International Journal of Machine Learning and Cybernetics (2022) 13:735–750
https://doi.org/10.1007/s13042-021-01321-9

ORIGINAL ARTICLE

Imbalanced data classification based on diverse sample generation and classifier fusion

Junhai Zhai[1] · Jiaxing Qi[1] · Sufang Zhang[2]

Received: 11 July 2020 / Accepted: 30 March 2021 / Published online: 12 April 2021

Abstract
Class imbalance problems are pervasive in many real-world applications, yet classifying imbalanced data remains to be a very challenging task in machine learning. SMOTE is the most influential oversampling approach. Based on SMOTE, many variants have been proposed. However, SMOTE and its variants have three drawbacks: (1) the probability distribution of the minority class samples is not considered; (2) the generated minority samples lack diversity; (3) the generated minority class samples overlap severely when oversampled many times for balancing with majority class samples. In order to overcome these three drawbacks, a generative adversarial network (GAN) based framework is proposed in this paper. The framework includes an oversampling method and a two-class imbalanced data classification approach. The oversampling method is based on an improved GAN model, and the classification approach is based on classifier fusion via fuzzy integral, which can well model the interactions among the base classifiers trained on the balanced data subsets constructed by the proposed oversampling method. Extensive experiments are conducted to compare the proposed methods with related methods on 5 aspects: MMD-score, Silhouette-score, F-measure, G-means, and AUC-area. The experimental results demonstrate that the proposed methods are more effective and efficient than the compared approaches.

Keywords Imbalanced data classification · Oversampling · Generative adversarial network · Diverse sample generation · Classifier fusion

1 Introduction

The class imbalance problem was originally proposed by Japkowicz [1]. It refers to the classification scenario where one class is represented by a large number of samples while the other is represented by only a few. Class imbalance problems are quite pervasive in many real-world applications, such as software defect prediction [2], machinery fault diagnosis [3], spam filtering [4], and so on. Class imbalance problems include two-class imbalance problems and multi-class imbalance problems. Since most existing solutions for multi-class imbalance problems first use class decomposition schemes to divide a multi-class problem into multiple two-class problems, and then to conquer each two-class imbalance subproblem [5, 6], this paper focuses on the two-class imbalance problem. In the two-class imbalance problem, the minority class is also called the positive class, while the majority class is also called the negative class. In the past two decades, many solutions to two-class imbalance problem have been proposed. SMOTE is the most influential oversampling method [7], which balances imbalanced dataset by generating synthetic positive class samples on the line between each positive class sample and its k nearest neighbors with same class. SMOTE and its variants have the following three drawbacks due to the mechanism by which they generate synthetic samples:

1. the probability distribution of the minority class samples is not considered;
2. the generated minority samples lack diversity;

✉ Junhai Zhai
mczjh@126.com

1 Hebei Key Laboratory of Machine Learning and Computational Intelligence, College of Mathematics and Information Science, Hebei University, Baoding 071002, Hebei, China

2 Hebei Branch of China Meteorological Administration Training Centre, China Meteorological Administration, Baoding 071000, Hebei, China

3. the generated minority class samples overlap heavily when oversample many times for balancing with majority class samples.

In order to overcome the three drawbacks, inspired by the idea of generative adversarial network (GAN) [8], we propose a framework which includes an oversampling method and a two-class imbalanced data classification approach based on classifier fusion via fuzzy integral. The main contributions of this paper include the following three folds:

1. We propose an oversampling method which is based on an improved GAN model. The improvement lies in introducing a regularization term of intra-class divergence into the loss function of the GAN, and replacing the discriminator of GAN with a classifier whose output is a vector with three entries: the probabilities that a predicted sample belongs to majority class, minority class, or generated sample.
2. Based on the proposed oversampling method, we propose a two-class imbalanced data classification approach based on classifier fusion via fuzzy integral. Fuzzy integral can well model the interactions among the base classifiers which are not independent, since all balanced data subsets used for training the base classifiers include the oversampled positive class samples. The proposed ensemble approach can enhance the classification accuracy of the positive class samples.
3. Extensive experiments are conducted to compare the proposed methods with related methods including 11 SMOTE related and 4 GAN related state-of-the-art approaches on 5 aspects: MMD-score, Silhouette-score, F-measure, G-means, and AUC-area. The experimental results demonstrate that the proposed methods are more effective and efficient than the compared approaches.

The rest of this paper is organized as follows. In Sect. 2, we review the works related to two-class imbalanced data classification. In Sect. 3, we describe the details of the proposed methods. In Sect. 4, the experimental results and analyses are presented. At last, we conclude our work in the Sect. 5.

2 Related works

Many methods have been proposed by different researchers for addressing two-class imbalanced data classification. These methods can be classified into three categories: data-level methods, algorithm-level methods, and ensemble methods. Considering that this paper focuses on the data-level and ensemble method, we only provide a brief review of algorithm-level methods, as a comprehensive review of algorithm-level approach can be found in [9, 10].

The basic idea of the algorithm-level methods is to modify the existing classification algorithms to adapt to the scenario of imbalanced data classification. The most common strategy of modification is to introduce cost sensitive mechanism to traditional classification algorithms. The pioneering work of cost sensitive methods for the class imbalance problem was presented by Sun et al. [11]. They introduced cost items into the famous ensemble algorithm AdaBoost, and proposed the AdaC algorithm family. Other representative works published in recent year are reported in [12–14]. Khan et al. [12] proposed a cost sensitive deep neural network model which can automatically learn good features [15–19] from imbalanced data by jointly optimizing the class correlation losses and network parameters. Tao et al. [13] proposed a self-adaptive cost weights-based support vector machine (SVM), and a cost-sensitive ensemble approach for imbalanced data classification. Wang et al. [14] proposed cost sensitive fuzzy multiple kernel learning method for addressing the imbalanced problem by introducing fuzzy memberships to characterize the feature of imbalanced data. The proposed method obtained more favorable classification performances on imbalanced datasets.

The basic idea of the data-level methods is to preprocess the original imbalanced dataset for balancing the distribution of samples in two classes by undersampling majority samples or oversampling minority samples. Some empirical comparisons demonstrate that oversampling is much more effective than undersampling [2, 20–22]. Among the oversampling methods, SMOTE [23] is the most influential oversampling approach. Since from SMOTE was proposed in 2002, many oversampling approaches have been proposed in the past 18 years. Based on *k*-means clustering and SMOTE, Douzas et al. [24] proposed an oversampling method which can avoid the generation of noise and effectively overcome imbalances between and within classes. Douzas and Bacao [25] proposed a geometric SMOTE which generates synthetic samples in a geometric region of the input space. The region is a hyper-sphere around each selected minority instance. Maldonado et al. [26] studied the SMOTE oversampling strategy for high-dimensional datasets, and proposed an alternative distance metric for the computation of the neighbors for each minority sample. Susan and Kumar [27] combined undersampling and oversampling, and proposed a three-step intelligent pruning strategy of majority and minority samples for learning from imbalanced datasets. Mathew et al. [28] proposed a weighted kernel-based SMOTE (WKSMOTE) approach, which generates synthetic positive class samples in feature space. WKSMOTE can overcome the limitation of the linear interpolation of SMOTE. Based on WKSMOTE, Raghuwanshi and Shukla

[29] proposed a SMOTE based class-specific extreme learning machine, which exploits the benefits of both the minority oversampling and the class-specific regularization. Pan et al. [30] proposed two oversampling methods. One is an adaptive SMOTE, which is an improved SMOTE by adaptively selecting groups of inner and danger data from the minority class. The other one adopts Gaussian oversampling, which provides a novel division strategy for sampling regions and makes sampling more reasonable. Zhang and Li [31] proposed an approach to balance different class samples by creating synthetic samples through randomly walking from the real data. Han et al. [32] presented a Gaussian mixture model based combined resampling approach. The resampling approach first determines the number of samples of the majority class and the minority class using a sampling factor. Then to balance the dataset, the Gaussian mixture clustering is used for undersampling of the majority of samples, and the synthetic minority oversampling technique is used for the rest of the samples. Zhang et al. [33] investigated a classification method of high-dimensional class imbalanced datasets and proposed an algorithm to improve the performance of SMOTE by adopting an adaptive over-sampling rate. Elreedy and Atiya [21] presented a theoretical and experimental analysis of the SMOTE method. Specifically, they explored the accuracy of how faithfully it emulates the underlying density, and analyzed the effect of different factors on generation accuracy, such as the dimension, the size of the training set and the considered number of neighbors K. Fernández et al. [22] presented a comprehensive survey on SMOTE-based approaches, in which the progress and challenges of SMOTE-based approaches over fifteen years (from 2003 to 2018) are well summarized.

