Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A novel overlapping minimization SMOTE algorithm for imbalanced classification*#

Yulin HE‡[1,2], Xuan LU[2], Philippe FOURNIER-VIGER[2], Joshua Zhexue HUANG[1,2]

[1]*Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen 518107, China*

[2]*College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China*

E-mail: yulinhe@gml.ac.cn; 2110276215@email.szu.edu.cn; philfv@szu.edu.cn; zx.huang@szu.edu.cn

Received Apr. 21, 2023; Revision accepted Sept. 25, 2023; Crosschecked July 3, 2024; Published online Sept. 5, 2024

Abstract: The synthetic minority oversampling technique (SMOTE) is a popular algorithm to reduce the impact of class imbalance in building classifiers, and has received several enhancements over the past 20 years. SMOTE and its variants synthesize a number of minority-class sample points in the original sample space to alleviate the adverse effects of class imbalance. This approach works well in many cases, but problems arise when synthetic sample points are generated in overlapping areas between different classes, which further complicates classifier training. To address this issue, this paper proposes a novel generalization-oriented rather than imputation-oriented minority-class sample point generation algorithm, named overlapping minimization SMOTE (OM-SMOTE). This algorithm is designed specifically for binary imbalanced classification problems. OM-SMOTE first maps the original sample points into a new sample space by balancing sample encoding and classifier generalization. Then, OM-SMOTE employs a set of sophisticated minority-class sample point imputation rules to generate synthetic sample points that are as far as possible from overlapping areas between classes. Extensive experiments have been conducted on 32 imbalanced datasets to validate the effectiveness of OM-SMOTE. Results show that using OM-SMOTE to generate synthetic minority-class sample points leads to better classifier training performances for the naive Bayes, support vector machine, decision tree, and logistic regression classifiers than the 11 state-of-the-art SMOTE-based imputation algorithms. This demonstrates that OM-SMOTE is a viable approach for supporting the training of high-quality classifiers for imbalanced classification. The implementation of OM-SMOTE is shared publicly on the GitHub platform at https://github.com/luxuan123123/OM-SMOTE/.

Key words: Imbalanced classification; Synthetic minority oversampling technique (SMOTE); Majority-class sample point; Minority-class sample point; Generalization capability; Overlapping minimization

https://doi.org/10.1631/FITEE.2300278 **CLC number:** TP301

1 Introduction

Classification is an important supervised machine learning task, where the goal is to assign a class label to each unlabeled sample point. Imbalanced classification (He HB and Garcia, 2009) is the task of creating a classifier to perform classification when the distribution of classes in the training data is imbalanced. In recent years, imbalanced classification has attracted the attention of numerous researchers because imbalanced datasets are common

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 61972261), the Natural Science Foundation of Guangdong Province, China (No. 2023A1515011667), the Key Basic Research Foundation of Shenzhen, China (No. JCYJ20220818100205012), and the Basic Research Foundation of Shenzhen, China (No. JCYJ20210324093609026)

Electronic supplementary materials: The online version of this article (https://doi.org/10.1631/FITEE.2300278) contains supplementary materials, which are available to authorized users

ORCID: Yulin HE, https://orcid.org/0000-0002-3415-0686

in many practical applications such as identifying positive COVID-19 cases (Moulaei et al., 2022) and predicting typhoon landing locations from annual climate data (Li W et al., 2022). In such applications, incorrectly classifying minority-class sample points (MinCSPs) may cause more serious consequences than the misclassification of majority-class sample points (MajCSPs). For instance, if a person is infected with COVID-19 but incorrectly diagnosed, it may cause further spread of the virus in the population. Similarly, failing to predict a typhoon sufficiently in advance may delay the necessary prevention steps and thus cause loss of life and property. Thus, for imbalanced classification tasks, it is generally crucial to construct classifiers that can predict the MinCSPs as correctly as possible.

A good classifier for imbalanced data should thus favor the MinCSPs over those from the majority class. This can be achieved by two main strategies. The first, called the data-oriented strategy, is to generate synthetic MinCSPs to increase their numbers and thus construct classifiers that are more accurate at identifying the minority class. The second strategy, namely the classifier-oriented strategy, is to force a classifier to learn more from the MinCSPs (He YL et al., 2022), thus moving the classification boundaries towards MinCSPs. In comparison to the first strategy, the second strategy has the disadvantages of frequently causing overfitting and being highly complex, particularly for relatively large imbalance ratios (IRs). Hence, over the last decade, data-oriented imbalanced classification techniques have drawn increasing attention from academia and industry (Guo et al., 2017). A representative algorithm for the generation of synthetic MinCSPs is the synthetic minority oversampling technique (SMOTE) (Chawla et al., 2002). It can considerably improve imbalanced classification performance by training classifiers on relatively balanced datasets composed of the original sample points (MajCSPs and MinCSPs) and synthetic MinCSPs. SMOTE first determines the k nearest neighbors of each original MinCSP and then repeatedly imputes synthetic MinCSPs between original MinCSPs and their nearest neighbors in a random way.

Numerous SMOTE enhancements have been proposed (Fernández et al., 2018; Kovács, 2019) to address various limitations of the SMOTE algorithm. These improvements focus on several aspects such as (1) the selection of seed MinCSPs and auxiliary MinCSPs, (2) the generation of synthetic MinCSPs, and (3) the application of SMOTE in other sample spaces. Experimental results have shown that these extended algorithms perform reasonably well for imbalanced classification tasks. However, none of them offers a solution to the problem that synthetic MinCSPs and original MajCSPs may be located in overlapping areas. This is because synthetic MinCSPs are generated by considering only the original MinCSPs rather than both the original MinCSPs and MajCSPs. This has three negative effects on the construction of subsequent classifiers:

1. Imputing inconsistent synthetic MinCSPs. The generation of an inconsistent probability distribution between synthetic MinCSPs and the original MinCSPs is unavoidable due to the linear imputation process of SMOTE and its variants. SMOTE and its extensions are based on the assumption that the linear or geometric space between two MinCSPs belongs to the minority class (Douzas et al., 2021). However, this assumption is violated when classes in the original data overlap heavily or are linearly indistinguishable.

2. Aggravating intra-class imbalance. Most current methods use the k-nearest neighbor (KNN) method to select auxiliary samples, which takes into account only the local information. The consequence, as depicted in Fig. 1a, is that new samples tend to be generated in areas where the MinCSPs are dense, which aggravates the intra-class imbalance.

3. Neglecting the information contained in the original MajCSPs. Synthetic MinCSPs are generated without taking into account the information of original MajCSPs, i.e., the adaptability of MajCSPs to synthetic MinCSPs. That is to say, synthetic MinCSPs may be located in an area containing original MajCSPs, which may result in sample pollution for the original MajCSPs, as shown in Fig. 1b. This overlap of different classes increases the classification uncertainty (Tang et al., 2010; Sáez et al., 2019).

To address the aforementioned issues of SMOTE-based algorithms for imbalanced learning, a novel overlapping minimization SMOTE (OM-SMOTE) algorithm is introduced. OM-SMOTE is a generation-oriented oversampling algorithm for binary class imbalance problems. Unlike most current oversampling methods that focus solely on data imputation, the OM-SMOTE algorithm prioritizes

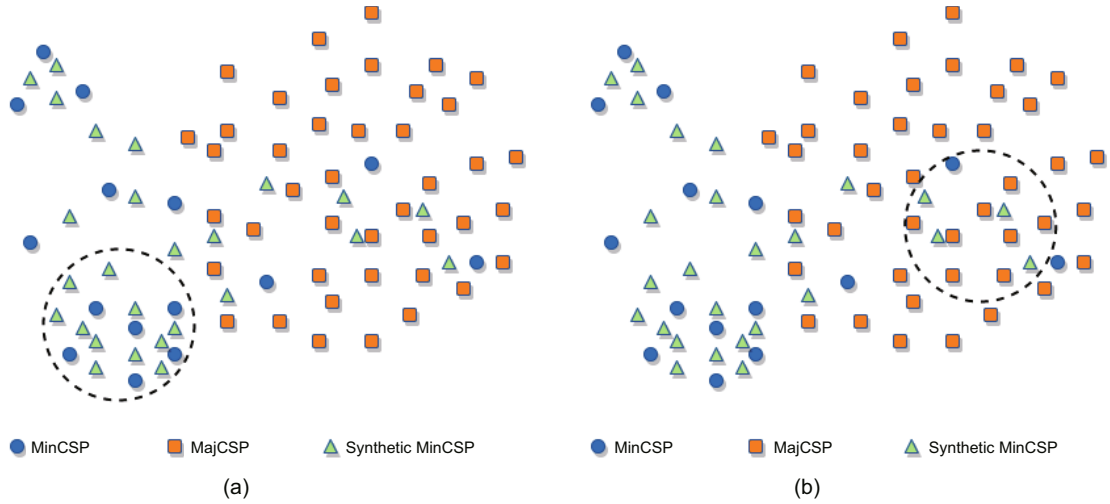

Fig. 1 Negative effects of existing synthetic minority oversampling techniques: (a) aggravating intra-class imbalance; (b) generating samples in overlapping regions

the impact of oversampling results on classifier performance. It avoids generating new samples in the overlapping regions, helping classifiers learn decision boundaries more easily and improving the overall generalization performance.

