 0.106	0.986 ± 0.008	0.881 ± 0.069	0.669 ± 0.054	0.947 ± 0.050	0.749 ± 0.164	0.579 ± 0.144	0.842 ± 0.076
Borderline1-SMOTE	0.877 ± 0.105	0.761 ± 0.091	0.980 ± 0.014	0.903 ± 0.083	0.851 ± 0.129	0.969 ± 0.040	0.718 ± 0.152	0.580 ± 0.111	0.864 ± 0.076
Borderline2-SMOTE	0.876 ± 0.107	0.719 ± 0.116	0.977 ± 0.017	0.901 ± 0.085	0.830 ± 0.146	0.931 ± 0.089	0.674 ± 0.174	0.473 ± 0.105	0.825 ± 0.100
K-means SMOTE	0.851 ± 0.084	0.768 ± 0.100	0.985 ± 0.009	0.904 ± 0.084	0.864 ± 0.132	0.963 ± 0.062	0.723 ± 0.158	0.658 ± 0.155	0.838 ± 0.123
SVMSMOTE	0.874 ± 0.103	0.744 ± 0.096	0.985 ± 0.013	0.901 ± 0.085	0.830 ± 0.146	0.964 ± 0.056	0.716 ± 0.154	0.514 ± 0.113	0.847 ± 0.078
SWIM-RBF	0.877 ± 0.084	0.758 ± 0.094	0.984 ± 0.010	0.914 ± 0.095	0.825 ± 0.161	0.948 ± 0.062	0.723 ± 0.158	0.658 ± 0.155	0.900 ± 0.065
LoRAS	0.862 ± 0.075	0.769 ± 0.097	0.985 ± 0.009	0.905 ± 0.084	0.877 ± 0.134	0.961 ± 0.051	0.723 ± 0.158	0.658 ± 0.155	0.897 ± 0.073
ASN-SMOTE	0.888 ± 0.071	0.770 ± 0.051	0.986 ± 0.009	0.917 ± 0.094	0.857 ± 0.156	0.963 ± 0.065	0.723 ± 0.158	0.658 ± 0.155	0.856 ± 0.083
Dataset	Yeast6			Yeast7			Yeast8		
Random	0.562 ± 0.128	0.387 ± 0.155	0.931 ± 0.045	0.908 ± 0.059	0.731 ± 0.070	0.993 ± 0.006	0.767 ± 0.069	0.582 ± 0.086	0.890 ± 0.084
SMOTE	0.674 ± 0.100	0.323 ± 0.036	0.926 ± 0.020	0.919 ± 0.068	0.735 ± 0.083	0.992 ± 0.007	0.782 ± 0.073	0.532 ± 0.085	0.911 ± 0.053
ADASYN	0.664 ± 0.107	0.345 ± 0.068	0.928 ± 0.023	0.919 ± 0.068	0.735 ± 0.083	0.992 ± 0.007	0.780 ± 0.072	0.512 ± 0.114	0.879 ± 0.062
Borderline1-SMOTE	0.673 ± 0.056	0.390 ± 0.034	0.938 ± 0.019	0.919 ± 0.068	0.735 ± 0.083	0.993 ± 0.006	0.766 ± 0.071	0.558 ± 0.107	0.920 ± 0.077
Borderline2-SMOTE	0.709 ± 0.117	0.331 ± 0.047	0.924 ± 0.026	0.921 ± 0.069	0.585 ± 0.089	0.988 ± 0.005	0.810 ± 0.113	0.495 ± 0.114	0.906 ± 0.065
K-means SMOTE	0.588 ± 0.154	0.363 ± 0.134	0.923 ± 0.033	0.897 ± 0.046	0.713 ± 0.065	0.991 ± 0.007	0.784 ± 0.073	0.599 ± 0.076	0.896 ± 0.090
SVMSMOTE	0.636 ± 0.125	0.397 ± 0.124	0.943 ± 0.025	0.908 ± 0.059	0.724 ± 0.079	0.992 ± 0.008	0.766 ± 0.070	0.555 ± 0.085	0.909 ± 0.076
SWIM-RBF	0.498 ± 0.151	0.346 ± 0.182	0.924 ± 0.039	0.932 ± 0.051	0.744 ± 0.097	0.993 ± 0.005	0.800 ± 0.043	0.496 ± 0.052	0.918 ± 0.048
LoRAS	0.259 ± 0.226	0.187 ± 0.179	0.929 ± 0.030	0.898 ± 0.034	0.765 ± 0.107	0.990 ± 0.009	0.731 ± 0.071	0.611 ± 0.082	0.936 ± 0.039
ASN-SMOTE	0.599 ± 0.105	0.399 ± 0.100	0.923 ± 0.048	0.921 ± 0.042	0.740 ± 0.094	0.993 ± 0.005	0.784 ± 0.073	0.569 ± 0.040	0.922 ± 0.072