In recent years, generative adversarial network (GAN) has become a popular research topic in deep learning. Some researchers have used the generation mechanism of GAN to generate synthetic positive class samples for balancing imbalanced datasets. For instance, inspired by the idea of auxiliary classifier GAN (AC-GAN) [34], Ali-Gombe and Elyan proposed an improved model multiple fake class GAN (MFC-GAN) [35] and used the MFC-GAN to handle imbalanced data classification problem. MFC-GAN differs from AC-GAN that it uses multiple fake classes rather than single fake class as in AC-GAN. Furthermore, MFC-GAN can preserve the structure of the minority classes by learning the correct data distribution, which is an intriguing property. Douzas and Bacao [36] applied conditional GAN (cGAN) on binary class imbalanced datasets, where the conditional GAN conditions on the class labels of the imbalanced datasets. Finally generative model is used to create artificial data for the minority class. Zheng et al. [37] introduced a gradient penalty into conditional Wasserstein GAN [38], and proposed a synthetic oversampling approach for imbalanced datasets. Different from these existing methods, the novelty of our proposed method lies in the following three aspects: (a) introducing intra-class divergence as a regularization term to the loss function of GAN to guarantee the diversity of the synthetic samples; (b) introducing MMD-score and Silhouette-score to measure diversity and separability, while the diversity and separability have great influence on the performance of imbalanced data classification; (c) replacing the discriminator of GAN with a classifier whose output is a vector with three entries: the probabilities that a predicted sample belongs to majority class, minority class, and generated sample.

Ensemble method usually combines the data-level and the algorithm-level approach to handle the class imbalance problem. Based on SMOTE combined with Adaboost SVM ensemble integrated with time weighting (ADASVM-TW), Sun et al. [39] proposed two class imbalanced dynamic financial distress prediction approaches. One is the simple integration model of SMOTE with ADASVM-TW, and the other is the embedding integration model of SMOTE with ADASVM-TW. González et al. [40] explored the effectiveness of the switching technique for classification of highly imbalanced problems, and proposed a switching-based ensemble to select the switched examples based on the nearest enemy distance. Gutiérrez-López et al. [41] also investigated the impact of switching technique on class imbalance learning, and proposed an asymmetric binary label switching algorithm to resist binary imbalance and presented a theoretical analysis, concluding that asymmetric switching binary classifiers offer an intrinsic resistance to imbalance effects. Raghuwanshi and Shukla [42] proposed an ensemble approach using a reduced kernelized weighted extreme learning machine as the base classifier to solve the class imbalance problem effectively. Hsiao et al. [43] proposed a method named MTSbag for class imbalance problems. MTSbag integrates the Mahalanobis–Taguchi system (MTS) and the bagging-based ensemble learning approaches to enhance the ability of conventional MTS in handling imbalanced data. Zhai et al. [44] combined oversampling method and ensemble learning, and proposed a MapReduce based imbalanced large scale data classification. The proposed oversampling method is based on enemy nearest neighbor. In this paper, we present a classifier fusion approach based on fuzzy integral for imbalanced data classification. Since the fusion method can well model the interactions among the base classifiers due to using fuzzy integral as an ensemble tool, the proposed approach can effectively enhance the generalization performance of the classification algorithm.

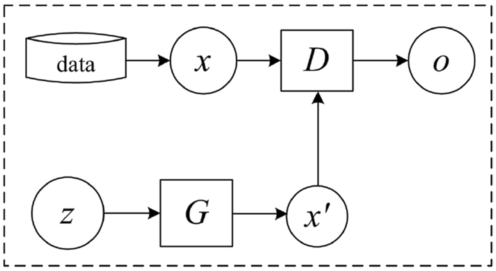

Fig. 1 The architecture of generative adversarial network

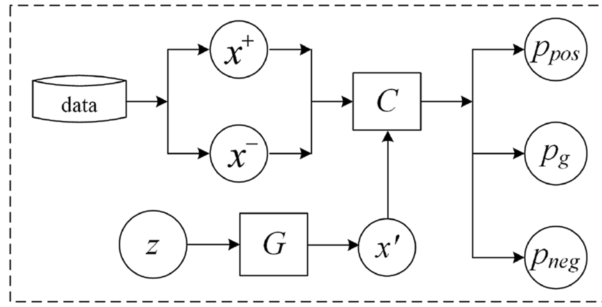

Fig. 2 The architecture of improved generative adversarial network

3 The proposed framework

In this section, we present the proposed framework for addressing the two-class imbalance problem. The framework includes an oversampling method which is based on an improved GAN model, and a two-class imbalanced data classification approach which is based on classifier fusion via fuzzy integral.

3.1 Oversampling method based on an improved GAN model

GAN is a generative model which consists of two neural networks G and D (see Fig. 1). The G is a generator network whose input, denoted by $\boldsymbol{z}$, is drawn from a known noise prior distribution p_{noise}, and its output is denoted by $\boldsymbol{x}'$. The D is a discriminator network, whose input includes the generated data $\boldsymbol{x}'$ and real data $\boldsymbol{x}$. The distribution of $\boldsymbol{x}$ is denoted by p_d which is unknown. The output of discriminator D is a probability distribution which indicates the support degrees that the input comes from p_{data} or from p_{gen}.

Since GAN is a probabilistic generative model, it is a natural idea to use GAN to generate synthetic positive class samples for addressing the two-class imbalanced data classification problem. However, we found that if we only learn the distribution of positive class samples using GAN, it is easy to incur overlap between the positive and the negative class samples. In addition, since GAN is prone to mode collapse, the generated synthetic positive class samples by GAN lack diversity. In this section, we present the proposed oversampling method to deal with these two problems, which is based on an improved GAN model.

In the proposed method, we improve the GAN model on two aspects: (1) We replace the discriminator of GAN with a classifier C (see Fig. 2), and its output would be p_{pos} for the positive class samples, p_{neg} for the negative class samples, and p_g for the generative samples by generator G. In the adversarial training process for generator G and classifier C, we want the samples generated by generator G to fool the classifier C, namely when a generated sample is fed as input to the classifier, we want the output to be close to p_{pos}. Classifier C can not only learn the distribution of samples, but also learn a good classification boundary between the positive and the negative class. (2) We introduce a regularization term of intra-class divergence into the loss function of the GAN, which can enhance the diversity of the generated samples by generator G and avoid mode collapse of GAN.

Let $S = S^+ \cup S^-$, S^+ and S^- denote the positive class and negative class respectively, and let S^+_{up} be the oversampled positive class, $\boldsymbol{m}$ and $\boldsymbol{m}'$ are the mean vectors of the positive class samples and the oversampled positive class samples respectively. The loss function of the improved GAN is given by Eq. (1)

$$
\begin{aligned}
L(G(z)) \\
&= \frac{1}{|S^+|}\sum_{x\in S^+}(\boldsymbol{x}-\boldsymbol{m})(\boldsymbol{x}-\boldsymbol{m})^{\mathrm{T}} \\
&\quad + \frac{1}{|S^+_{up}|}\sum_{G(z)\in S^+_{up}}(G(z)-\boldsymbol{m}')(G(z)-\boldsymbol{m}')^{\mathrm{T}}
\end{aligned}
\tag{1}
$$

The objective functions of C and G of the improved GAN model are given by Eqs. (2) and (3) respectively.

$$\max_C J = J_1 + J_2 + J_3 \tag{2}$$

$$\max_G L = J_4 + \lambda L(G(z)) \tag{3}$$

where λ is a parameter, and

$$
\begin{aligned}
J_1 &= E_{\boldsymbol{x}\sim p_{neg}} \log C_1(\boldsymbol{x}) + E_{\boldsymbol{x}\sim p_{neg}} \log(1 - C_2(\boldsymbol{x})) \\
&\quad + E_{\boldsymbol{x}\sim p_{neg}} \log(1 - C_3(\boldsymbol{x}))
\end{aligned}
\tag{4a}
$$

$$J_2 = E_{\boldsymbol{x}\sim p_{pos}} \log C_2(\boldsymbol{x}) + E_{\boldsymbol{x}\sim p_{pos}} \log(1 - C_1(\boldsymbol{x})) + E_{\boldsymbol{x}\sim p_{pos}} \log(1 - C_3(\boldsymbol{x})) \tag{4b}$$

$$J_3 = E_{\boldsymbol{x}\sim p_g} \log C_3(\boldsymbol{x}) + E_{\boldsymbol{x}\sim p_g} \log(1 - C_1(\boldsymbol{x})) + E_{\boldsymbol{x}\sim p_g} \log(1 - C_2(\boldsymbol{x})) \tag{4c}$$

$$J_4 = E_{\boldsymbol{x}\sim p_g} \log C_2(\boldsymbol{x}) - E_{\boldsymbol{x}\sim p_g} \log C_1(\boldsymbol{x}) - E_{\boldsymbol{x}\sim p_g} \log C_3(\boldsymbol{x}) \tag{4d}$$

In the adversarial learning process, G attempts to generate diverse positive class samples and expect that C can categorize the generated samples to minority class, while C attempts to classify correctly the positive, negative and generated samples. It is can be proved that the optimal C will result in the following formula (5).

$$L = -KL\left(p_g \parallel p_{pos}\right) + H\left(p_g, p_{neg}\right) \tag{5}$$

where $KL\left(p_g \parallel p_{pos}\right)$ is the KL divergence between p_g and p_{pos}, and $H\left(p_g, p_{neg}\right)$ is the cross entropy between p_g and p_{neg}. In the following, we prove that Eq. (5) is hold. Because the item of intr-class divergence is not related to the classifier C, hence for $C_i(\boldsymbol{x})$, $1 \le i \le 3$, we can obtain the following equation.