The proposed OM-SMOTE algorithm has two main components: the overlapping alleviation transformation (OAT) and retreating interpolation (RI) to impute synthetic MinCSPs. OM-SMOTE first maps the original sample points, including MinCSPs and MajCSPs, into a new sample space by balancing sample transformation and classifier generalization. Then, the novel MinCSP imputation rules are applied to generate synthetic sample points that are as far as possible from overlapping areas between the classes. Extensive experiments are conducted to validate the effectiveness of the OM-SMOTE algorithm, including a comparison with 11 state-of-the-art SMOTE-based imputation algorithms on 32 imbalanced datasets. Results show that the new OM-SMOTE algorithm can help the naive Bayes, support vector machine (SVM), decision tree (DT), and logistic regression (LR) classifiers obtain better imbalanced classification performances in terms of testing accuracy, area under the receiver operating characteristic (ROC) curve (AUC) (Hand and Till, 2001), geometric means of classification accuracy of every class (G-mean) (Sun et al., 2006), and F1-score (F1) (Lipton et al., 2014). The main contributions of this paper are threefold:

1. We propose a sophisticated nonlinear data transformation method. By using the structure of an auto-encoder (AE) as the starting point, we design an OAT method which provides bidirectional transformation between the original space and the latent space to ensure the quality of data reconstruction while transforming data into a more separable space.

2. We construct an effective synthetic MinCSP generation mechanism. The furthest neighbors of seed samples are selected as auxiliary samples to mitigate the negative effect of sample selection on synthetic sample generation, and a set of sophisticated imputation rules are developed to avoid generating samples in overlapping regions.

3. We design exhaustive experiments to systematically evaluate the performance of OM-SMOTE, including validating its convergence, verifying its imputation distribution consistency, and comparing its performance with those of SMOTE-based sample generation algorithms.

2 Related works

SMOTE (Chawla et al., 2002) was proposed to improve the performance of classifiers in identifying minority-class samples. SMOTE is an oversampling algorithm to rebalance an original training dataset. SMOTE is applied in two phases: data selection and data generation.

1. In the data selection phase, SMOTE randomly selects a MinCSP x_i as the seed sample and

then randomly selects an auxiliary sample x_j from its k nearest minority-class neighbors.

2. In the data generation phase, a synthetic sample $\hat{x}$ is generated along the line between the seed sample and the auxiliary sample. The generation mechanism is depicted in Eq. (1):

$$\hat{x} = x_i + \lambda(x_j - x_i), \tag{1}$$

where λ is a random number in $[0, 1]$.

SMOTE is the first algorithm for balancing datasets by generating synthetic minority samples and has become one of the most popular oversampling techniques for handling imbalanced classification problems. Up until now, more than 100 extensions of SMOTE have been proposed. These extensions improve SMOTE in several aspects:

1. Modify the selection rules for seed samples and auxiliary samples. This approach assumes that selecting better candidate samples to be oversampled can reduce overlaps and noise in the final dataset (Fernández et al., 2018). A representative algorithm of this type is Borderline-SMOTE (Han et al., 2005), which focuses on samples in the borderline region which are more prone to be misclassified. Borderline-SMOTE selects MinCSPs close to the border as seed samples. Another algorithm called the adaptive synthetic (ADASYN) sampling approach (He HB et al., 2008) adaptively adjusts the weights of different MinCSPs according to their distribution. Then, samples with higher weights are more likely to be oversampled. The k-means SMOTE (Douzas et al., 2018) algorithm selects seed samples from clusters with a higher representation of MinCSPs to avoid noise generation. In contrast with most oversampling methods based on the KNN algorithm, the farthest SMOTE (FSMOTE) algorithm (Gosain and Sardana, 2019) selects an auxiliary sample from the k farthest neighbors of a seed sample based on the assumption that this data selection strategy can widen the decision area corresponding to the minority class. The synthetic minority based on probabilistic distribution (SyMProD) approach (Kunakorntum et al., 2020) assigns a probability to each MinCSP and selects seed samples based on the probability distribution rather than the sample weight.

2. Modify the synthetic MinCSP generation mechanisms. Many studies proposed new data generation rules to replace the random linear interpolation of SMOTE. In Safe-Level-SMOTE (Bunkhumpornpat et al., 2009), the safe level ratio is calculated to restrict the range of λ, so that the newly synthesized samples are closer to high-safe-level samples. In Random-SMOTE (Dong and Wang, 2011), a triangle is formed by the seed sample and two auxiliary samples, and new minority examples are generated randomly within the triangle area. The cluster-based synthetic oversampling (CBSO) algorithm (Barua et al., 2011) incorporates unsupervised clustering in the data generation phase to ensure that the synthesized samples always lie within MinCSP clusters. The geometric SMOTE (G-SMOTE) (Douzas and Bacao, 2019) algorithm generates synthetic samples within a flexible geometric region around each selected seed MinCSP. The localized random affine shadow sampling (LoRAS) algorithm (Bej et al., 2021) adds Gaussian noise to the local distribution of each minority feature to form shadow samples and uses the multiple shadow samples for data generation to better account for local information.

3. Transform samples into another space. Data transformation methods have been proposed to transform data into a new space that provides desirable data properties. The isometric feature mapping (Isomap) algorithm maps data into a low-dimensional space (Gu et al., 2009), where the data are more separable. Some studies have used kernel functions to transform input data into a kernel space (Mathew et al., 2015; Pérez-Ortiz et al., 2016) and then generate synthetic MinCSPs in a new space. AEs are used to transform MinCSPs from the original space into a latent space, and then new samples are generated by adding Gaussian noise (Bellinger et al., 2015) or applying SMOTE (Bellinger et al., 2016) in the latent space. The doping with infrequent normal generator (DOPING) (Lim et al., 2018) algorithm uses adversarial AEs to encode MinCSPs into a Gaussian mixture latent space and then interpolates samples near the distribution boundaries. The auto-encoder extreme learning machine (AE-ELM) (He YL et al., 2022) algorithm encodes the original MinCSPs in a latent space, and then applies crossover, mutation, and filtration operations in the latent space to generate new samples.

Additionally, there are hybrid-sampling methods that combine SMOTE with undersampling algorithms. For example, SMOTE combined with iterative-partitioning filter (SMOTE-IPF) (Sáez

et al., 2015) applies noise filtering after generating new samples, whereas the adaptive multi-objective swarm crossover optimization (AMSCO) (Li JY et al., 2018) algorithm undersamples the MajCSPs while oversampling the MinCSPs, and particle swarm optimization (PSO) is used to find the optimal ratio of the two classes.

3 The proposed OM-SMOTE

This section introduces the proposed OM-SMOTE algorithm, which is applied in two phases. First, OM-SMOTE uses an OAT method to map data into a more separable space, and then an RI mechanism is used to generate only new instances in a safe region. These two components of OM-SMOTE are described next.

3.1 Overlapping alleviation transformation

An AE is a type of neural network that uses the back-propagation algorithm to reconstruct its inputs. The goal of an AE is to minimize the reconstruction error between the input and output while learning an "informative" data representation in the latent space (Bank et al., 2020). Fig. 2a illustrates the structure of a general AE. Based on the AE concept, we propose the OAT method which provides a nonlinear bidirectional transformation between the original space and latent space to alleviate the data overlap problem.

According to the Cover theorem (Cover, 1965), it is possible to transform a dataset that is not linearly separable into a linearly separable dataset with a high probability by projecting it into a higher-dimensional space via some nonlinear transformation. In this work, to ensure that the data are more separable in the new space, we transform the data to a higher-dimensional space, i.e., $\mathcal{L} > \mathcal{D}$, where $\mathcal{D}$ is the dimension of the original dataset and $\mathcal{L}$ is the dimension of the hidden space in the AE network.

Furthermore, we use label information to guide the data transformation process. Fig. 2b illustrates the AE network structure of the proposed OAT method. By comparing Fig. 2b with Fig. 2a, it can be seen that the general AE compresses high-dimensional data into a lower-dimensional hidden space, and hence the dimension of the hidden layer is smaller than that of the input layer, i.e., $\mathcal{L} < \mathcal{D}$. In OAT, two networks are connected to the same hidden layer, i.e., the reconstruction network and the classification network. The reconstruction network's goal is to produce an output $\overline{\boldsymbol{X}}$ that is as similar as possible to the input $\boldsymbol{X}$. The classification network is learning in a supervised manner to predict the targets. We use class labels to guide the data transformation process so that the inter-class distance between the MinCSPs and MajCSPs in the latent space can be maximized.

The learning process of OAT is given by Eq. (2) as

$$\begin{cases} \boldsymbol{H}_{\mathcal{N}\times\mathcal{L}} = s(\boldsymbol{X}_{\mathcal{N}\times\mathcal{D}}\boldsymbol{V}_{\mathcal{D}\times\mathcal{L}}), \\ \overline{\boldsymbol{X}}_{\mathcal{N}\times\mathcal{D}} = s(\boldsymbol{H}_{\mathcal{N}\times\mathcal{L}}\boldsymbol{W}_{\mathcal{L}\times\mathcal{D}}), \\ \overline{\boldsymbol{Y}}_{\mathcal{N}\times 2} = s(\boldsymbol{H}_{\mathcal{N}\times\mathcal{L}}\boldsymbol{U}_{\mathcal{L}\times 2}), \end{cases} \tag{2}$$

where $s(\cdot)$ is the sigmoid function, $\boldsymbol{V}_{\mathcal{D}\times\mathcal{L}}$ is the input-layer weight matrix, and $\boldsymbol{W}_{\mathcal{L}\times\mathcal{D}}$ and $\boldsymbol{U}_{\mathcal{L}\times 2}$ are the output-layer weight matrices. The input $\boldsymbol{X}$ is first transformed to a latent space representation $\boldsymbol{H}$. Then, the reconstruction network uses $\boldsymbol{H}$ to reconstruct the output $\overline{\boldsymbol{X}}$, while the classification network learns to predict the targets of $\boldsymbol{H}$.