bold values indicate the best performances

Fig. 6 The comprehensive ranking of oversampling techniques in 24 data sets. **a** Achieved by KNN classifier, **b** achieved by SVM classifier, **c** achieved by RF classifier

Fig. 7 Distributions of minority instances generated by SMOTE, SMOTE/NF and ASN-SMOTE on various two dimensional datasets

differences in treatments across multiple test attempts. The null hypothesis in Friedman test is whether all classifiers are performing similarly in mean rankings without significant difference. From the results of Friedman's test in Table 6, it can be found that there exists enough evidence at $\alpha = 0.05$ to reject the null hypothesis for all classifiers and measures, which demonstrates significant differences between

ASN-SMOTE and other oversampling methods in a statistical sense.

Since the null hypothesis is rejected for all classifiers and performance measures, a post hoc test is applied. Holm's test is used for this study where the proposed method was regarded as the control method. The Holm's test is the non-parametric equivalent of multiple t test that adjusts α to

Table 6 Friedman' test results

<i>F</i> -measure		<i>G</i> -mean		AUC	
Classification method	<i>P</i> value	Classification method	<i>P</i> value	Classification method	<i>P</i> value
KNN	1.63E−09	KNN	1.68E−10	KNN	2.81E−06
SVM	6.91E−08	SVM	1.76E−09	SVM	3.69E−10
RF	2.51E−05	RF	8.70E−05	RF	2.51E−05