$$\begin{aligned} J(C_1(\boldsymbol{x})) &= E_{\boldsymbol{x}\sim p_{neg}} \log C_1(\boldsymbol{x}) + E_{\boldsymbol{x}\sim p_{pos}} \log(1 - C_1(\boldsymbol{x})) + E_{\boldsymbol{x}\sim p_g} \log(1 - C_1(\boldsymbol{x})) \\ &= \int \big(p_{neg} \log C_1(\boldsymbol{x}) + p_{pos} \log(1 - C_1(\boldsymbol{x})) + p_g \log(1 - C_1(\boldsymbol{x}))\big) d\boldsymbol{x} \end{aligned}$$

Take the partial derivative of the integrand, and set it equal to zero, we have the following equation.

$$\frac{p_{neg}}{C_1(\boldsymbol{x})} - \frac{p_{pos}}{1 - C_1(\boldsymbol{x})} - \frac{p_g}{1 - C_1(\boldsymbol{x})} = 0$$

Hence,

$$C_1^*(\boldsymbol{x}) = \frac{p_{neg}}{p_{neg} + p_{pos} + p_g}$$

Similarly, we have,

$$C_2^*(\boldsymbol{x}) = \frac{p_{pos}}{p_{neg} + p_{pos} + p_g}$$
$$C_3^*(\boldsymbol{x}) = \frac{p_g}{p_{neg} + p_{pos} + p_g}$$

Substitute $C_1^*(\boldsymbol{x})$, $C_2^*(\boldsymbol{x})$ and $C_3^*(\boldsymbol{x})$ into L [i.e. (4d)], we have,

$$\begin{aligned} L &= E_{\boldsymbol{x}\sim p_g} \log C_2^*(\boldsymbol{x}) - E_{\boldsymbol{x}\sim p_g} \log C_1^*(\boldsymbol{x}) - E_{\boldsymbol{x}\sim p_g} \log C_3^*(\boldsymbol{x}) \\ &= \int \left(p_g \log p_{pos}\right) d\boldsymbol{x} - \int \left(p_g \log p_{neg}\right) d\boldsymbol{x} - \int \left(p_g \log p_g\right) d\boldsymbol{x} \\ &= \int \left(p_g \log \frac{p_{pos}}{p_g}\right) d\boldsymbol{x} - \int \left(p_g \log p_{neg}\right) d\boldsymbol{x} \\ &= -KL\left(p_g \parallel p_{pos}\right) + H\left(p_g, p_{neg}\right) \end{aligned}$$

Note: (1) For $KL\left(p_g \parallel p_{pos}\right)$, since p_{pos} is fixed, we want p_g to be as close to p_{pos} as possible. It is noted that $KL(\cdot \parallel \cdot)$ is not symmetric, for different optimization objective, the results are different (see Fig. 3). Obviously, we should adopt the optimization objective given in Fig. 3b. (2) The cross entropy $H\left(p_g, p_{neg}\right)$ is used to distinguish the generated samples from the negative class samples as much as possible. (3) For some cases, the number of positive class samples are too small to train a model, accordingly we train the model with an incremental iterative mode. The pseudo code of the proposed oversampling algorithm is given in Algorithm 1.

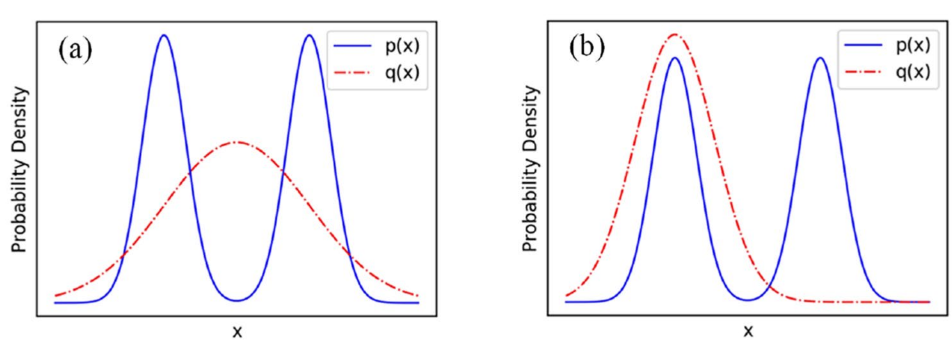

Fig. 3 **a** Optimization for $argmin_q\, KL(p \parallel q)$, **b** optimization for $argmin_q\, KL(q \parallel p)$

Algorithm 1: Oversampling algorithm based on an improved GAN model

Input: Imbalanced dataset $S = S^+ \cup S^-$, the size of batch m, the iterative number n, and the number of training t.
Output: The parameter $\boldsymbol{\theta}_g$ of generator G.

Let $S^+_{up} = S^+$;
Initialize the parameter $\boldsymbol{\theta}_g$ of generator G and the parameter $\boldsymbol{\theta}_c$ of classifier C with small random numbers.
for $(i = 1; i \le n; i = i + 1)$ **do**
  for $(j = 1; j \le t; j = j + 1)$ **do**
    Sample m samples from noise prior distribution p_{noise}, and input them to G, obtain m generated samples $\{\boldsymbol{x}'_1, \boldsymbol{x}'_2, \cdots, \boldsymbol{x}'_m\}$;
    Sample m samples $\{\boldsymbol{x}^+_1, \boldsymbol{x}^+_2, \cdots, \boldsymbol{x}^+_m\}$ from S^+;
    Sample m samples $\{\boldsymbol{x}^-_1, \boldsymbol{x}^-_2, \cdots, \boldsymbol{x}^-_m\}$ from S^-;
    Fix the parameter $\boldsymbol{\theta}_g$ of generator G, update the parameter $\boldsymbol{\theta}_c$ of classifier C by ascending its stochastic gradient;
    Sample m samples $\{\boldsymbol{z}^-_1, \boldsymbol{z}^-_2, \cdots, \boldsymbol{z}^-_m\}$ from noise prior distribution p_{noise};
    Fix the parameter $\boldsymbol{\theta}_c$ of classifier C, update the parameter $\boldsymbol{\theta}_g$ of generator G by ascending its stochastic gradient;
    Sample m samples $\{\boldsymbol{z}^-_1, \boldsymbol{z}^-_2, \cdots, \boldsymbol{z}^-_m\}$ from noise prior distribution p_{noise};
    Input $\{\boldsymbol{z}^-_1, \boldsymbol{z}^-_2, \cdots, \boldsymbol{z}^-_m\}$ into generator G, obtain $S_g = \{\boldsymbol{x}'_1, \boldsymbol{x}'_2, \cdots, \boldsymbol{x}'_m\}$
    Let $S^+_{up} = S^+_{up} \cup S_g$;
  end
end
Return S^+_{up};

3.2 Two-class imbalanced data classification approach based on classifier fusion via fuzzy integral

On the basis of the above oversampling method, we proposed a two-class imbalanced data classification approach based on classifier fusion via fuzzy integral [45]. The proposed approach includes the following two stages:

(1) Construct balance training sets and train base classifiers

In this stage, we first partition S^- into l subsets $S^-_1, S^-_2, \dots, S^-_l$, where $l = \frac{|S^-|}{|S^+_{up}|}$. Next, construct l balance training sets $S_i = S^-_i \cup S^+_{up}$, $1 \le i \le l$. Finally, train l classifiers $C = \{C_1, C_2, \dots, C_l\}$ on the l balance training sets. The l classifiers are fused for imbalanced data classification via fuzzy integral in the next stage.

(2) Fuse the trained base classifiers via fuzzy integral

As a classifier fusion method, fuzzy integral is distinguished from other fusion methods due to its intriguing property, that is it can well model the interactions among the base classifiers, including positive interaction and negative interaction, this is the reason why we select fuzzy integral to fuse the trained base classifiers.

Let $D = \{(\boldsymbol{x}_i, y_i) | x_i \in R^d, y_i \in Y\}$ be a training set, $1 \le i \le n$, $Y = \{\omega_1, \omega_2, \dots, \omega_k\}$ be a set of class labels, $C = \{C_1, C_2, \dots, C_l\}$ be a set of classifiers trained on D or on subsets of D. For $\forall \boldsymbol{x} \in R^d$, the output of classifier C_i is a k-dimensional vector $(p_{i1}(\boldsymbol{x}), p_{i2}(\boldsymbol{x}), \dots, p_{ik}(\boldsymbol{x}))$. The $p_{ij}(\boldsymbol{x}) \in [0, 1] (1 \le i \le l; 1 \le j \le k)$ denotes the support degree given by classifier C_i to the hypothesis that $\boldsymbol{x}$ comes from class ω_j, $\sum_{j=1}^{k} p_{ij}(\boldsymbol{x}) = 1$.

Given $C = \{C_1, C_2, \dots, C_l\}$, $Y = \{\omega_1, \omega_2, \dots, \omega_k\}$, and arbitrary test sample $\boldsymbol{x}$. The following matrix is called decision matrix with respect to $\boldsymbol{x}$.

$$DM(\boldsymbol{x}) = \begin{bmatrix} p_{11}(\boldsymbol{x}) & \cdots & p_{1j}(\boldsymbol{x}) & \cdots & p_{1k}(\boldsymbol{x}) \\ \vdots & & \vdots & & \vdots \\ p_{i1}(\boldsymbol{x}) & \cdots & p_{ij}(\boldsymbol{x}) & \cdots & p_{ik}(\boldsymbol{x}) \\ \vdots & & \vdots & & \vdots \\ p_{l1}(\boldsymbol{x}) & \cdots & p_{lj}(\boldsymbol{x}) & \cdots & p_{lk}(\boldsymbol{x}) \end{bmatrix} \tag{6}$$

In the matrix $DM(\boldsymbol{x})$, the *ith* row of the matrix is the output of classifier C_i, the *jth* column of the matrix are the support degrees from classifiers $C_1, C_2, \dots, C_l$ for class ω_j.