In the reconstruction network, the mean squared error (MSE) loss is used as the objective function to minimize the reconstruction loss of inputs, which is defined as

$$L_{\mathrm{MSE}}(\boldsymbol{X}, \overline{\boldsymbol{X}}) = \frac{1}{\mathcal{N}}||\boldsymbol{X} - \overline{\boldsymbol{X}}||_2^2, \tag{3}$$

where $\mathcal{N}$ is the number of original sample points including MajCSPs and MinCSPs.

The objective function of the classification network is a measure of the classification errors by minimizing the cross-entropy loss and is defined as

$$L_{\mathrm{CE}}(\boldsymbol{Y}, \overline{\boldsymbol{Y}}) = -\frac{1}{\mathcal{N}}\sum_{i=1}^{\mathcal{N}}(y_i \ln \overline{y}_i + (1-y_i)\ln(1-\overline{y}_i)), \tag{4}$$

where y_i and $\overline{y}_i$ are the true and predicted labels of the i^{th} sample point, respectively.

L_{OAT} is the combination of the reconstruction loss L_{MSE} and classification loss L_{CE}, defined as

$$L_{\mathrm{OAT}} = \alpha L_{\mathrm{MSE}} + (1-\alpha)L_{\mathrm{CE}}, \tag{5}$$

where $\alpha \in [0, 1]$ is a weighting factor to balance sample transformation and classifier generalization.

3.2 Retreating interpolation

By transforming samples into the latent space, a latent space representation of the input is obtained,

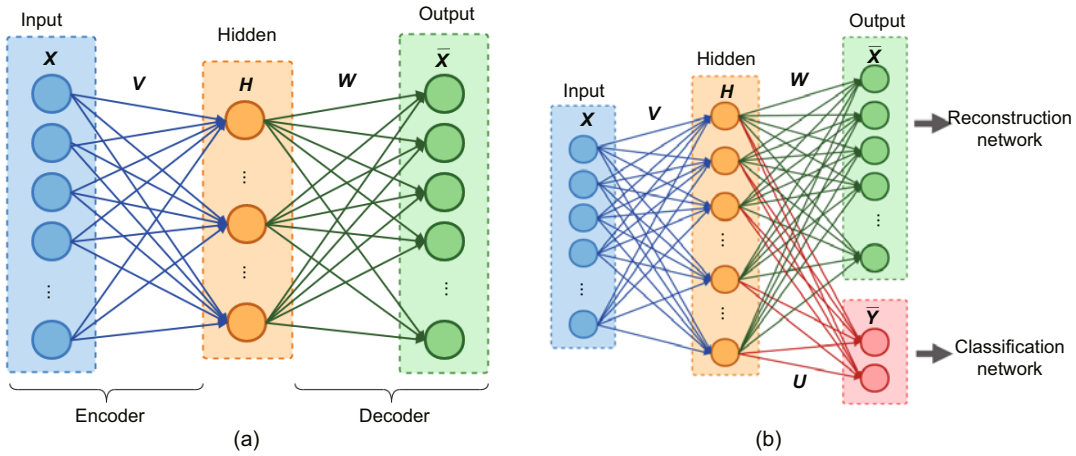

Fig. 2 Network structures corresponding to general auto-encoder and overlapping alleviation transformation methods: (a) general auto-encoder structure; (b) auto-encoder structure in an overlapping alleviation transformation method

i.e., $\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_{\min} \\ \boldsymbol{H}_{\mathrm{maj}} \end{bmatrix}$, where $\boldsymbol{H}_{\min}$ and $\boldsymbol{H}_{\mathrm{maj}}$ are the latent space representations of MinCSPs and MajCSPs, respectively. In an ideal situation, OAT is able to map data to a fully linearly separable space. However, for some complex datasets, there is still a partial overlap between $\boldsymbol{H}_{\min}$ and $\boldsymbol{H}_{\mathrm{maj}}$. Hence, we propose a mechanism called RI to avoid generating samples in the overlapping regions.

In the latent space, we define two hyperspheres $S_{\min}$ and S_{maj} to represent the distribution space of $\boldsymbol{H}_{\min}$ and $\boldsymbol{H}_{\mathrm{maj}}$, respectively. The locations of spheric centers and radii of $S_{\min}$ and S_{maj} are defined as

$$\begin{cases} \boldsymbol{o}_{\min} = \dfrac{1}{n_1}\displaystyle\sum_{k=1}^{n_1} \boldsymbol{h}_k, \\ \boldsymbol{o}_{\mathrm{maj}} = \dfrac{1}{n_2}\displaystyle\sum_{l=1}^{n_2} \boldsymbol{h}_l, \end{cases} \tag{6}$$

$$\begin{cases} r_{\min} = \displaystyle\max_{k=1,2,\cdots,n_1} ||\boldsymbol{h}_k - \boldsymbol{o}_{\min}||, \\ r_{\mathrm{maj}} = \displaystyle\max_{l=1,2,\cdots,n_2} ||\boldsymbol{h}_l - \boldsymbol{o}_{\mathrm{maj}}||, \end{cases} \tag{7}$$

where $\boldsymbol{h}_k \in \boldsymbol{H}_{\min}$, $\boldsymbol{h}_l \in \boldsymbol{H}_{\mathrm{maj}}$, and n_1 and n_2 represent the numbers of MinCSPs and MajCSPs, respectively.

The sample space is divided into three regions, i.e., the safe region, overlapping region, and danger region. As shown in Fig. 3, the safe region is defined as the part of $S_{\min}$ that does not intersect with S_{maj}, the overlapping region is the region where two hyperspheres overlap, and the danger region is the part of

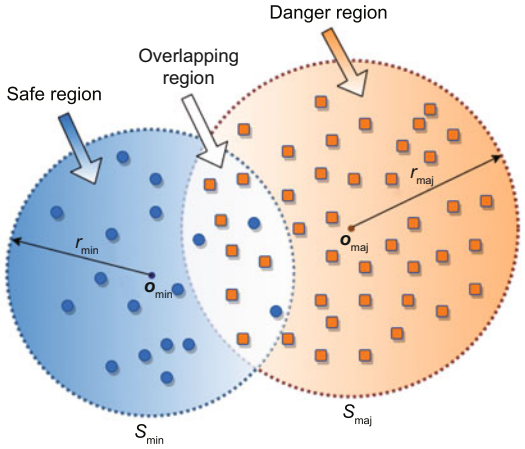

Fig. 3 Hyperspheres corresponding to MajCSPs and MinCSPs

S_{maj} that does not intersect with $S_{\min}$. These three regions can thus be formally described as

$$\begin{cases} \text{safe region} : S_{\min} - S_{\mathrm{maj}}, \\ \text{overlapping region} : S_{\min} \cap S_{\mathrm{maj}}, \\ \text{danger region} : S_{\mathrm{maj}} - S_{\min}. \end{cases} \tag{8}$$

In the data selection phase, to widen the decision area of the minority class, $\boldsymbol{h}_i \in \boldsymbol{H}_{\min}$ is randomly selected as the seed sample, and its farthest neighbor $\boldsymbol{h}_j \in \boldsymbol{H}_{\min}$ is chosen as the auxiliary sample. In the data generation phase, four imputation rules are applied which are designed to generate synthetic samples in the safe region.

1. Direct interpolation. As shown in Fig. 4a, both the seed sample and auxiliary sample are located in the safe region, i.e., $\boldsymbol{h}_i, \boldsymbol{h}_j \in (S_{\min} - S_{\mathrm{maj}})$.

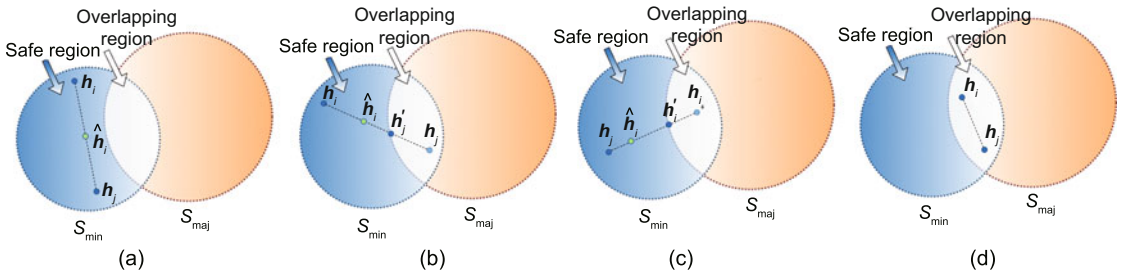

Fig. 4 Imputation rules of the retreating interpolation method: (a) direct interpolation; (b) auxiliary sample retreating interpolation; (c) seed sample retreating interpolation; (d) no interpolation

The synthetic sample is generated on the line between $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$. Eq. (9) represents this imputation rule:

$$\hat{\boldsymbol{h}} = \boldsymbol{h}_i + \lambda(\boldsymbol{h}_j - \boldsymbol{h}_i). \tag{9}$$

2. Auxiliary sample RI. As shown in Fig. 4b, the seed sample is located in the safe region, while the auxiliary sample is located in the overlapping region, i.e., $\boldsymbol{h}_i \in (S_{\min} - S_{\mathrm{maj}}), \boldsymbol{h}_j \in (S_{\min} \cap S_{\mathrm{maj}})$. In this case, $\boldsymbol{h}_j$ retreats back to the boundary of the safe region and overlapping region along the line from $\boldsymbol{h}_j$ to $\boldsymbol{h}_i$ and forms a shadow sample $\boldsymbol{h}'_j$. The newly generated samples can be depicted as

$$\hat{\boldsymbol{h}} = \boldsymbol{h}_i + \lambda(\boldsymbol{h}'_j - \boldsymbol{h}_i). \tag{10}$$

3. Seed sample RI. As shown in Fig. 4c, the seed sample is located in the overlapping region, while the auxiliary sample is located in the safe region, i.e., $\boldsymbol{h}_i \in (S_{\min} \cap S_{\mathrm{maj}}), \boldsymbol{h}_j \in (S_{\min} - S_{\mathrm{maj}})$. In this case, $\boldsymbol{h}_i$ retreats to the boundary of the safe region and overlapping region along the line from $\boldsymbol{h}_i$ to $\boldsymbol{h}_j$ and forms a shadow sample $\boldsymbol{h}'_i$. The newly generated samples can be depicted as

$$\hat{\boldsymbol{h}} = \boldsymbol{h}_j + \lambda(\boldsymbol{h}'_i - \boldsymbol{h}_j). \tag{11}$$

4. No interpolation. As shown in Fig. 4d, both the seed sample and auxiliary sample are located in the overlapping region, i.e., $\boldsymbol{h}_i, \boldsymbol{h}_j \in (S_{\min} \cap S_{\mathrm{maj}})$. To avoid generating noisy samples, no samples are generated in this case.