Table 7 Results of Holm's test

Classifier methods	KNN		SVM		RF	
	$\alpha = 0.1$	<i>P</i> value	$\alpha = 0.1$	<i>P</i> value	$\alpha = 0.1$	<i>P</i> value
<i>G</i> -mean						
Random	0.0131	2.14E−10	0.0174	3.20E−05	0.0131	1.17E−05
SMOTE	0.0345	4.88E−05	0.0513	5.19E−02	0.0260	3.15E−02
ADASYN	0.0174	1.11E−07	0.0149	5.98E−07	0.0174	3.16E−04
Borderline1-SMOTE	0.0260	3.02E−05	0.0260	3.51E−04	0.0345	4.05E−02
Borderline2-SMOTE	0.0209	1.98E−07	0.0116	4.12E−10	0.0513	8.12E−02
KMeansSMOTE	0.0116	1.54E−10	0.0131	2.66E−07	0.0116	2.35E−06
SVMSMOTE	0.0149	1.52E−09	0.0209	5.05E−05	0.0209	7.94E−04
SWIM-RBF	0.0513	6.62E−03	0.0345	4.56E−02	0.1	9.05E−02
LoRAS	0.1	2.95E−01	0.1	5.04E−01	0.0149	2.15E−04
<i>F</i> -measure						
Random	0.0174	1.97E−07	0.0131	1.62E−07	0.0174	1.09E−02
SMOTE	0.0345	8.71E−05	0.0345	1.51E−03	0.0209	1.45E−02
ADASYN	0.0131	1.77E−11	0.0149	2.75E−07	0.0116	3.23E−06
Borderline1-SMOTE	0.0260	2.16E−05	0.0209	4.26E−05	0.1	1.83E−01
Borderline2-SMOTE	0.0116	6.36E−12	0.0116	5.94E−09	0.0131	4.07E−06
KMeansSMOTE	0.0513	8.71E−05	0.0513	6.85E−03	0.0345	3.64E−02
SVMSMOTE	0.0149	3.52E−08	0.0174	9.89E−07	0.0149	3.78E−03
SWIM-RBF	0.0209	8.11E−06	0.0260	1.86E−04	0.0513	9.17E−02
LoRAS	0.1	5.36E−02	0.1	1.22E−01	0.0260	2.49E−02
AUC						
RandomOver	0.0116	2.17E−07	0.0149	6.69E−07	0.0131	1.17E−05
SMOTE	0.0345	5.31E−02	0.0345	6.38E−03	0.0260	3.15E−02
ADASYN	0.0131	4.53E−06	0.0174	8.72E−07	0.0174	3.16E−04
Borderline1-SMOTE	0.0209	3.65E−04	0.0260	5.70E−05	0.0345	4.05E−02
Borderline2-SMOTE	0.0174	2.45E−04	0.0116	1.56E−09	0.0513	8.12E−02
KMeansSMOTE	0.0260	7.45E−03	0.0131	1.71E−07	0.0116	2.35E−06
SVMSMOTE	0.0149	3.64E−05	0.0209	2.47E−06	0.0209	7.94E−04
SWIM-RBF	0.0513	1.78E−01	0.0513	1.98E−01	0.1	9.05E−02
LoRAS	0.1	2.36E−01	0.1	4.32E−01	0.0149	2.15E−04

compensate for multiple comparisons in a step-down procedure. The largest α is equal to 0.1 in our experiments. The null hypothesis is that ASN-SMOTE does not performs better than the other methods as the control algorithm. The adjusted α and the corresponding p value for each method are shown in Table 7. It can be read from the results that ASN-SMOTE outperforms all other oversampling approach except LoRAS

in terms of all metrics in a statistical sense when using SVM and KNN as the classifier. When RF was used as the classifier, in 26 of the 27 tests, ASN-SMOTE performs better than the counterparts statistically.

Next, we evaluate the runtimes of ASN-SMOTE and all comparative techniques over the 24 benchmark datasets. The average results on ten independent tests are shown in Table 8.

Table 8 Runtimes (in second) of different over-sampling techniques over 24 tested datasets