Let $P(C)$ be the power set of C, the fuzzy measure on C is a set function: $g : P(C) \to [0, 1]$, which satisfies the following two conditions:

1. $g(\emptyset) = 1$, $g(C) = 1$;
2. For $\forall C_i, C_j \subseteq C$, if $C_i \subset C_j$, then $g(C_i) \le g(C_j)$.

For $\forall C_i, C_j \subseteq C$ and $C_i \cap C_j = \emptyset$, g is called λ-fuzzy measure, if it satisfies the following condition:

$$g(C_i \cup C_j) = g(C_i) + g(C_j) + \lambda g(C_i) g(C_j) \tag{7}$$

where $\lambda > -1$ and $\lambda \neq 0$.

The value of λ can be determined by solving the following Eq. (8).

$$\lambda + 1 = \prod_{i=1}^{l} (1 + \lambda g_i) \tag{8}$$

where $g_i = g(\{C_i\})$, it is usually determined by the following formula (9) [46]:

$$g_i = \frac{p_i}{\sum_{j=1}^{l} p_j} \delta. \tag{9}$$

where $\delta \in [0, 1]$ and p_i is testing accuracy or verification accuracy of classifier $C_i (1 \le i \le l)$.

Let $h : C \to [0, 1]$ be a function defined on C. The Choquet fuzzy integral of function h with respect to g is defined by the following Eq. (10).

$$(C) \int h d\mu = \sum_{i=2}^{l+1} \left(h(C_{i-1}) - h(C_i) \right) g(F_{i-1}) \tag{10}$$

where $h(C_1) \ge h(C_2) \ge \cdots \ge h(C_l)$, $h(C_{l+1}) = 0$, $F_{i-1} = \{C_1, C_2, \ldots, C_{i-1}\}$.

Given a test instance $\boldsymbol{x}$, when we use fuzzy integral to fuse l base classifiers $C_1, C_2, \ldots, C_l$ for classifying $\boldsymbol{x}$, the process includes three step: Firstly, compute decision matrix $DM(\boldsymbol{x})$. Secondly, sort $j^{th} (1 \le j \le k)$ column of $DM(\boldsymbol{x})$ in descending order and obtain $(p_{i_1 j}, p_{i_2 j}, \ldots, p_{i_l j})$. Finally, calculate the support degree $p_j(\boldsymbol{x})$ by the following formula (11).

$$p_j(\boldsymbol{x}) = \sum_{t=2}^{l+1} \left(p_{i_{t-1} j}(\boldsymbol{x}) - p_{i_t j}(\boldsymbol{x}) \right) g(F_{t-1}) \tag{11}$$

The pseudo code of the proposed two-class imbalanced data classification algorithm based on classifier fusion via fuzzy integral is given in Algorithm 2.

Table 1 The basic information of the 11 datasets

Datasets	♯Attribute	♯Instances	IR	Note
Gaussian	2	10,000	100	1 artificial dataset
Blocks0	10	5472	8.79	4 KEEL datasets
Segment0	19	2308	6.02	
Yeast1	8	1484	2.46	
Vowel0	13	988	9.98	
Liver1	5	12,400	61	3 liver datasets
Liver2	5	14,000	13	
Liver3	5	13,000	25	
MNIST	784	54,100	540	3 image datasets
Fashion-MNIST	784	54,100	540	
Cifar10	3072	44,100	440	

Table 2 The mean vectors and covariance matrices of two Gaussian distributions

i	μ_i	Σ_i
1	$(1.0, 1.0)^T$	$\begin{bmatrix} 0.6 & -0.2 \\ -0.2 & 0.6 \end{bmatrix}$
2	$(2.5, 2.5)^T$	$\begin{bmatrix} 0.2 & -0.1 \\ -0.1 & 0.2 \end{bmatrix}$

Algorithm 2: The two-class imbalanced data classification algorithm based on classifier fusion via fuzzy integral

Input: Imbalanced dataset $S = S^+ \cup S^-$, test sample $\boldsymbol{x}$.
Output: j^*, the class label of $\boldsymbol{x}$.

Call algorithm 1, and obtain S^+_{up};
`// The first stage: Construct balance training sets and train base classifiers;`
Partition S^- into l subsets $S^-_1, S^-_2, \cdots, S^-_l$, where $l = \frac{|S^-|}{|S^+_{up}|}$;
for $(i = 1; i \leq l; i = i + 1)$ **do**
  Construct balance training sets $S_i = S^-_i \cup S^+_{up}$;
  Train base classifier C_i on S_i, and soft-maximize its outputs, obtain a probability distribution $(p_{i1}(\boldsymbol{x}), p_{i2}(\boldsymbol{x}), \cdots, p_{ik}(\boldsymbol{x}))$;
end
`// The second stage: fuse the trained base classifiers by fuzzy integral;`
Calculate fuzzy densities $g_i (1 \leq i \leq l)$ by (9);
Calculate parameter λ by (8);
Calculate $DM(\boldsymbol{x})$ by (6);
for $(j = 1; j \leq k; j = j + 1)$ **do**
  Sort j^{th} column of $DM(\boldsymbol{x})$ in descending order and obtain $(d_{i_1 j}, d_{i_2 j}, \cdots, d_{i_l j})$;
  Set $g(F_1) = g_{i_1}$;
  for $(t = 2; t \leq l; t = t + 1)$ **do**
    Calculate $g(F_t) = g_{i_t} + g(F_{t\text{-}1}) + \lambda g_{i_t} g(F_{t\text{-}1})$;
  end
  Calculate $p_j(x) = \sum_{t=2}^{l+1} [d_{i_{t-1} j}(x) \text{-} d_{i_t j}(x)] g(F_{t\text{-}1})$;
end
Calculate $p_{j^*}(\boldsymbol{x}) = argmax_{1 \leq j \leq k} \{p_j(\boldsymbol{x})\}$;
Return j^*.

4 Experimental results and analyses

4.1 datasets and experimental environments

To demonstrate the superiority of the proposed framework denoted by GANDO (generative adversarial network based diverse oversampling), we conducted extensive experiments on 11 datasets including 8 numeric datasets and 3 image datasets. We use the 8 numeric datasets to compare GANDO with 11 SMOTE related state-of-the-art approaches which are SMOTE [23], B-SMOTE [47], ADASYN [48], CCR [49], ANS [50], K-SMOTE [24], NRPSOS [51], OUPS [52], GAN [8], AC-GAN [34], MFC-GAN [35], and use 3 image datasets to compare GANDO with 4 GAN related state-of-the-art methods which are AUGMENT [53], GAN [8], AC-GAN [34], and MFC-GAN [35]. The 8 numeric datasets include 1 artificial dataset, 4 KEEL datasets [54], and 3 liver datasets [55]. The basic information of the 11 datasets is given in Table 1. All experiments were carried out on the same hardware platform with Intel(R) Core(TM) i7-6600k CPU @ 3.10 GHz, 16.0 G memory, 64 bit MAC operation system. The programming environment consists of PyCharm Community Edition 2017.1.1, scikit-learn, smote-variants and keras. Our code is publicly available at https://github.com/xichie/oversample.

Table 3 Model parameter settings used for 8 numeric datasets

Datasets	♯HNodesG	♯HNodesC	n	k	λ	♯Oversampling
Gaussian	150	100	3	500	0.01	1970
Blocks0	250	100	5	500	0.01	460
Segment0	250	100	3	2000	0.01	330
Yeast1	250	150	5	500	0.01	100
Vowel0	250	100	5	500	0.01	80
Liver1	250	150	5	2000	0.10	2240
Liver2	250	150	5	2000	0.10	1600
Liver3	250	150	5	2000	0.10	2000

Table 4 The network structures of generator and classifier used for 3 image datasets

Datasets	Structure of G	Structure of C
MNIST Fashion-MNIST	Dense,256,LeakyReLU	
	BatchNormalization	
	Dense,512,LeakyReLU	Flatten
	BatchNormalization	Dense,512,LeakyReLU
	Dense,1024,LeakyReLU	Dense,256,LeakyReLU
	BatchNormalization	Dense,3,Softmax
	Dense,784,Tanh	
	Reshape,(28,28,1)	
Cifar10	Dense,32768,LeakyReLU	
	Reshape,(16,16,128)	3×3 Conv,128,LeakyReLU
	5×5 Conv,256,LeakyReLU	3×3 Conv,128,LeakyReLU
	4×4 DeConv,256,LeakyReLU	3×3 Conv,128,LeakyReLU
	5×5 Conv,256,LeakyReLU	3×3 Conv,128,LeakyReLU
	5×5 Conv,256,LeakyReLU	Dropout,0.5
	7×7 Conv,3,Tanh	Dense,3,Softmax

Table 5 Model parameter settings used for 3 image datasets

Datasets	n	k	λ	♯Oversampling
MNIST	5	10,000	0.1	2000
Fashion-MNIST	7	20,000	0.03	2000
Cifar10	3	30,000	0.01	3500

In Table 1, IR $= \frac{|S^-|}{|S^+|}$. Gaussian is an artificial dataset which is a two-dimensional dataset with two classes followed two Gaussian distributions, the mean vectors and covariance matrices of the two Gaussian distributions are given in Table 2. The artificial dataset Gaussian is used for illustrating the feasibility of the proposed approach and visualizing the generated synthetic samples.