It is important to note that in very exceptional cases, due to the specific positions of the seed and auxiliary samples, the newly generated samples might fall into the overlapping region, as illustrated in Fig. 5. For such scenarios, direct interpolation is adopted. Considering the infrequency of such cases, coupled with the fact that the imputed positions remain close to the minority-class region, their impacts on the overall oversampling outcomes can be negligible. To illustrate the rarity of this example, we select five datasets to record the locations of sample generation and calculate the proportion of samples falling into the overlapping region relative to the total number of synthetic samples, as shown in Table 1. According to the experimental results, it can be observed that when both the root sample and auxiliary sample are located in the safe region, the probability of generated samples falling into the overlapping region is very low.

Finally, the OAT method maps the resulting dataset $\hat{H}$ back to the original space, i.e.,

$$\hat{\boldsymbol{X}}_{\mathcal{K}\times\mathcal{D}} = \hat{\boldsymbol{H}}_{\mathcal{K}\times\mathcal{L}}\boldsymbol{W}_{\mathcal{L}\times\mathcal{D}}, \tag{12}$$

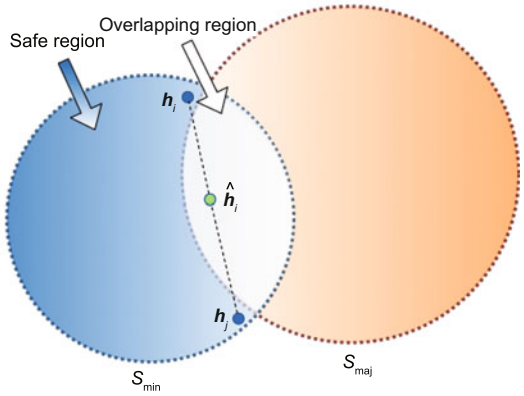

Fig. 5 Exceptional case

Table 1 Ratio of samples falling into the overlapping region corresponding to five datasets

Dataset	Number of samples		Ratio (%)
	Safe region	Overlapping region	
ecoli3	266	0	0.00
yeast1	624	2	0.32
vowel0	807	1	0.12
segment0	1645	5	0.30
abalone19	4101	9	0.22

where $\mathcal{K}$ is the number of synthetic MinCSPs. The process of RI is summarized in Algorithm 1.

Algorithm 1 Retreating interpolation

Require: $\boldsymbol{H}_{\min}$, the transformed representation of MinCSPs; $\boldsymbol{H}_{\mathrm{maj}}$, the transformed representation of MajCSPs; n_1, the number of MinCSPs; n_2, the number of MajCSPs

Ensure: $\hat{H} = \{\hat{\boldsymbol{h}}_1, \hat{\boldsymbol{h}}_2, ..., \hat{\boldsymbol{h}}_{n_2-n_1}\}$, the set of synthetic MinCSPs

Calculate $\boldsymbol{o}_{\min}$, $\boldsymbol{o}_{\mathrm{maj}}$, $r_{\min}$, and r_{maj}
Initialize $\hat{H}$ to the empty set, $k = 0$
while $k < n_2 - n_1$ **do**
  Randomly select $\boldsymbol{h}_i \in \boldsymbol{H}_{\min}$ as the seed sample
  Select $\boldsymbol{h}_i$'s farthest neighbor $\boldsymbol{h}_j$ as the auxiliary sample
  if $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ are both in the safe region **then**
    $\hat{\boldsymbol{h}}_k \leftarrow \boldsymbol{h}_i + \lambda(\boldsymbol{h}_j - \boldsymbol{h}_i)$, $\lambda \in [0, 1]$
    $k+ = 1$
  else if $\boldsymbol{h}_i$ is in the safe region and $\boldsymbol{h}_j$ is in the overlapping region **then**
    Let $\boldsymbol{h}_j$ retreat along the line from $\boldsymbol{h}_j$ to $\boldsymbol{h}_i$ until $||\boldsymbol{h}_j - \boldsymbol{o}_{\mathrm{maj}}|| = r_{\mathrm{maj}}$, and form $\boldsymbol{h}'_j$
    $\hat{\boldsymbol{h}}_k \leftarrow \boldsymbol{h}_i + \lambda(\boldsymbol{h}'_j - \boldsymbol{h}_i)$, $\lambda \in [0, 1]$
    $k+ = 1$
  else if $\boldsymbol{h}_i$ is in the overlapping region and $\boldsymbol{h}_j$ is in the safe region **then**
    Let $\boldsymbol{h}_i$ retreat along the line from $\boldsymbol{h}_i$ to $\boldsymbol{h}_j$ until $||\boldsymbol{h}_i - \boldsymbol{o}_{\mathrm{maj}}|| = r_{\mathrm{maj}}$, and form $\boldsymbol{h}'_i$
    $\hat{\boldsymbol{h}}_k \leftarrow \boldsymbol{h}_j + \lambda(\boldsymbol{h}'_i - \boldsymbol{h}_j)$, $\lambda \in [0, 1]$
    $k+ = 1$
  end if
  Add $\hat{\boldsymbol{h}}_k$ to $\hat{H}$
end while

4 Experiments

This section describes a series of experiments that were conducted to demonstrate the algorithmic convergence, imputation visualization, and effectiveness of OM-SMOTE. All experiments were carried out using Python on a workstation equipped with an Intel® Core™ i7-11700 2.50 GHz CPU and 16 GB of memory.

4.1 Parameter selection and convergence validation

In this experiment, we first verified the influence of parameter selection on the performance and convergence of the OM-SMOTE algorithm. The tunable parameters include the weight α of two loss functions and the dimension of hidden space $\mathcal{L}$. The weight α controls the trade-off between two loss functions for data reconstruction and data classification during gradient descent. The parameter $\mathcal{L}$ determines the model's complexity: a small value may result in an underfitting model, while a large value may produce an overly complex model. For this initial experiment, the ecoli3 dataset was selected and the SVM classifier (https://scikit-learn.org/stable/modules/svm.html) in the Scikit-Learn toolkit was trained to validate OM-SMOTE's performance. The dataset was randomly partitioned into two parts; i.e., 70% of data were used for training and the remaining 30% for testing.

For this experiment, a parameter $\mathcal{M}$ was introduced, called the multiple of ascending dimensions, which was used as a multiplying factor to select a value for $\mathcal{L}$ as $\mathcal{L} = \mathcal{M}\mathcal{D}$. The set of values to be tested for parameter α was set to $\alpha \in \{0, 0.1, 0.2, ..., 1.0\}$ and $\mathcal{M}$ was set to $\mathcal{M} \in \{1, 2, 3, ..., 50\}$. By applying the grid search strategy, as shown in Fig. 6, it was found that the optimal parameter values were $\{\alpha, \mathcal{M}\} = \{0.4, 20\}$ for OM-SMOTE's training on the ecoli3 dataset.

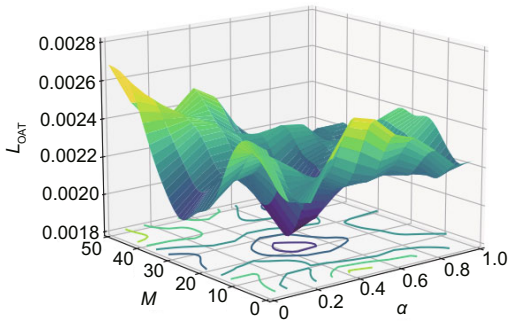

Fig. 6 Impact of parameter pair (α, $\mathcal{M}$) on L_{OAT}

We first verified the effect of α on the convergence of loss functions by fixing $\mathcal{M} = 20$ and changing α to $\{0, 0.1, 0.2, ..., 1.0\}$. Fig. 7 shows the results of L_{OAT}, L_{MSE}, and L_{CE}. As observed in Fig. 7a, the overall trend of L_{OAT} is that loss first decreased and then stabilized for all values of α. This demonstrates that the structure of the proposed OAT method can make the loss function converge. As shown in Fig. 7b, when α was set to 0, L_{MSE} had no impact on gradient descent. In addition, except for $\alpha = 0$, all the losses decreased below 0.01 after converging. The experimental results demonstrated the effectiveness of the reconstruction network. As depicted in Fig. 7c, when $\alpha = 1.0$, L_{CE} had no impact on gradient descent, and except for $\alpha = 1.0$, all the losses decreased rapidly and then converged.