Datasets	page-blocks0	Pima	Car	Iris	Ecoli	Seeds	Wine	Yeast
Random	0.005	0.0007	0.0016	0.0005	0.001	0.0005	0.0006	0.0012
SMOTE	0.0081	0.0027	0.0029	0.0018	0.0017	0.002	0.002	0.0048
ADASYN	0.0275	0.0051	0.0057	0.0024	0.0028	0.0023	0.0025	0.0118
Borderline1-SMOTE	0.0266	0.0053	0.0059	0.0025	0.0032	0.0029	0.0023	0.0128
Borderline2-SMOTE	0.0247	0.0073	0.0072	0.0032	0.0028	0.0035	0.0028	0.0123
KMeansSMOTE	0.0416	0.0218	0.0315	0.0217	0.0242	0.0207	0.0257	0.0325
SVMSMOTE	0.2653	0.0183	0.0205	0.0054	0.005	0.0052	0.0042	0.0539
SWIM-RBF	928.2842	7.3782	113.1541	0.2437	2.6716	0.4631	0.382	28.5437
LoRAS	2.2849	1.2847	0.4571	0.3874	0.394	0.4881	0.4193	1.6953
ASN-SMOTE	0.4448	0.1744	0.0532	0.0383	0.0339	0.0527	0.0442	0.2276
Datasets	Newthyroid2	Haberman	Glass0	Glass1	Glass4	Glass5	Glass6	Yeast1
Random	0.0006	0.0007	0.0005	0.0006	0.0007	0.0007	0.0007	0.0014
SMOTE	0.0019	0.0018	0.002	0.0025	0.0014	0.0019	0.0021	0.0033
ADASYN	0.0019	0.0035	0.0022	0.0025	0.0025	0.0017	0.0024	0.0109
Borderline1-SMOTE	0.0027	0.0036	0.004	0.0022	0.0022	0.0023	0.0023	0.0083
Borderline2-SMOTE	0.0026	0.0038	0.0028	0.0028	0.0023	0.0022	0.0027	0.0108
KMeansSMOTE	0.0138	0.0245	0.0164	0.0148	0.0136	0.0285	0.0145	0.0331
SVMSMOTE	0.0041	0.0102	0.0052	0.0072	0.0041	0.0033	0.0044	0.0333
SWIM-RBF	1.1159	1.8899	0.5036	0.4286	1.5251	1.6239	1.1726	51.7894
LoRAS	0.3249	0.447	0.4443	0.4627	0.2321	0.2433	0.3189	1.0616
ASN-SMOTE	0.0273	0.0501	0.0449	0.0478	0.0129	0.0066	0.0199	0.1357
Datasets	Dermatology-6	Yeast2	Yeast3	Yeast4	Yeast5	Yeast6	Yeast7	Yeast8
Random	0.0008	0.0011	0.0007	0.0006	0.0006	0.0014	0.0011	0.0011
SMOTE	0.0017	0.0025	0.0017	0.0017	0.002	0.0021	0.0017	0.0018
ADASYN	0.0032	0.0071	0.0029	0.0026	0.0027	0.0048	0.0046	0.0047
Borderline1-SMOTE	0.005	0.0071	0.0031	0.0025	0.0028	0.0049	0.0048	0.0049
Borderline2-SMOTE	0.0037	0.0077	0.0034	0.0033	0.0035	0.0057	0.0056	0.0052
KMeansSMOTE	0.028	0.0281	0.0215	0.0254	0.0231	0.0257	0.0243	0.0234
SVMSMOTE	0.0097	0.0187	0.0061	0.0048	0.0058	0.0146	0.0108	0.0104
SWIM-RBF	4.7086	62.1207	7.57	6.6333	8.0393	75.9582	76.495	78.1148
LoRAS	0.3022	0.7054	0.3852	0.3325	0.2877	0.3806	0.3592	0.3272
ASN-SMOTE	0.0206	0.1137	0.0348	0.0272	0.0124	0.0337	0.0371	0.0271

To make the comparisons more intuitive, the ranking of average running time of these algorithms is provided in Fig. 8.

From the results, it can be found that the random method is the fastest of all oversampling techniques due to its simplicity and SMOTE ranks the second. Compared with the original SMOTE, the variants of SMOTE such as ADASYN, Borderline-smote, K -means SMOTE and SVMSMOTE need more runtimes. It makes sense since they introduce additional mechanisms to better overcome the imbalance which meanwhile make them more complex. Similarly, ASN-SMOTE involves an adaptive sampling strategy, which needs to calculate the k nearest neighbors of each qualified minority instance in the dataset and hence increases the computational

complexity of the proposed algorithm. Nevertheless, as can be read from Table 8, its runtime of much less than one second is perfectly acceptable.

As for LoRAS, it uses the manifold learning-based dimension reduction technology to reduce the dimension of minority instances before selecting the k -nearest neighbors of instances, which makes it spend more time on it and ranks the second to last. In addition, not surprisingly, SWIM-RBF ranks the last and spend much more time since it utilizes the radial basis function (RBF) with Gaussian kernel to estimate the local density of minority samples and majority classes. The computation of the square root and the whitening operation have $O(d^3)$ and $O(bd^2)$ time complexity, where d is

Fig. 8 Ranking of averaged runtimes of all oversampling techniques in 24 tested datasets

the dimension of the data and b is the number of instances in the minority class.