The three well known image datasets are not imbalanced, so we transform them into imbalanced ones.The purpose of selecting the three datasets is used to demonstrate the feasibility and effectiveness of the proposed method for image data.

MNIST is a handwritten digital dataset which includes 70,000 28×28 grayscale images, the training and test set contain 60,000 and 10,000 images respectively. We randomly select 100 images from zero class as positive class sample, and put other classes images together as negative class.

Fashion-MNIST dataset is similar to MNIST, it also includes 70,000 28×28 grayscale images of 70,000 fashion products from 10 categories. We randomly select 100 images from T-Shirt class as positive class sample, and put other classes images together as negative class.

Cifar10 consists of 60,000 32×32 colour images containing one of 10 object classes, with 6000 images per class.

Table 6 The architectures of the two different neural networks

Datasets	Structures
MNIST/Fashion-MNIST	Dense,256,LeakyReLU
	Dense,128,LeakyReLU
	Dense,1,Sigmoid
Cifar10	3×3 Conv,128,LeakyReLU
	3×3 Conv,128,LeakyReLU
	3×3 Conv,128,LeakyReLU
	DropOut,0.5
	Dense,1,Sigmoid

The training and test set contain 50,000 and 10,000 images respectively. We randomly select 100 images from airplane class as positive class sample, and put other classes images together as negative class.

4.2 Performance evaluation measures

The used performance evaluation measures include MMD-score [56], Silhouette-score [57], F-measure [58], G-mean [58], and AUC-area [58]. The MMD is a statistics for measuring the mean squared difference of two sets of samples. Given two sets of samples $\mathbf{X} = \{\mathbf{x}_i\}$, $1 \le i \le n$ and $\mathbf{Y} = \{\mathbf{y}_i\}$, $1 \le i \le m$, the MMD of $\mathbf{X}$ and $\mathbf{Y}$ is defined by Eq. (12).

Table 7 Experimental comparison of MMD-score on the 8 numeric datasets

Approaches	Gaussian	Blocks0	Segment0	Yeast1	Vowel0	Liver1	Liver2	Liver3
SMOTE	0.026	0.006	0.012	0.011	0.042	0.018	0.004	0.008
B-SMOTE	0.394	0.348	1.179	0.040	0.425	0.049	0.056	0.051
ADASYN	0.397	0.322	0.624	0.023	0.206	0.046	0.049	0.038
CCR	0.260	0.070	0.364	0.036	0.115	0.039	0.020	0.025
ANS	0.197	0.078	0.053	0.044	0.037	0.193	0.085	0.105
K-SMOTE	0.024	**2.441**	0.406	0.285	0.038	0.014	0.003	0.060
NRPSOS	0.251	0.071	0.013	0.130	0.045	0.201	0.029	0.064
OUPS	0.126	0.053	0.076	0.069	0.149	0.102	0.027	0.026
GAN	0.580	0.125	0.408	0.067	0.398	0.238	0.203	0.277
AC-GAN	1.181	0.437	1.371	0.653	1.139	0.959	0.391	1.321
MFC-GAN	1.436	1.124	1.495	0.805	1.217	0.253	0.033	1.357
GANDO	**2.138**	1.167	**2.815**	**1.784**	**3.249**	**3.519**	**2.537**	**2.800**

The maximum values are in bold, indicating the best performance

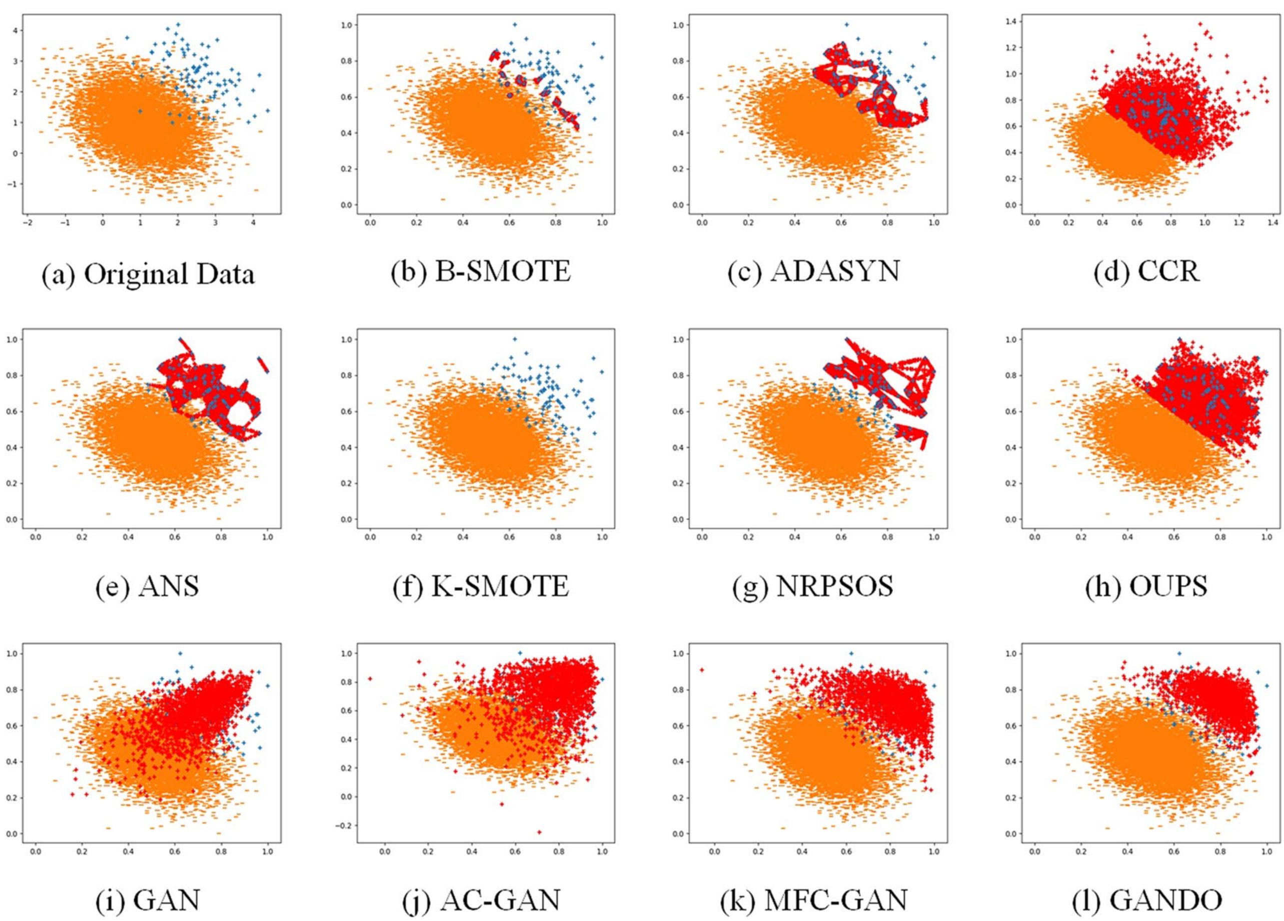

Fig. 4 The visualization of the generated synthetic positive class samples of the artificial dataset

$$\begin{aligned}\mathrm{MMD} &= \left\| \frac{1}{n}\sum_{i=1}^{n}\phi(\mathbf{x}_i) - \frac{1}{m}\sum_{j=1}^{m}\phi(\mathbf{y}_j)\right\|^2 \\ &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{i'=1}^{n}\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_{i'}) \\ &\quad - \frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\phi(\mathbf{x}_i)^T\phi(\mathbf{y}_j) \\ &\quad + \frac{1}{m^2}\sum_{j=1}^{m}\sum_{j'=1}^{m}\phi(\mathbf{y}_j)^T\phi(\mathbf{y}_{j'})\end{aligned} \tag{12}$$

In Eq. (12), $\phi(\cdot)$ is a kernel mapping, using kernel trick, Eq. (12) can be written as Eq. (13).

$$\begin{aligned}\mathrm{MMD} =& \frac{1}{n^2}\sum_{i=1}^{n}\sum_{i'=1}^{n}k(\mathbf{x}_i, \mathbf{x}_{i'}) \\ &- \frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}k(\mathbf{x}_i, \mathbf{y}_j) \\ &+ \frac{1}{m^2}\sum_{j=1}^{m}\sum_{j'=1}^{m}k(\mathbf{y}_j.\mathbf{y}_{j'})\end{aligned} \tag{13}$$

The Silhouette coefficient (Silhouette-score) is an evaluation index of clustering algorithms. Given a sample $\mathbf{x}$ which belongs to cluster A, the Silhouette coefficient of $\mathbf{x}$ is defined by Eq. (14).