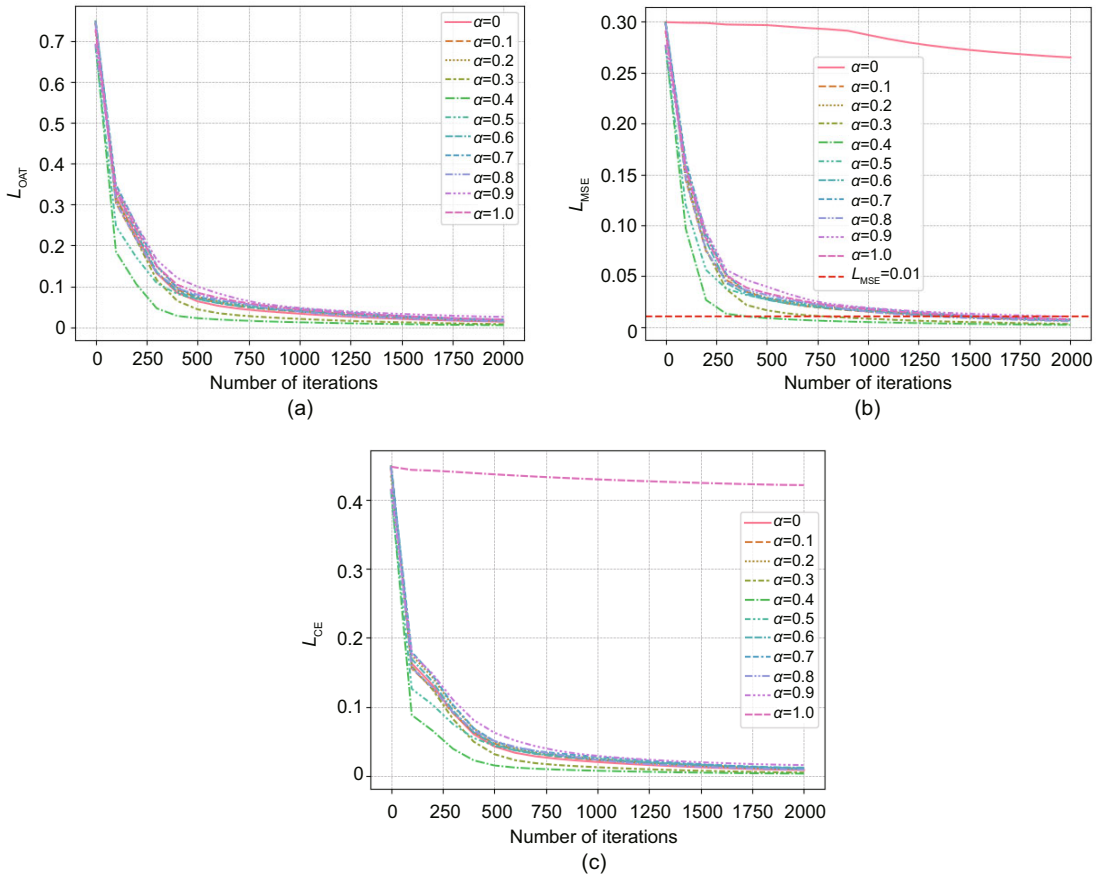

Fig. 7 Impact of α on the loss function: (a) L_{OAT}; (b) L_{MSE}; (c) L_{CE}

It can also be seen that the loss decreased more quickly for $\alpha = 0.4$. Then, we verified the effect of the parameter $\mathcal{M}$ on OM-SMOTE's performance by using $\alpha = 0.4$ and varied $\mathcal{M}$ with the values $\{1, 2, 3, ..., 50\}$. Results are shown in Fig. 8. The AUC, G-mean, F1, and accuracy curves showed that the OM-SMOTE performance tended to increase and then decrease as the multiplying factor $\mathcal{M}$ increased, and that OM-SMOTE achieved optimal performance for $\mathcal{M} = 20$.

Then, we verified the convergence of the OM-SMOTE algorithm as the number of iterations increased. We set the parameters as $\{\alpha, \mathcal{M}\} = \{0.4, 20\}$, and used different numbers of iterations $\{0, 100, 200, ..., 5000\}$ to observe the convergence of OM-SMOTE. The results are shown in Fig. 9. It can be seen that as the number of iterations increased, AUC, G-mean, F1, and accuracy generally first increased and then stabilized. These experimental results demonstrated the convergence of the OM-SMOTE algorithm.

4.2 Imputation visualization

In this subsection, we performed experiments to visually demonstrate the oversampling results of the OM-SMOTE algorithm, the data-transforming process of the OAT method, and the interpolation results of the RI mechanism in the hidden space. The experiments were conducted using a two-dimensional (2D) imbalanced dataset, which can be downloaded from the OM-SMOTE Data Sets folder at https://pan.baidu.com/s/1X3UMInVZUDOzcogO3a5_MA using the extraction code 69da. The dataset consists of 240 MajCSPs and 60 MinCSPs. Fig. 10a shows the distribution and kernel density estimate (KDE) for this dataset. As can be observed, the original data had some degree of overlapping, making the classes linearly inseparable in the original space. Fig. 10b gives the oversampling results produced by the SMOTE algorithm. It can be seen that a portion of the synthetic MinCSPs were generated in the overlapping region, which aggravates the

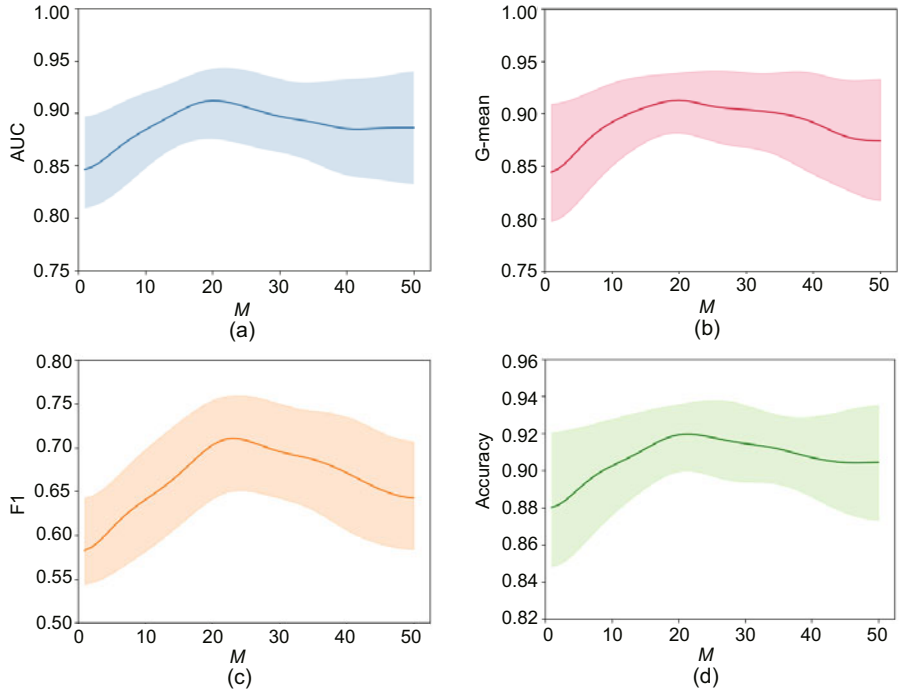

Fig. 8 Effect of $\mathcal{M}$ on the performance of the OM-SMOTE algorithm: (a) AUC; (b) G-mean; (c) F1; (d) accuracy

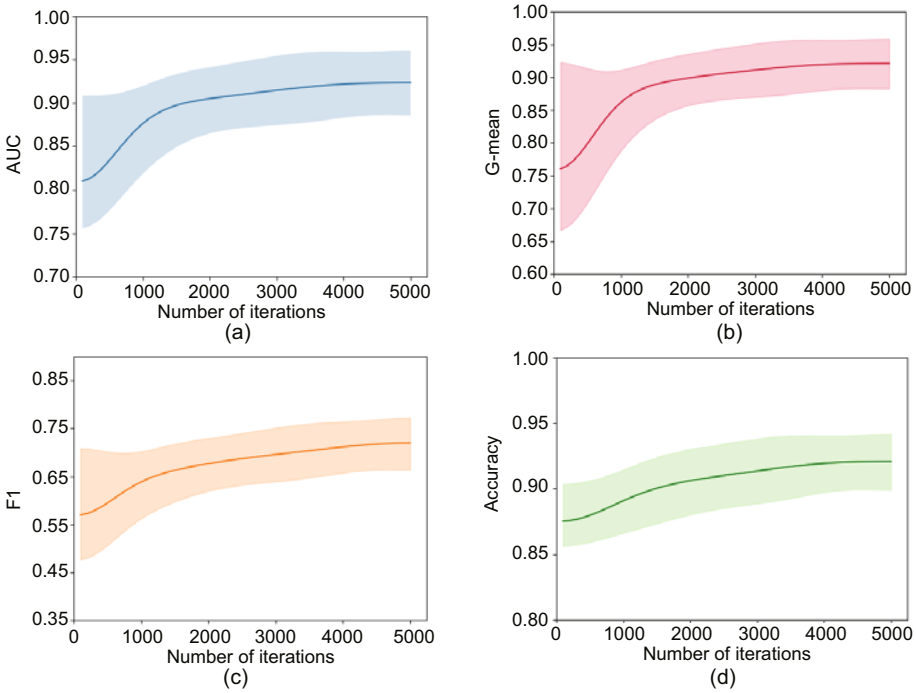

Fig. 9 Convergence validation of the OM-SMOTE algorithm: (a) AUC; (b) G-mean; (c) F1; (d) accuracy

overlapping phenomenon. This is because SMOTE does not take into account the information about the distribution of MajCSPs. Because some synthetic MinCSPs fall into the overlapping region, a mixture of MinCSPs and MajCSPs is created and thus the difficulty of training a classifier increases.