Parameter analysis

Since ASN-SMOTE uses the minority instances in k nearest qualified neighbors to synthesize new minority instances, the essential parameter k significantly affects its performance. To better investigate the effect of the value of k upon the performance of ASN-SMOTE, comparative evaluations were conducted on six benchmark databases randomly selected from the above 24 ones (see Table 2). Considering the least number of parameters required, we chose SVM as the classifier. The three performance measures were evaluated by fivefold cross validation. Each experiment was repeated three times to obtain the average metrics value to avoid contingency.

Figure 9 shows the results where k is selected among values $(1, 2, \dots, 10)$. As can be seen from the results, the performance of ASN-SMOTE varies along with the value of k . Take the Haberman dataset as an example, the performance measures degrade with the increase of k first, and the performance gets the worst when $k = 3$. After that, as k continues to increase, the performance measures begin to improve, until k reaches 7, where the best performance is achieved. This demonstrated that k should be carefully tuned when ASN-SMOTE is used to balance the dataset.

Conclusion

In this paper, we propose an SMOTE based oversampling method, ASN-SMOTE. Compared to existing oversampling methods, the key advantages of ASN-SMOTE is that ASN-SMOTE performs the novel noise filtering and adaptive neighbor selection mechanisms before synthesizing minority instances. The former filters any minority instance whose nearest neighbor is a majority instance, and the latter only selects minority neighbors that are closer than nearest majority instance for synthetic oversampling. By employing these two mechanisms, ASN-SMOTE can effectively avoid to generate synthetic minority instances in majority class region which results in a more general decision boundary.

ASN-SMOTE is evaluated against nine state-of-the-art oversampling methods on 24 public datasets whose imbalance ratios range from 1.82 to 41.4, with three different well-known classifiers, namely KNN, SVM and RF. The results, based on the G -mean, F -measure and AUC metrics, show that ASN-SMOTE ranks the first and is more effective than all other comparative oversampling techniques. Although the runtime of ASN-SMOTE is not competitive, however, its runtime of much less than one second is perfectly acceptable.

In the future work, we intend to extend ASN-SMOTE to deal with multi-classification imbalanced datasets since the problem of imbalanced datasets not only appears in binary classifications, but also in multi-classifications in real-world applications.

Fig. 9 Performances and rankings of ASN-SMOTE under various k values. **a** G -means, **b** F -measures, **c** AUC

Acknowledgements This work was supported in part by the Zhejiang Provincial Natural Science Foundation of China under Grants LZ20F010008 and in part by the National Undergraduate Innovation and Entrepreneurship Training Program under Grants 202110351060.

Author Contributions ZT, XY and YX contributed to the conception of the study; XY and YX performed the experiment; XY, YX, and QH contributed significantly to data analysis and manuscript preparation; XY, SK, YX, QH and WL performed the data analyses and wrote the manuscript; ZT helped perform the analysis with constructive discussions.

Availability of data and material The datasets extracted and analyzed during the current study are available in the KEEL repository and the UCI Machine Learning Repository.

Declarations

Conflict of interest The authors declare that there are no conflicts of interest regarding the publication of this paper.

Code availability The ASN-SMOTE implementation is available at <https://github.com/yixinkai123/ASNSMOTE/>.

Ethics approval The authors declare that there is no ethics involved in this experiment.

Consent to participate Written informed consent for publication was obtained from all participants.

Consent for publication The study received approval from all participants.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copy-

right holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Alcalá-Fdez, Fernández J, Luengo A, Derrac J, García J, Sánchez S, Herrera L F (2011) Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *J Mult Valued Logic Soft Comput* 17: 255–287
- Bach M, Werner A, żywiec J, Pluskiewicz W (2017) The study of under- and over-sampling methods' utility in analysis of highly imbalanced data on osteoporosis. *Inf Sci* 384:174–190. <https://doi.org/10.1016/j.ins.2016.09.038>
- Barua S, Islam MM, Yao X, Murase K (2014) MWMOTE-majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Trans Knowl Data Eng* 26(2):405–425. <https://doi.org/10.1109/TKDE.2012.232>
- Batista GEAPA, Prati RC, Monard MC (2004) A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor Newsl* 6(1):20–29. <https://doi.org/10.1145/1007730.1007735>
- Bedi P, Gupta N, Jindal V (2020) I-SiamIDS: an improved Siam-IDS for handling class imbalance in network-based intrusion detection systems. *Appl Intell* 51:1133–1151 (2021). <https://doi.org/10.1007/s10489-020-01886-y>
- Bej S, Davtyan N, Wolfien M, Nassar M, Wolkenhauer O (2021) Loras: an oversampling approach for imbalanced datasets. *Mach Learn* 110(2):279–301
- Bellinger C, Sharma S, Japkowicz N, Zaiane OR (2020) Framework for extreme imbalance classification: swim-sampling with the majority class. *Knowl Inf Syst* 62(3):841–866
- Breiman L (2001) Random forests. *Mach Learn* 45(1):5–32
- Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C (2009) Safe-Level-SMOTE: safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In: *PAKDD*
- Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C (2011) DBSMOTE: density-based synthetic minority over-sampling technique. *Appl Intell* 36:664–684
- Castro CL, Braga AP (2013) Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data. *IEEE Trans Neural Netw Learn Syst* 24(6):888–899. <https://doi.org/10.1109/TNNLS.2013.2246188>
- Chambolle A, De Vore R, Lee NY, Lucier B (1998) Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Trans Image Process* 7(3):319–335. <https://doi.org/10.1109/83.661182>
- Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. *J Artif Intell Res* 16:321–357
- Chawla NV, Japkowicz N, Kotcz A (2004) Editorial: special issue on learning from imbalanced data sets. *SIGKDD Explor Newsl* 6(1):1–6. <https://doi.org/10.1145/1007730.1007733>
- Chen XS, Kang Q, Zhou MC, Wei Z (2016) A novel under-sampling algorithm based on iterative-partitioning filters for imbalanced classification. In: *IEEE international conference on automation science and engineering*
- Cieslak D, Chawla N (2006) Combating imbalance in network intrusion datasets. In: *2006 IEEE international conference on granular computing*, pp 732–737. <https://doi.org/10.1109/GRC.2006.1635905>
- Cortes C, Vapnik V (1995) Support-vector networks. *Mach Learn* 20(3):273–297
- Cover T, Hart P (1967) Nearest neighbor pattern classification. *IEEE Trans Inf Theory* 13(1):21–27. <https://doi.org/10.1109/TIT.1967.1053964>
- Devi D, Biswas SK, Purkayastha B (2019) Learning in presence of class imbalance and class overlapping by using one-class SVM and undersampling technique. *Connect Sci* 31:105–142