$$s(\mathbf{x}) = \frac{b(\mathbf{x}) - a(\mathbf{x})}{\max\{a(\mathbf{x}), b(\mathbf{x})\}} \tag{14}$$

where $a(\mathbf{x})$ is the average dissimilarity of sample $\mathbf{x}$ to all other samples of A, $b(\mathbf{x}) = \mathrm{minimum}_{\mathrm{C}\neq\mathrm{A}} d(\mathbf{x}, \mathrm{C})$, while $d(\mathbf{x}, \mathrm{C})$ is the average dissimilarity of sample $\mathbf{x}$ to all samples

Table 8 Experimental comparison of Silhouette-score on the 8 numeric datasets

Approaches	Gaussian	Blocks0	Segment0	Yeast1	Vowel0	Liver1	Liver2	Liver3
SMOTE	0.449	0.171	0.186	0.051	0.094	0.081	0.069	0.074
B-SMOTE	0.394	0.098	0.229	0.042	0.254	0.049	0.049	0.053
ADASYN	0.380	0.091	0.214	0.042	0.106	0.057	0.042	0.049
CCR	0.352	0.153	0.103	0.036	0.072	0.055	0.052	0.052
ANS	0.537	0.182	0.158	0.071	0.098	0.151	0.120	0.121
K-SMOTE	0.438	**0.520**	0.218	0.133	0.038	-0.13	-0.09	-0.12
NRPSOS	0.548	0.237	0.185	0.114	0.094	0.152	0.091	0.106
OUPS	0.442	0.190	0.213	0.095	0.110	0.064	0.085	0.082
GAN	0.441	0.216	0.080	0.066	0.124	-0.15	-0.080	-0.11
AC-GAN	0.382	0.231	0.439	0.391	0.556	0.280	0.233	0.207
MFC-GAN	0.425	0.407	0.244	0.360	0.471	0.670	0.151	0.187
GANDO	**0.624**	0.412	**0.582**	**0.435**	**0.561**	**0.768**	**0.277**	**0.287**

The maximum values are in bold, indicating the best performance

Table 9 Experimental comparisons of F-measure on the 8 numeric datasets

Approaches	Gaussian	Blocks0	Segment0	Yeast1	Vowel0	Liver1	Liver2	Liver3
SMOTE	0.34	0.65	0.88	0.46	0.93	0.85	0.86	0.86
B-SMOTE	0.33	0.56	0.63	0.45	0.92	0.05	0.87	0.86
ADASYN	0.26	0.56	0.66	0.40	0.93	0.05	0.89	0.78
CCR	0.23	0.64	0.83	0.50	0.84	0.05	0.28	0.16
ANS	0.58	0.62	0.87	0.49	0.93	0.05	0.92	0.81
K-SMOTE	0.14	0.64	0.90	0.50	0.93	0.05	0.90	0.91
NRPSOS	0.56	0.61	0.88	0.46	0.90	0.05	0.91	0.88
OUPS	0.55	0.23	0.89	0.48	0.86	0.05	0.82	0.87
GAN	0.61	0.63	0.89	0.49	0.93	0.59	**0.92**	0.81
AC-GAN	0.62	0.54	0.86	0.39	0.88	0.63	0.91	0.76
MFC-GAN	0.68	0.60	0.96	0.40	0.90	0.71	0.87	0.82
GANDO	**0.73**	**0.70**	**0.91**	**0.60**	**0.95**	**0.74**	**0.92**	**0.90**

The maximum values are in bold, indicating the best performance

Table 10 Experimental comparison of G-mean on the 8 numeric datasets

Approaches	Gaussian	Blocks0	Segment0	Yeast1	Vowel0	Liver1	Liver2	Liver3
SMOTE	0.45	0.72	0.90	0.58	0.95	0.85	0.95	0.93
B-SMOTE	0.46	0.63	0.72	0.59	0.93	0.67	0.95	0.95
ADASYN	0.34	0.63	0.70	0.59	0.94	0.16	0.95	0.89
CCR	0.31	0.71	0.85	0.61	0.86	0.16	0.40	0.29
ANS	0.59	0.69	0.92	0.60	0.96	0.16	0.97	**0.98**
K-SMOTE	0.54	0.73	0.91	0.63	0.96	0.00	0.98	**0.98**
NRPSOS	0.79	0.70	0.89	**0.64**	0.92	0.16	**0.99**	**0.98**
OUPS	0.76	0.36	0.90	0.59	0.87	0.16	0.89	0.88
GAN	0.69	0.70	0.91	0.58	0.93	0.70	0.91	0.91
AC-GAN	0.63	0.63	0.92	0.62	0.96	0.84	0.98	0.90
MFC-GAN	0.70	0.69	**0.93**	0.62	**0.97**	0.75	0.98	0.97
GANDO	**0.87**	**0.85**	**0.93**	**0.64**	0.96	**0.86**	**0.99**	0.97

The maximum values are in bold, indicating the best performance

Table 11 Experimental comparison of AUC-area on the 8 numeric datasets

Approaches	Gaussian	Blocks0	Segment0	Yeast1	Vowel0	Liver1	Liver2	Liver3
SMOTE	0.60	0.76	0.90	0.62	0.95	0.90	0.91	0.92
B-SMOTE	0.61	0.70	0.76	0.59	**0.98**	0.67	0.92	0.91
ADASYN	0.56	0.70	0.75	0.62	**0.98**	0.67	**0.93**	0.89
CCR	0.55	0.75	0.86	0.65	**0.98**	0.66	0.76	0.76
ANS	0.67	0.73	0.92	0.64	0.97	0.62	0.91	0.91
K-SMOTE	0.27	0.85	0.91	0.65	0.97	0.50	0.92	0.90
NRPSOS	0.70	**0.86**	**0.97**	0.63	0.97	0.61	0.92	0.89
OUPS	0.77	0.59	**0.97**	0.63	0.96	0.66	0.92	0.92
GAN	0.74	0.74	0.92	0.62	0.92	0.58	0.90	0.91
AC-GAN	0.67	0.82	0.95	0.59	0.94	0.77	**0.93**	0.84
MFC-GAN	0.70	0.84	0.96	0.60	0.93	0.76	0.89	0.86
GANDO	**0.88**	**0.86**	0.93	**0.67**	**0.98**	**0.88**	**0.93**	**0.92**

The maximum values are in bold, indicating the best performance

of cluster C. With respect to a cluster (or a set) A, the Silhouette coefficient of A is $s(\mathrm{A}) = \frac{1}{|A|}\sum_{\mathbf{x}\in \mathrm{A}} s(\mathbf{x})$. From Eq. (14), it is easy to find that the value of $s(\mathbf{x})$ is between $[-1, 1]$, and the closer the value of $s(\mathbf{x})$ to 1, the better the separability is.

4.3 Network architecture and parameter settings

For two different kind of datasets, we employ different network architecture and parameter settings. Specifically, for the 8 numeric datasets, the generator and the classifier are all single hidden layer feedforward neural networks, the dimension of noise z is uniformly set to 100. Other parameters including the number of hidden nodes of generator (denoted by ♯HNodesG), the number of hidden nodes of classifier (denoted by ♯HNodesC), the number of iteration (n), the number of training (k), the weighted parameter λ, and the number of oversampling samples (denoted by ♯Oversampling) at each time are given in Table 3. In the second stage, we use support vector machine as the base classifier for fusion via fuzzy integral to classify two-class imbalanced data.

For the 3 image datasets, because MNIST and Fashion-MNIST are single channel grayscale image datasets, the generator and classifier are all use same fully connected networks. Since Cifar10 is three channel color image dataset, the generator and classifier are all use convolutional neural networks, the ADAM is used as the optimization method, the mini-batch size is 100. The network structures of G and C are given in Table 4, other model parameter settings are given in Table 5. In the second stage, we use two different neural networks as the base classifiers for fusion via fuzzy integral to classify two-class imbalanced data. For MNIST and Fashion-MNIST, we use same neural network as base classifier, whereas a different neural network is employed for Cifar10, the architectures of the two different neural

Table 12 Experimental comparison of MMD-score on the 3 image datasets

Approaches	MNIST	Fashion-MNIST	Cifar10
AUGMENT	0.864	0.670	0.762
GAN	0.921	1.159	0.894
AC-GAN	**2.512**	1.325	**3.534**
MFC-GAN	1.057	1.432	1.364
GANDO	1.260	**1.727**	2.121

The maximum values are in bold, indicating the best performance

Table 13 Experimental comparison of Silhouette-score on the 3 image datasets

Approaches	MNIST	Fashion-MNIST	Cifar10
AUGMENT	0.338	0.382	0.017
GAN	0.236	0.278	0.421
AC-GAN	0.182	0.205	-0.223
MFC-GAN	0.516	**0.453**	0.484
GANDO	**0.520**	0.410	**0.678**

The maximum values are in bold, indicating the best performance

networks are given in Table 6. Regarding the parameter choice, we use grid search strategy to select parameters and pick the ones which resulted in the best performance. For example, regarding the numbers of hidden node of encoder and decoder networks used for 8 numeric datasets given in Table 3 and the numbers of hidden node of generator and discriminator networks used for 3 image datasets given in Table 5. For each dataset, we determine the suitable numbers of hidden node of neural networks by grid search strategy in same interval [50, 150].