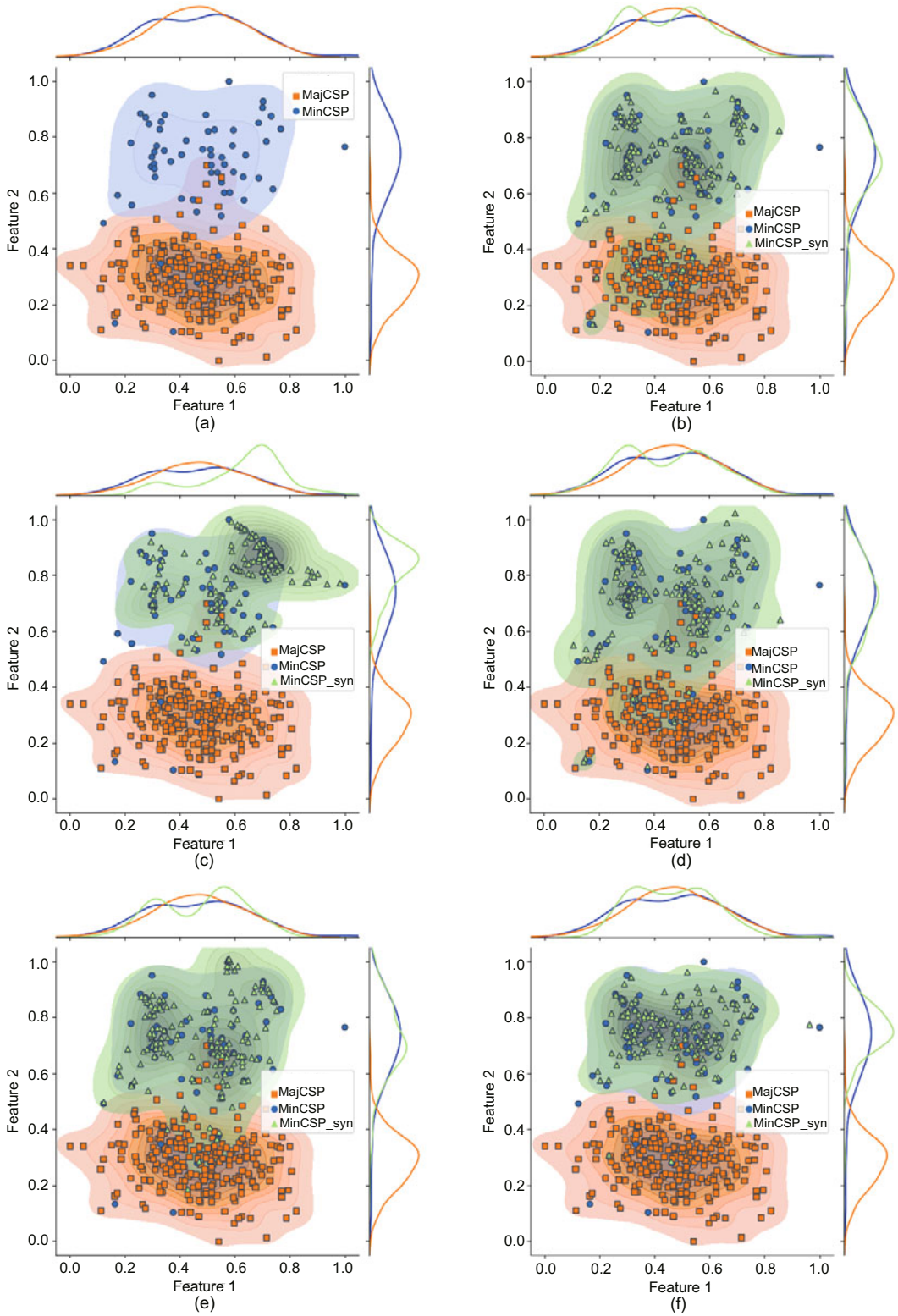

Fig. 10 Original dataset and oversampled datasets by state-of-the-art SMOTE-based algorithms with one-dimensional kernel density estimates displayed along the feature: (a) original dataset; (b) oversampled dataset by SMOTE; (c) oversampled dataset by *k*-means SMOTE; (d) oversampled dataset by G-SMOTE; (e) oversampled dataset by LoRAS; (f) oversampled dataset by OM-SMOTE

In addition, SMOTE uses KNN rules to select the auxiliary samples and considers only local information: the overall distribution of MinCSPs is ignored. Hence, the synthetic samples tend to be generated in areas where the MinCSPs are more concentrated. As shown in Fig. 10b, the KDE of synthetic MinCSPs differed significantly from that of the original MinCSPs.

We also compared the interpolation performance of OM-SMOTE with those of SMOTE-based algorithms from the perspective of probability distribution. The comparative results are shown in Figs. 10c and 10d. k-means SMOTE (Douzas et al., 2018) avoids generating data in overlapping regions by using clustering. However, due to its application of SMOTE solely within minority-class clusters, it still exhibited issues of exacerbating intra-class imbalance and difficult-to-generate high-quality synthetic samples. As shown in Fig. 10c, KDE of synthetic MinCSPs deviated significantly from the original data. Figs. 10d and 10e show that both G-SMOTE (Douzas and Bacao, 2019) and LoRAS (Bej et al., 2021) failed to address the problem of generating data in overlapping regions. The KDE graphs show that the overlapping region was exacerbated. It can also be seen that the synthetic MinCSPs generated by these algorithms tended to be located in dense areas containing MinCSPs. Hence, the intra-class imbalance will be further exacerbated by the significant increase in synthetic MinCSPs. Fig. 10f presents the oversampling results of the proposed OM-SMOTE algorithm. In terms of space distribution, the OM-SMOTE algorithm avoided generating data in overlapping regions as much as possible and generated samples that were more evenly distributed in the MinCSP region and had approximately consistent KDE graphs with the original data.

Then we visualized the data transforming process of the OAT method. For the aforementioned 2D dataset, we used the OAT method to transform the original data into a four-dimensional hidden space. Figs. 11a and 11b show the data distribution for the feature 1 and feature 2 dimensions, and feature 3 and feature 4 dimensions, respectively. The data that were linearly indistinguishable in the original space can now be better separated in the higher dimensional space. We found that the data overlapping was alleviated in the higher dimensional space, which validates the ability of the OAT algorithm to map the data into a more separable space.

Finally, we visualized the interpolation results of the RI mechanism in the hidden space. The synthetic data generated using the RI mechanism are presented in Fig. 12. For each dimension, the KDE curves for synthetic MinCSPs (green curve) almost fitted the original data (blue curve) for each feature. This verified that the synthetic MinCSPs generated by the RI mechanism have a probability distribution that is consistent with that of the original data. Also, the interpolation mechanism avoided generating data in overlapping regions. The aforementioned experiments demonstrated that the OM-SMOTE algorithm is reasonable for generating high-quality synthetic data and has great potential to improve a classifier's imbalanced learning performance.

4.3 Effectiveness of the OM-SMOTE

In this experiment, we compared the imbalanced classification performance of the OM-SMOTE algorithm with 11 SMOTE-based algorithms, i.e., SMOTE (Chawla et al., 2002), Borderline-SMOTE (Han et al., 2005), ADASYN (He HB et al., 2008), Random-SMOTE (Dong and Wang, 2011), SMOTE-IPF (Sáez et al., 2015), AMSCO (Li JY et al., 2018), k-means SMOTE (Douzas et al., 2018), G-SMOTE (Douzas and Bacao, 2019), SyMProD (Kunakorntum et al., 2020), LoRAS (Bej et al., 2021), and AE-ELM-SynMin (He YL et al., 2022), based on four typical classifiers, i.e., SVM, naive Bayes (https://scikit-learn.org/stable/modules/naive_bayes.html), DT (https://scikit-learn.org/stable/modules/tree.html), and LR (https://scikit-learn.org/stable/modules/linear_model.html). The implementation of OM-SMOTE is shared publicly on the GitHub platform (https://github.com/luxuan123123/OM-SMOTE/).

We selected 32 imbalanced datasets, listed in Table 2, to conduct the experimental comparison. These experimental datasets can be obtained from the KEEL repository.

For each dataset, we used a cross-validation strategy to evaluate the performance of each algorithm, where the performance was measured with AUC, G-mean, F1, and accuracy. The experimental steps are as follows:

Step 1: divide the dataset into two parts, 70% for training and 30% for testing.

Step 2: perform oversampling or hybrid-sampling on the training set using OM-SMOTE and other SMOTE-based algorithms.

Step 3: train a classifier based on the oversampled or hybrid-sampled dataset generated by each method in step 2.

Step 4: calculate the AUC, G-mean, F1, and accuracy of the classifier generated by each method based on the testing set.

Step 5: repeat steps 1–4 20 times and calculate the average AUC, G-mean, F1, and accuracy values.