- Douzas G, Bacao F (2019) Geometric SMOTE a geometrically enhanced drop-in replacement for SMOTE. *Inf Sci* 501:118–135. <https://doi.org/10.1016/j.ins.2019.06.007>
- Douzas G, Bacao F, Last F (2018) Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Inf Sci* 465:1–20. <https://doi.org/10.1016/j.ins.2018.06.056>
- Dua D, Graff C (2019) UCI machine learning repository. University of California, School of Information and Computer Science, Irvine. <http://archive.ics.uci.edu/ml>
- Fernández A, López V, Galar M, del Jesus MJ, Herrera F (2013) Analysing the classification of imbalanced data-sets with multiple classes: binarization techniques and ad-hoc approaches. *Knowl Based Syst* 42:97–110. <https://doi.org/10.1016/j.knosys.2013.01.018>
- Galar M, Fernandez A, Barrenechea E, Bustince H, Herrera F (2012) A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Trans Syst Man Cybern Part C (Appl Rev)* 42(4):463–484. <https://doi.org/10.1109/TSMCC.2011.2161285>
- Gao M, Hong X, Chen S, Harris CJ (2011) A combined SMOTE and PSO based RBF classifier for two-class imbalanced problems. *Neurocomputing* 74(17):3456–3466. <https://doi.org/10.1016/j.neucom.2011.06.010>
- García S, Herrera F (2009) Evolutionary undersampling for classification with imbalanced datasets: proposals and taxonomy. *Evol Comput* 17(3):275–306. <https://doi.org/10.1162/evco.2009.17.3.275>
- Ghazikhani A, Monsefi R, Yazdi H (2014) Online neural network model for non-stationary and imbalanced data stream classification. *Int J Mach Learn Cybern* 5:51–62
- Han H, Wang WY, Mao BH (2005) Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: *Proceedings of the 2005 international conference on advances in intelligent computing—volume part I, ICIC'05*. Springer, Berlin, Heidelberg, pp 878–887. https://doi.org/10.1007/11538059_91
- Hart P (1968) The condensed nearest neighbor rule (corresp.). *IEEE Trans Inf Theory* 14(3):515–516. <https://doi.org/10.1109/TIT.1968.1054155>
- He H, Bai Y, Garcia EA, Li S (2008) ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: *2008 IEEE international joint conference on neural networks (IEEE World congress on computational intelligence)*, pp 1322–1328. <https://doi.org/10.1109/IJCNN.2008.4633969>
- He H, Garcia EA (2009) Learning from imbalanced data. *IEEE Trans Knowl Data Eng* 21(9):1263–1284. <https://doi.org/10.1109/TKDE.2008.239>
- Hollander M, Wolfe DA, Chicken E (2013) *Nonparametric statistical methods*, vol 751. Wiley, New York
- Holm S (1979) A simple sequentially rejective multiple test procedure. *Scand J Stat* 65–70
- hui Hou W, kang Wang X, yu Zhang H, qiang Wang J, Li L (2020) A novel dynamic ensemble selection classifier for an imbalanced data set: an application for credit risk assessment. *Knowl Based Syst* 208:106462. <https://doi.org/10.1016/j.knosys.2020.106462>
- Hu S, Liang Y, Ma L, He Y (2009) MSMOTE: improving classification performance when training data is imbalanced. In: *2009 second international workshop on computer science and engineering*, vol 2, pp 13–17. <https://doi.org/10.1109/WCSE.2009.756>