4.4 Comparisons with 11 SMOTE related state-of-the-art approaches on the 8 numeric datasets

We use 5-fold cross validation to experimentally compare the proposed method GANDO with the 11 SMOTE related state-of-the-art approaches on 5 aspects: MMD-score, Silhouette-score, F-measure, G-means, and AUC-area, and the generated synthetic samples are visualized on the artificial dataset to demonstrate effectiveness and superiority of the proposed approach GANDO. The experimental results of MMD-score compared with the 11 SMOTE related state-of-the-art approaches on the 8 numeric datasets are given in Table 7, and the experimental results of Silhouette-score compared with the 11 SMOTE related state-of-the-art approaches on the 8 numeric datasets are given in Table 8.

From the experimental results listed in Table 7, the MMD-scores of the proposed method GANDO on 7 numeric datasets are greater than the ones of the 10 SMOTE related state-of-the-art approaches, which means that the oversampled positive class samples by GANDO have better diversities than the 10 SMOTE related state-of-the-art approaches. This conclusion is further confirmed by the visualization of the generated synthetic positive class samples on the artificial dataset (see Fig. 4). In the Fig. 4, the yellow "–" represents the negative class sample, the blue "+" represents the positive class sample, and the red "+" represents the generated positive class sample. It can be seen from the Fig. 4 that the samples generated by the proposed method GANDO have better diversity than the 11 SMOTE state-of-the-art approaches. Although MFC-GAN has good diversity, it has bad separability, i.e. the generated synthetic positive class samples overlap with the original negative samples. K-SMOTE is an exception that K-SMOTE can not generate synthetic positive class samples on the artificial dataset.

Table 14 Experimental comparison of F-measure on the 3 image datasets

Approaches	MNIST	Fashion-MNIST	Cifar10
AUGMENT	0.86	0.50	0.22
GAN	0.89	0.62	0.60
AC-GAN	0.87	0.66	0.12
MFC-GAN	0.88	**0.71**	0.67
GANDO	**0.96**	0.69	**0.80**

The maximum values are in bold, indicating the best performance

Table 15 Experimental comparison of G-mean on the 3 image datasets

Approaches	MNIST	Fashion-MNIST	Cifar10
AUGMENT	0.93	0.93	0.09
GAN	0.88	0.92	0.65
AC-GAN	0.84	0.89	0.17
MFC-GAN	0.88	0.93	0.71
GANDO	**0.99**	**0.94**	**0.76**

The maximum values are in bold, indicating the best performance

Table 16 Experimental comparison of AUC-area on the 3 image datasets

Approaches	MNIST	Fashion-MNIST	Cifar10
AUGMENT	0.86	0.50	0.08
GAN	0.89	0.62	0.64
AC-GAN	0.87	0.66	0.17
MFC-GAN	0.88	**0.71**	0.68
GANDO	**0.96**	0.69	**0.77**

The maximum values are in bold, indicating the best performance

This is due to its oversampling mechanism: K-SMOTE first use K-means to cluster the artificial dataset, and then for each cluster, K-SMOTE calculates it's IR, and select the clusters whose IR is less than a threshold for oversampling with SMOTE. In our experiments, the threshold is set to 2.0. Since the IR of each cluster is greater than 2.0, no oversampling is performed.

It is well known that the better the diversity of generated synthetic positive class samples, the better the quality of the generated synthetic positive class samples. High quality generated synthetic positive class samples can effectively expand the training field of positive class samples, and effectively improve the performance of the proposed classification algorithm. This point is confirmed by the experimental results on three classification performance metrics: F-measure, G-means, and AUC-area (see Tables 9, 10, 11). The reason why the proposed method GANDO can generate synthetic positive class samples with good diversity is that we introduce a regularization term of intra-class divergence into the loss function of the improved GAN model.

From the experimental results listed in Table 8, the Silhouette-scores of the proposed method GANDO on 7 numeric datasets are greater than the ones of the 10 SMOTE related state-of-the-art approaches, which demonstrates that the oversampled positive class samples by GANDO also have better separability than the 10 SMOTE related state-of-the-art approaches. This conclusion is further confirmed by the visualization of the generated synthetic positive class samples on the artificial dataset (see Fig. 4). It can be seen from the Fig. 4 that the samples generated by the proposed method GANDO has better separability than the 11 SMOTE related state-of-the-art approaches. Although B-SMOTE, ANS, and NRPSOS have good separability, they have low diversities.

The experimental results of F-measure, G-means and AUC-area compared with the 11 SMOTE related state-of-the-art approaches on the 8 numeric datasets are given in Tables 9, 10, 11 respectively. From the experimental results listed in Tables 9, 10, 11, it is observed that (a) the F-measure of the proposed method GANDO are greater than the ones of the 11 SMOTE related state-of-the-art approaches on the 8 numeric datasets; (b) the G-means and AUC-area of the proposed method GANDO are greater than the ones of the 11 SMOTE related state-of-the-art approaches on 6 and 7 numeric datasets respectively. Overall, the performance of the proposed method GANDO outperforms the 11 SMOTE related state-of-the-art approaches in terms of F-measure, G-means, and AUC-area. We think that the reasons include the following three points:

1. Introducing the regularization term of intra-class divergence into the loss function of the GAN can guarantee good diversity of the generated synthetic positive class samples. Good diversity can effectively expand the training field of the positive class samples.
2. Introducing the Silhouette-score can guarantee good separability between the generated synthetic positive class samples and negative, and the combination of MMD-score and Silhouette-score can further improve the quality of the generated synthetic positive class samples, all of which contribute to the good performance of the proposed method GANDO.
3. Since the base classifiers are trained on balanced training sets containing the same set of oversampling positive class samples, intrinsic interactions exist among different base classifiers. The interactions may be positively correlated, in this case, the base classifiers enhance each other. The interactions may also be negatively correlated, in this case, the base classifiers suppress each other. Fuzzy integral can well model the interactions among the base classifiers, which enhance the generalization performance of the ensemble classifier.

From the experimental results on F-measure and G-mean listed in Tables 9 and 10, we find that some traditional methods (e.g. ADASYN, CCR, ANS, etc) obtained exceptional results on liver 1 dataset, we believe that the reason for the exceptional results is that this dataset has very high IR. Yet, the proposed method GANDO obtained competitive result on this severely imbalanced dataset.

4.5 Comparisons with the 4 GAN related state-of-the-art methods on the 3 image datasets

It is well known that GAN can generate realistic images, which can be viewed as a data augmentation technique, while oversampling is also a data augmentation technique. In order to further demonstrate the effectiveness of GANDO for classifying imbalanced image datasets, we conduct experiments on three famous images datasets to compare GANDO with 4 GAN related state-of-the-art methods, AUGMENT, GAN, AC-GAN, and MFC-GAN. The experimental results of the 5 evaluation measures compared with the 4 GAN related state-of-the-art approaches on the 3 image datasets are listed in Tables 12, 13, 14, 15 and 16.

From the experimental results of MMD-score listed in Table 12, we find that the proposed method GANDO obtained 1 maximum on fashion-MNIST, AC-GAN obtained the other two maxima on datasets MNIST and Cifar10. However, AC-GAN has bad separability on the three image datasets, while GANDO has much better separability than AC-GAN (see Table 13). In other words, GANDO achieves the optimal tradeoff between diversity and separability, this will result in that GANDO outperforms AC-GAN on classification performance, which can be confirmed by the experimental results of F-measure, G-means, and AUC-area listed Tables 14, 15 and 16. From the experimental results of Silhouette-score listed in Table 13, we find that the proposed method GANDO obtained 2 maximum on MNIST and Cifar10, MFC-GAN obtained the other maxima on datasets fashion-MNIST. Compared GANDO and MFC-GAN on three classification performance measures, i.e. F-measure, G-means, and AUC-area, GANDO is superior to MFC-GAN on 2 image datasets MNIST and Cifar10, and MFC-GAN is superior to GANDO on image dataset fashion-MNIST. In summary, GANDO outperforms other 4 GAN related state-of-the-art methods.

5 Conclusions

Based on an improved GAN model and a classifier fusion mechanism via fuzzy integral, a framework for classifying imbalanced data was proposed in this paper. The framework contains an oversampling method and an ensemble classification approach for the classification of imbalanced data. The oversampling method is based on the improved GAN model by introducing a regularization term of intra-class divergence into the loss function of the GAN, and replacing the discriminator of GAN with a classifier with three outputs. The ensemble classification approach is based on fuzzy integral. Since the base classifiers are trained on balanced training sets containing the same positive class set, there are intrinsic interactions among the base classifiers. Fuzzy integral can well model the interactions, thus effectively enhance the classification performance. The proposed classification framework has four advantages: (1) It can generate synthetic positive class samples with good diversity and good separability. (2) The improved GAN model can effectively avoid mode collapse. (3) It has good classification generalization performance due to diverse oversampling and controllable separability. (4) It is effective not only for datasets with medium imbalanced ratio, but also for datasets with very high imbalanced ratio. The promising future works of this study include (1) extending GANDO to multi-class imbalanced data classification; (2) expanding the scalability of GANDO for imbalanced big data scenarios.

Acknowledgements This research is supported by the national natural science foundation of China (71371063), by the key R&D program of science and technology foundation of Hebei Province (19210310D), by the natural science foundation of Hebei Province (F2017201026), by Postgraduate's Innovation Fund Project of Hebei University (hbu2019ss077).