Tables S1–S16 in the supplementary materials show the comparison results for the naive Bayes, SVM, DT, and LR classifiers. It can be seen that

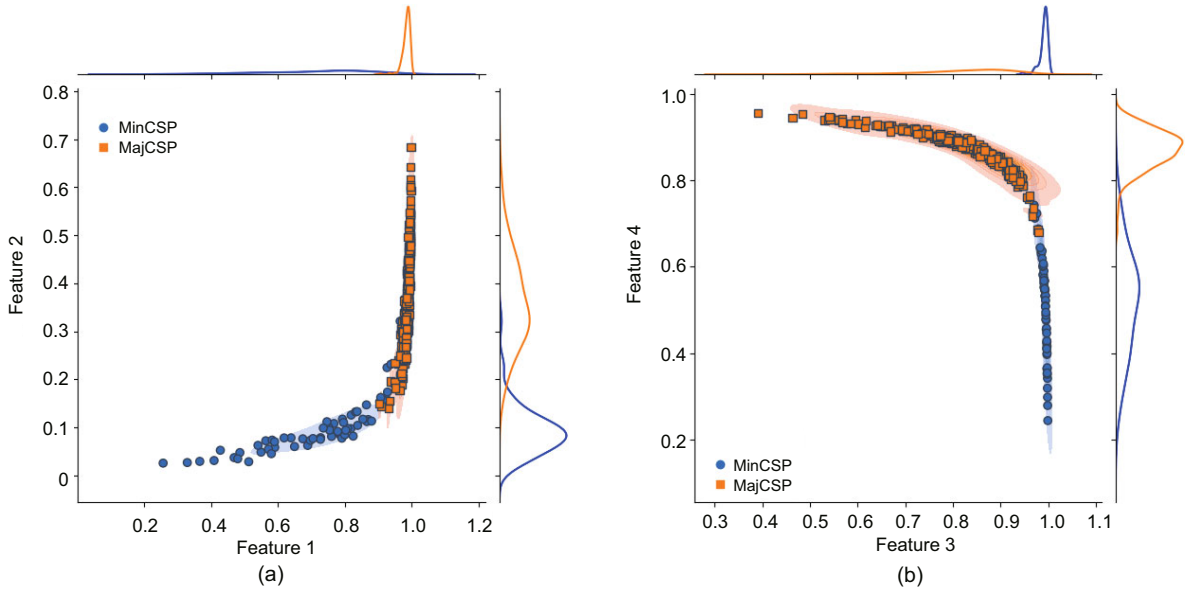

Fig. 11 Data distribution in the higher-dimensional space with one-dimensional kernel density estimates displayed along the feature: (a) data distribution for feature 1 and feature 2; (b) data distribution for feature 3 and feature 4

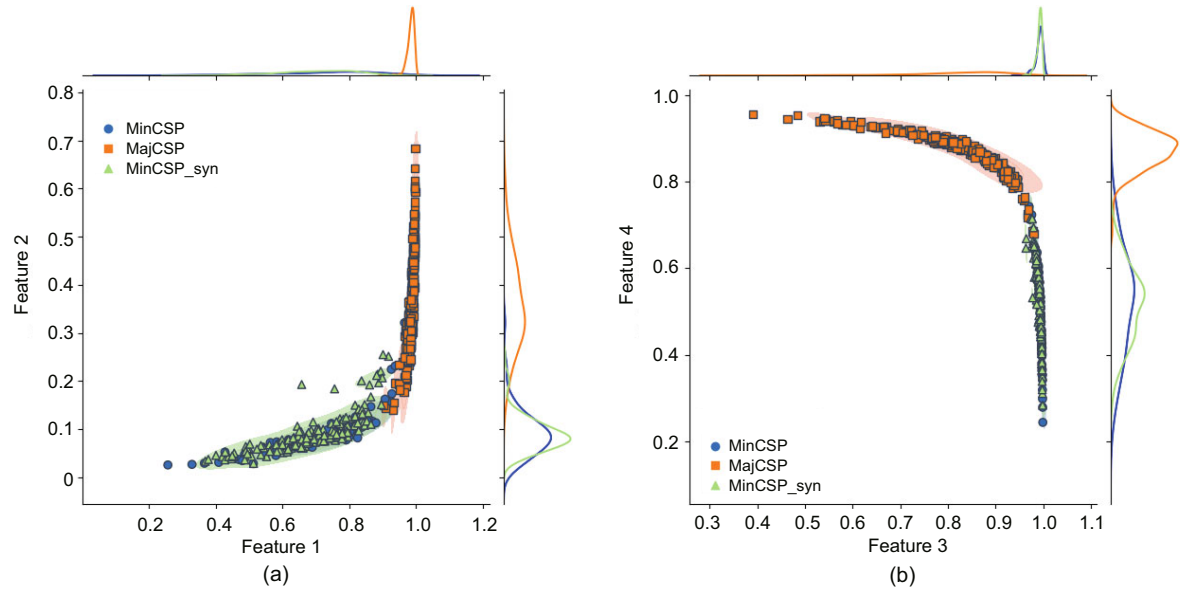

Fig. 12 Applying retreating interpolation in the hidden space with one-dimensional kernel density estimates displayed along the feature: (a) data distribution for feature 1 and feature 2; (b) data distribution for feature 3 and feature 4 (References to color refer to the online version of this figure)

Table 2 Details of 32 imbalanced datasets

Dataset	Number of features	Number of samples	Imbalance ratio (%)
ecoli1	7	336	3.36
ecoli2	7	336	5.46
ecoli3	7	336	8.60
ecoli4	7	336	15.80
yeast1	8	1484	2.46
yeast3	8	1484	8.10
yeast5	8	1484	32.73
yeast-0-5-6-7-9_vs_4	8	528	9.35
yeast-1-2-8-9_vs_7	8	947	30.57
yeast-1_vs_7	7	459	14.30
yeast-2_vs_8	8	482	23.10
glass0	9	214	2.06
glass2	9	214	11.59
glass4	9	214	15.47
glass6	9	214	6.38
glass-0-1-6_vs_2	9	192	10.29
glass-0-1-6_vs_5	9	184	19.44
glass-0-4_vs_5	9	92	9.22
vowel0	13	988	9.98
new-thyroid1	5	215	5.14
shuttle-C0_vs_C4	9	1829	13.87
segment0	19	2308	6.02
vehicle0	18	846	3.25
page-blocks-1-3_vs_4	10	472	15.86
abalone9_18	8	731	16.40
abalone19	8	4174	129.44
abalone-3_vs11	8	502	32.47
abalone-21_vs_8	8	581	40.50
wisconsin	9	683	1.86
haberman	3	306	2.78
pima	8	768	1.87
winequality-red-3_vs_5	11	691	68.10

for the naive Bayes and DT classifiers, OM-SMOTE achieved better average scores in terms of AUC, G-mean, F1, and accuracy than the 11 SMOTE-based algorithms. For the SVM and LR classifiers, OM-SMOTE achieved better average scores in terms of AUC, G-mean, and F1 than the 11 SMOTE-based algorithms.

For accuracy, OM-SMOTE ranked third in SVM and LR (only k-means SMOTE and SyMProD had higher scores). We conducted an analysis of this phenomenon, focusing on the SyMProD algorithm, and found that although SyMProD performed better in terms of accuracy, it had the worst AUC and G-mean as shown in Tables S5–S8. For the yeast-1-2-8-9_vs_7, glass2, glass-0-1-6_vs_2, abalone19, and winequality-red-3_vs_5 datasets, SyMProD obtained a 0.500 ± 0.000 score in AUC and 0.000 ± 0.000 in G-mean and F1. This phenomenon occurred when the classifier trained with the data generated by SyMProD predicted all MinCSPs as MajCSPs (He HB and Garcia, 2009). That is to say, the SyMProD algorithm failed to handle the imbalanced classification problems in some datasets when using the SVM and LR classifiers.

We used critical difference (CD) diagrams (Demšar, 2006) to perform a statistical analysis of the experimental results. For a given significance level of 0.1, the calculated CD value was 2.731. The CD value is a statistical measure used in multiple comparisons to determine the significance of differences in means between different groups. If the difference in means between two groups exceeds the CD value, it can be considered statistically significant. The CD diagram figures are shown in Figs. S1–S4 in the supplementary materials, where an interval was drawn around the average rank of the OM-SMOTE algorithm. Then, any algorithm having a rank outside that area was significantly different from the OM-SMOTE algorithm. In this type of diagram, an algorithm that appeared more to the right has a smaller average rank, which means that it has better performance. Figs. S1 and S3 show the CD diagrams of OM-SMOTE and 11 SMOTE-based algorithms for the naive Bayes and DT classifiers, respectively. We can see that the ranks of the OM-SMOTE algorithm corresponding to the AUC, G-mean, F1, and accuracy were obviously smaller than those of the 11 SMOTE-based algorithms. Figs. S2 and S4 show the CD diagrams of OM-SMOTE and 11 SMOTE-based algorithms for SVM and LR classifiers, respectively. We can see that the ranks of the OM-SMOTE algorithm corresponding to the AUC, G-mean, and F1 were obviously smaller than those of the 11 SMOTE-based algorithms. In terms of accuracy, for the SVM classifier, OM-SMOTE ranked second, while SyMProD ranked first. For the LR classifier, OM-SMOTE ranked third, while k-means SMOTE and SyMProD achieved higher ranks. According to Figs. S1–S4, we can conclude that for naive Bayes and DT, OM-SMOTE is significantly better than the 11 algorithms in terms of AUC, G-mean, F1, and accuracy; for SVM and LR, OM-SMOTE was significantly better than other algorithms in AUC, G-mean, and F1 score.

The experimental and statistical results indicated that OM-SMOTE is a viable algorithm to handle imbalanced classification problems.