36. Jensen D (1997) Prospective assessment of AI technologies for fraud detection: a case study. In: AAAI workshop on AI approaches to fraud detection and risk management. Citeseer, pp 34–38
37. Kotsiantis S, Kanellopoulos D, Pintelas P et al (2006) Handling imbalanced datasets: a review. *GESTS Int Trans Comput Sci Eng* 30(1):25–36
38. Kubát M, Matwin S (1997) Addressing the curse of imbalanced training sets: one-sided selection. In: *ICML*
39. Laurikkala J (2001) Improving identification of difficult small classes by balancing class distribution. In: *Proceedings of the 8th conference on AI in medicine in Europe: artificial intelligence medicine, AIME '01*. Springer, Berlin, Heidelberg, pp 63–66
40. Lee H, Kim J, Kim S (2017) Gaussian-based SMOTE algorithm for solving skewed class distributions. *Int J Fuzzy Log Intell Syst* 17:229–234
41. Li Y, Guo H, Zhang Q, Gu M, Yang J (2018) Imbalanced text sentiment classification using universal and domain-specific knowledge. *Knowl Based Syst* 160:1–15. <https://doi.org/10.1016/j.knosys.2018.06.019>
42. Lin WC, Tsai CF, Hu YH, Jhang JS (2017) Clustering-based under-sampling in class-imbalanced data. *Inf Sci* 409–410:17–26. <https://doi.org/10.1016/j.ins.2017.05.008>
43. Nekooimehr I, Lai-Yuen SK (2016) Adaptive semi-supervised weighted oversampling (A-SUWO) for imbalanced datasets. *Expert Syst Appl* 46:405–416. <https://doi.org/10.1016/j.eswa.2015.10.031>
44. Nguyen H.M, Cooper E, Kamei K (2011) Borderline over-sampling for imbalanced data classification. *Int. J. Knowl. Eng. Soft Data Paradigms* 3:4–21
45. Orriols-Puig A, Bernado-Mansilla E, Goldberg DE, Sastry K, Lanzi PL (2009) Facetwise analysis of xcs for problems with class imbalances. *IEEE Trans Evol Comput* 13(5):1093–1119
46. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in python. *J Mach Learn Res* 12(null):2825–2830
47. Popel MH, Hasib KM, Ahsan Habib S, Muhammad Shah F (2018) A hybrid under-sampling method (HUSBoost) to classify imbalanced data. In: *2018 21st international conference of computer and information technology (ICCIT)*, pp 1–7. [10.1109/ICCITECHN.2018.8631915](https://doi.org/10.1109/ICCITECHN.2018.8631915)
48. Ramentol E, Caballero Y, Bello R, Herrera F (2012) SMOTE-RSB*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory. *Knowl Inf Syst* 33(2):245–265. <https://doi.org/10.1007/s10115-011-0465-6>
49. Smith MR, Martinez T, Giraud-Carrier C (2014) An instance level analysis of data complexity. *Mach Learn* 95(2):225–256. <https://doi.org/10.1007/s10994-013-5422-z>
50. Sáez JA, Luengo J, Stefanowski J, Herrera F (2015) SMOTE-IPF: addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Inf Sci* 291:184–203. <https://doi.org/10.1016/j.ins.2014.08.051>
51. Tahir MA, Kittler J, Yan F (2012) Inverse random under sampling for class imbalance problem and its application to multi-label classification. *Pattern Recognit* 45(10):3738–3750. <https://doi.org/10.1016/j.patcog.2012.03.014>
52. Tang B, He H (2015) KernelADASYN: kernel based adaptive synthetic data generation for imbalanced learning. In: *2015 IEEE congress on evolutionary computation (CEC)*, pp. 664–671. <https://doi.org/10.1109/CEC.2015.7256954>
53. Tao X, Li Q, Guo W, Ren C, He Q, Liu R, Zou J (2020) Adaptive weighted over-sampling for imbalanced datasets based on density peaks clustering with heuristic filtering. *Inf Sci* 519:43–73. <https://doi.org/10.1016/j.ins.2020.01.032>
54. Tomek I (1976) Two modifications of CNN. *IEEE Trans Syst Man Cybern SMC*–6(11):769–772. <https://doi.org/10.1109/TSMC.1976.4309452>
55. Vo MT, Nguyen T, Vo HA, Le T (2021) Noise-adaptive synthetic oversampling technique. *Appl Intell* 51:7827–7836 (2021). <https://doi.org/10.1007/s10489-021-02341-2>
56. Weiss GM (1995) Learning with rare cases and small disjuncts. In: *Prieditis A, Russell S (eds) Machine learning Proceedings 1995*. Morgan Kaufmann, San Francisco (CA), pp 558–565. <https://doi.org/10.1016/B978-1-55860-377-6.50075-X>
57. Wilson DL (1972) Asymptotic properties of nearest neighbor rules using edited data. *IEEE Trans Syst Man Cybern SMC*–2(3):408–421. <https://doi.org/10.1109/TSMC.1972.4309137>
58. Yu H, Ni J, Zhao J (2013) ACOSampling: an ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data. *Neurocomputing* 101:309–318. <https://doi.org/10.1016/j.neucom.2012.08.018>
59. Zhou M, Lin F, Hu Q, Tang Z, Jin C (2020) AI-enabled diagnosis of spontaneous rupture of ovarian endometriomas: a PSO enhanced random forest approach. *IEEE Access* 8:132253–132264. <https://doi.org/10.1109/ACCESS.2020.3008473>
60. Zhou ZH, Liu XY (2006) Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Trans Knowl Data Eng* 18(1):63–77. <https://doi.org/10.1109/TKDE.2006.17>

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