References

1. Japkowicz N (2000) The class imbalance problem: significance and strategies. In: Proceedings of the 2000 international conference on artificial intelligence, pp 111–117
2. Malhotra R, Kamal S (2019) An empirical study to investigate oversampling methods for improving software defect prediction using imbalanced data. Neurocomputing 343:120–140
3. Zhou J, Liu Y, Zhang TH (2019) Fault diagnosis based on relevance vector machine for fuel regulator of aircraft engine. Int J Mach Learn Cybern 10(7):1779–1790
4. Dhingra K, Yadav SK (2019) Spam analysis of big reviews dataset using fuzzy ranking evaluation algorithm and Hadoop. Int J Mach Learn Cybern 10(8):2143–2162
5. Wang S, Yao X (2012) Multiclass imbalance problems: analysis and potential solutions. IEEE Trans Syst Man Cybern Part B Cybern 42(4):1119–1129
6. Bia JJ, Zhang CS (2018) An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme. Knowl Based Syst 158:81–93
7. García V, Sánchez JS, Marqués AI et al (2020) Understanding the apparent superiority of over-sampling through an analysis of local information for class-imbalanced data. Expert Syst Appl 158:113026
8. Goodfellow I, Pouget-Abadie J, Mirza M et al (2014) Generative adversarial nets. Adv Neural Inf Process Syst 1:2672–2680
9. Branco P, Torgo L, Ribeiro R (2016) A survey of predictive modeling on imbalanced domains. ACM Comput Surv 49(2):1–50
10. Krawczyk B (2016) Learning from imbalanced data: open challenges and future directions. Prog Artif Intell 5:221–232
11. Sun Y, Kamel MS, Wong AK et al (2007) Cost-sensitive boosting for classification of imbalanced data. Pattern Recognit 40(12):3358–3378
12. Khan SH, Hayat M, Bennamoun M et al (2018) Cost-sensitive learning of deep feature representations from imbalanced data. IEEE Trans Neural Netw Learn Syst 29(8):3573–3587
13. Tao X, Li Q, Guo W et al (2019) Self-adaptive cost weights-based support vector machine cost-sensitive ensemble for imbalanced data classification. Inf Sci 487:31–56
14. Wang Z, Wang B, Cheng Y et al (2019) Cost-sensitive fuzzy multiple kernel learning for imbalanced problem. Neurocomputing 366:178–193
15. Wang CZ, Wang Y, Shao MW et al (2020) Fuzzy rough attribute reduction for categorical data. IEEE Trans Fuzzy Syst 28(5):818–830
16. Wang CZ, Huang Y, Shao MW et al (2020) Feature selection based on neighborhood self-information. IEEE Trans Cybern 50(9):4031–4042
17. Wang CZ, Huang Y, Shao MW et al (2019) Fuzzy rough set-based attribute reduction using distance measures. Knowl Based Syst 164:205–212
18. Ni P, Zhao SY, Wang XZ et al (2020) Incremental feature selection based on fuzzy rough sets. Inf Sci 536:185–204
19. Ni P, Zhao SY, Wang XZ et al (2019) PARA: a positive-region based attribute reduction accelerator. Inf Sci 503:533–550
20. Kovács G (2019) An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets. Appl Soft Comput J 83:105662
21. Elreedy D, Atiya AF (2019) A comprehensive analysis of synthetic minority oversampling technique (SMOTE) for handling class imbalance. Inf Sci 505:32–64
22. Fernández A, García 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

23. Chawla NV, Bowyer KW, Hall LO et al (2002) SMOTE: synthetic minority oversampling technique. J Artif Intell Res 16:321–357
24. 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
25. Douzas G, Bacao F (2019) Geometric SMOTE a geometrically enhanced drop-in replacement for SMOTE. Inf Sci 501:118–135
26. Maldonado S, López J, Vairetti C (2019) An alternative SMOTE oversampling strategy for high-dimensional datasets. Appl Soft Comput 76:380–389
27. Susan S, Kumar A (2019) SSO $_{Maj}$ -SMOTE-SSO $_{Min}$: three-step intelligent pruning of majority and minority samples for learning from imbalanced datasets. Appl Soft Comput 78:141–149
28. Mathew J, Pang CK, Luo M et al (2018) Classification of imbalanced data by oversampling in kernel space of support vector machines. IEEE Trans Neural Netw Learn Syst 29(9):4065–4076
29. Raghuwanshi SS, Shukla S (2020) SMOTE based class-specific extreme learning machine for imbalanced learning. Knowl Based Syst 187:104814
30. Pan TT, Zhao JH, Wu W et al (2020) Learning imbalanced datasets based on SMOTE and Gaussian distribution. Inf Sci 512:1214–1233
31. Zhang H, Li M (2014) RWO-sampling: a random walk oversampling approach to imbalanced data classification. Inf Fusion 20:99–116
32. Han X, Cui R, Lan Y et al (2019) A Gaussian mixture model based combined resampling algorithm for classification of imbalanced credit datasets. Int J Mach Learn Cybern 10:3687–3699
33. Zhang CK, Zhou Y, Guo JW et al (2019) Research on classification method of high-dimensional class imbalanced datasets based on SVM. Int J Mach Learn Cybern 10:1765–1778
34. Odena A, Olah C, Shlens J (2017) Conditional image synthesis with auxiliary classifier GANs. Proc Int Conf Mach Learn 70:2642–2651
35. Ali-Gombe A, Elyan E (2019) MFC-GAN: class-imbalanced dataset classification using multiple fake class generative adversarial network. Neurocomputing 361:212–221
36. Douzas G, Bacao F (2018) Effective data generation for imbalanced learning using conditional generative adversarial networks. Expert Syst Appl 91:464–471
37. Zheng M, Li T, Zhu R et al (2020) Conditional Wasserstein generative adversarial network-gradient penalty-based approach to alleviating imbalanced data classification. Inf Sci 512:1009–1023
38. Arjovsky M, Chintala S, Bottou L (2017) Wasserstein generative adversarial networks. In: International conference on machine learning, pp 214–223
39. Sun J, Li H, Fujita H et al (2020) Class-imbalanced dynamic financial distress prediction based on Adaboost-SVM ensemble combined with SMOTE and time weighting. Inf Fusion 54:128–144
40. González S, García S, Lázaro M et al (2017) Class switching according to nearest enemy distance for learning from highly imbalanced data-sets. Pattern Recognit 70:12–24
41. Gutiérrez-López A, Gutiérrez-López FJA, Figueiras-Vidal AR (2020) Asymmetric label switching resists binary imbalance. Inf Fusion 60:20–24
42. Raghuwanshi BS, Shukla S (2019) Classifying imbalanced data using ensemble of reduced kernelized weighted extreme learning machine. Int J Mach Learn Cybern 10:3071–3097
43. Hsiao YH, Su CT, Fu PC (2020) Integrating MTS with bagging strategy for class imbalance problems. Int J Mach Learn Cybern 11:1217–1230
44. Zhai JH, Zhang SF, Wang CX (2017) The classification of imbalanced large datasets based on MapReduce and ensemble of ELM classifiers. Int J Mach Learn Cybern 8(3):1009–1017
45. Abdallah ACB, Frigui H, Gader P (2012) Adaptive local fusion with fuzzy integrals. IEEE Trans Fuzzy Syst 20(5):849–864
46. Zhan YZ, Zhang J, Mao QR (2012) Fusion recognition algorithm based on fuzzy density determination with classification capability and supportability. Pattern Recognit Artif Intell 25(2):346–351
47. Han H, Wang WY, Mao BH (2005) Borderline-SMOTE: a new oversampling method in imbalanced datasets learning. In: International conference on advances in intelligent computing. Springer, pp 878–887
48. He HB, Bai Y, Garcia EA et al (2008) ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: IEEE international joint conference on neural networks, IJCNN, pp 1322–1328
49. Koziarski M, Wozniak M (2017) CCR: a combined cleaning and resampling algorithm for imbalanced data classification. Int J Appl Math Comput Sci 27:727–736
50. Siriseriwan W, Sinapiromsaran K (2017) Adaptive neighbor synthetic minority oversampling technique under 1NN outcast handling. Songklanakarin J Sci Technol 39(5):565–576
51. Rivera WA (2017) Noise reduction a priori synthetic over-sampling for class imbalanced datasets. Inf Sci 408:146–161
52. Rivera WA, Xanthopoulos P (2016) A priori synthetic over-sampling methods for increasing classification sensitivity in imbalanced datasets. Expert Syst Appl 66:124–135
53. Brownlee J (2016) Image augmentation for deep learning with Keras. https://machinelearningmastery.com/image-augmentation-deep-learning-keras
54. Alcalá-Fdez J, Fernandez A, Luengo J et al (2011) KEEL datamining software tool: dataset repository, integration of algorithms and experimental analysis framework. J Mult Valued Logic Soft Comput 17(2–3):255–287
55. Zhai JH, Zhang SF, Zhang MY et al (2018) Fuzzy integral-based ELM ensemble for imbalanced big data classification. Soft Comput 22(11):3519–3531
56. Gretton A, Borgwardt KM, Rasch M et al (2016) A kernel method for the two-sample problem. In: Advances in neural information processing systems, vol 19 (NIPS), pp 1672–1679
57. Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65
58. He HB, Garcia EA (2009) Learning from imbalanced data. IEEE Trans Knowl Data Eng 21(9):1263–1284