5 Conclusions and future work

In this paper, we proposed a novel overlapping minimization SMOTE (OM-SMOTE) algorithm for imbalanced classification problems. OM-SMOTE was applied in two steps. First, a novel overlapping alleviation transformation (OAT) method transformed the original data into a more separable space, and then a retreating interpolation (RI) mechanism was used to avoid generating new data in the overlapping region. Through extensive experiments, we have demonstrated the algorithm convergence, imputation visualization, and effectiveness of the OM-SMOTE algorithm. Moreover, a statistical analysis has shown that OM-SMOTE can help classifiers perform significantly better for imbalanced classification tasks than 11 SMOTE-based algorithms.

In the future, we will focus on three directions. First, we will generalize this algorithm to multi-class classification problems. Second, we will consider interpolation methods based on probability distributions to further improve the consistency between the probability distributions of synthetic data and original data. Third, we will investigate the integration of this algorithm with random sample partition for big data processing (Salloum et al., 2019).

Contributors

Yulin HE was responsible for conceptualization, methodology, and writing the first draft. Xuan LU helped writing the first draft, validation, data curation, and data analysis. Philippe FOURNIER-VIGER edited the first draft. Joshua Zhexue HUANG was in charge of investigation and supervision.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Bank D, Koenigstein N, Giryes R, 2020. Autoencoders. https://arxiv.org/abs/2003.05991

Barua S, Islam M, Murase K, 2011. A novel synthetic minority oversampling technique for imbalanced data set learning. Proc 18th Int Conf on Neural Information Processing, p.735-744. https://doi.org/10.1007/978-3-642-24958-7_85

Bej S, Davtyan N, Wolfien M, et al., 2021. LoRAS: an oversampling approach for imbalanced datasets. *Mach Learn*, 110(2):279-301. https://doi.org/10.1007/s10994-020-05913-4

Bellinger C, Japkowicz N, Drummond C, 2015. Synthetic oversampling for advanced radioactive threat detection. IEEE 14th Int Conf on Machine Learning and Applications, p.948-953. https://doi.org/10.1109/ICMLA.2015.58

Bellinger C, Drummond C, Japkowicz N, 2016. Beyond the boundaries of SMOTE. Proc 13th Pacific-Asia Conf on Knowledge Discovery and Data Mining, p.248-263. https://doi.org/10.1007/978-3-319-46128-1_16

Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C, 2009. Safe-Level-SMOTE: safe-level-synthetic minority oversampling technique for handling the class imbalanced problem. Proc 13th Pacific-Asia Conf on Knowledge Discovery and Data Mining, p.475-482. https://doi.org//10.1007/978-3-642-01307-2_43

Chawla NV, Bowyer KW, Hall LO, et al., 2002. SMOTE: synthetic minority over-sampling technique. *J Artif Intell Res*, 16:321-357. https://doi.org/10.1613/jair.953

Cover TM, 1965. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Trans Electron Comput*, EC-14(3):326-334. https://doi.org/10.1109/PGEC.1965.264137

Demšar J, 2006. Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res*, 7:1-30. https://doi.org/10.1007/s10846-005-9016-2

Dong YJ, Wang XH, 2011. A new over-sampling approach: Random-SMOTE for learning from imbalanced data sets. Proc 5th Int Conf on Knowledge Science, Engineering and Management, p.343-352. https://doi.org/10.1007/978-3-642-25975-3_30

Douzas G, Bacao F, 2019. Geometric SMOTE a geometrically enhanced drop-in replacement for SMOTE. *Inform 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. *Inform Sci*, 465:1-20. https://doi.org/10.1016/j.ins.2018.06.056

Douzas G, Rauch R, Bacao F, 2021. G-SOMO: an oversampling approach based on self-organized maps and geometric SMOTE. *Expert Syst Appl*, 183:115230. https://doi.org/10.1016/j.eswa.2021.115230

Fernández A, Garcia S, Herrera F, et al., 2018. SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *J Artif Intell Res*, 61:863-905. https://doi.org/10.1613/jair.1.11192

Gosain A, Sardana S, 2019. Farthest SMOTE: a modified SMOTE approach. In: Behera HS, Nayak J, Naik B, et al. (Eds.), Computational Intelligence in Data Mining. Springer, Singapore, p.309-320. https://doi.org/10.1007/978-981-10-8055-5_28

Gu Q, Cai ZH, Zhu L, 2009. Classification of imbalanced data sets by using the hybrid re-sampling algorithm based on Isomap. Proc 4th Int Symp on Intelligence Computation and Applications, p.287-296. https://doi.org/10.1007/978-3-642-04843-2_31

Guo HX, Li YJ, Shang J, et al., 2017. Learning from class-imbalanced data: review of methods and applications. *Expert Syst Appl*, 73:220-239. https://doi.org/10.1016/j.eswa.2016.12.035

Han H, Wang WY, Mao BH, 2005. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. Proc Int Conf on Intelligent Computing, p.878-887. https://doi.org/10.1007/11538059_91

Hand DJ, Till RJ, 2001. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Mach Learn*, 45(2):171-186. https://doi.org/10.1023/A:1010920819831

He HB, Garcia EA, 2009. Learning from imbalanced data. *IEEE Trans Knowl Data Eng*, 21(9):1263-1284. https://doi.org/10.1109/TKDE.2008.239

He HB, Bai Y, Garcia EA, et al., 2008. ADASYN: adaptive synthetic sampling approach for imbalanced learning. Proc IEEE Int Joint Conf on Neural Networks, p.1322-1328. https://doi.org/10.1109/IJCNN.2008.4633969

He YL, Xu SS, Huang JZ, 2022. Creating synthetic minority class samples based on autoencoder extreme learning machine. *Patt Recogn*, 121:108191. https://doi.org/10.1016/j.patcog.2021.108191

Kovács G, 2019. SMOTE-variants: a Python implementation of 85 minority oversampling techniques. *Neurocomputing*, 366:352-354. https://doi.org/10.1016/j.neucom.2019.06.100

Kunakorntum I, Hinthong W, Phunchongharn P, 2020. A synthetic minority based on probabilistic distribution (SyMProD) oversampling for imbalanced datasets. *IEEE Access*, 8:114692-114704. https://doi.org/10.1109/ACCESS.2020.3003346

Li JY, Fong S, Wong RK, et al., 2018. Adaptive multi-objective swarm fusion for imbalanced data classification. *Inform Fus*, 39:1-24. https://doi.org/10.1016/j.inffus.2017.03.007

Li W, Zhao SS, Chen Y, et al., 2022. State of China's climate in 2021. *Atmos Ocean Sci Lett*, 15(4):100211. https://doi.org/10.1016/j.aosl.2022.100211

Lim SK, Tran NT, Cheung NM, 2018. DOPING: generative data augmentation for unsupervised anomaly detection with GAN. Proc IEEE Int Conf on Data Mining, p.1122-1127. https://doi.org/10.1109/ICDM.2018.00146

Lipton ZC, Elkan C, Naryanaswamy B, 2014. Optimal thresholding of classifiers to maximize F1 measure. Proc Joint European Conf on Machine Learning and Knowledge Discovery in Databases, p.225-239. https://doi.org/10.1007/978-3-662-44851-9_15

Mathew J, Luo M, Pang CK, et al., 2015. Kernel-based SMOTE for SVM classification of imbalanced datasets. Proc 41st Annual Conf of the IEEE Industrial Electronics Society, p.1127-1132. https://doi.org/10.1109/IECON.2015.7392251

Moulaei K, Shanbehzadeh M, Mohammadi-Taghiabad Z, et al., 2022. Comparing machine learning algorithms for predicting COVID-19 mortality. *BMC Med Inform Decis Mak*, 22(1):2. https://doi.org/10.1186/s12911-021-01742-0

Pérez-Ortiz M, Gutiérrez PA, Tino P, et al., 2016. Oversampling the minority class in the feature space. *IEEE Trans Neur Netw Learn Syst*, 27(9):1947-1961. https://doi.org/10.1109/TNNLS.2015.2461436

Sáez JA, Luengo J, Stefanowski J, et al., 2015. SMOTE-IPF: addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Inform Sci*, 291:184-203. https://doi.org/10.1016/j.ins.2014.08.051

Sáez JA, Galar M, Krawczyk B, 2019. Addressing the overlapping data problem in classification using the One-vs-One decomposition strategy. *IEEE Access*, 7:83396-83411. https://doi.org/10.1109/ACCESS.2019.2925300

Salloum S, Huang JZ, He YL, 2019. Random sample partition: a distributed data model for big data analysis. *IEEE Trans Ind Inform*, 15(11):5846-5854. https://doi.org/10.1109/TII.2019.2912723

Sun YM, Kamel MS, Wang Y, 2006. Boosting for learning multiple classes with imbalanced class distribution. Proc 6th Int Conf on Data Mining, p.592-602. https://doi.org/10.1109/ICDM.2006.29

Tang W, Mao KZ, Mak LO, et al., 2010. Classification for overlapping classes using optimized overlapping region detection and soft decision. Proc 13th Int Conf on Information Fusion, p.1-8. https://doi.org/10.1109/ICIF.2010.5712008

List of supplementary materials

Tables S1–S4 Comparison results of AUC, G-mean, F1, and accuracy for imbalanced classification problems based on the naive Bayes classifier

Tables S5–S8 Comparison results of AUC, G-mean, F1, and accuracy for imbalanced classification problems based on the SVM classifier

Tables S9–S12 Comparison results of AUC, G-mean, F1, and accuracy for imbalanced classification problems based on the DT classifier

Tables S13–S16 Comparison results of AUC, G-mean, F1, and accuracy for imbalanced classification problems based on the LR classifier

Figs. S1–S4 Critical difference diagrams comparing OM-SMOTE with 11 SMOTE-based algorithms on the naive Bayes, SVM, DT, and LR classifiers