IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received August 24, 2020, accepted September 3, 2020, date of publication September 14, 2020, date of current version September 25, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3023949

Binary Imbalanced Data Classification Based on Modified D2GAN Oversampling and Classifier Fusion

JUNHAI ZHAI[1], (Member, IEEE), JIAXING QI[1], AND SUFANG ZHANG[2]

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

[2]Hebei Branch of China Meteorological Administration Training Center, China Meteorological Administration, Baoding 071000, China

Corresponding author: Sufang Zhang (mczsf@126.com)

This work was supported in part by the Key Research and Development Program of Science and Technology Foundation of Hebei Province under Grant 19210310D, in part by the Natural Science Foundation of Hebei Province under Grant F2017201026, and in part by the Postgraduate's Innovation Fund Project of Hebei University under Grant hbu2019ss077.

ABSTRACT Binary imbalance problem refers to such a classification scenario where one class contains a large number of samples while another class contains only a few samples. When traditional classifiers face with imbalanced datasets, they usually bias towards majority class resulting in poor classification performance. Oversampling is an effective method to address this problem, yet how to conduct diversity oversampling is a challenge. In this article, we proposed a diversity oversampling method based on a modified D2GAN model, and on the basis of diversity oversampling, we also proposed a binary imbalanced data classification approach based on classifier fusion by fuzzy integral. Extensive experiments are conducted on 8 data sets to compare the proposed methods with 7 state-of-the-art methods on 5 aspects: MMD-score, Silhouette-score, F-measure, G-means, and AUC-area. The 7 methods include 4 SMOTE related approaches and 3 GAN related approaches. The experimental results demonstrate that the proposed methods are more effective and efficient than the compared approaches.

INDEX TERMS Binary class imbalance, diversity oversampling, generative adversarial network, classifier fusion, fuzzy integral.

I. INTRODUCTION

In the real world, there are many binary imbalanced data classification problems [1], such as software defect prediction [2], machinery fault diagnosis [3], spam filtering [4], extreme weather prediction [5], and so on. In the last few decades, many oversampling methods are proposed by different researchers, among which SMOTE [6] is the most influential oversampling approach. However, SMOTE has three drawbacks:

1) It doesn't take into account the probability distribution of the minority class samples;
2) The samples generated by SMOTE lack diversity;
3) If SMOTE is iterated many times, the generated synthetic samples overlap heavily.

The associate editor coordinating the review of this manuscript and approving it for publication was Keli Xiao.

The main contributions of this article include the following three folds:

1) We proposed a diversity oversampling method based on modified D2GAN model, the modification lies in introducing a classifier to the D2GAN model for learning the difference between positive samples and negative samples, so as to ensure the correctness of the class of the generated samples. Consequently, there is no overlap between the generated samples and the majority class samples.
2) Based on the proposed oversampling method, we designed an ensemble classification algorithm by fuzzy integral for binary imbalanced data classification. Fuzzy integral can well model the interactions among the base classifiers, the interactions may be positive correlated, also may be negative correlated. The proposed algorithm can boost the classification accuracy of the positive class samples.

3) Extensive experiments are conducted on 8 data sets to compare the proposed methods with 7 state-of-the-art 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.

The rest of this article is organized as follows. In section II, we review SMOTE related works. In section III, we describe the details of the proposed methods. In section IV, Experimental results and analyses are presented. At last, we conclude our work in the section V.

II. RELATED WORKS

Since from SMOTE was proposed by Chawla *et al.* in 2002, many variants of SMOTE have been proposed in the past 18 years. Han *et al.* [7] found that the samples on or near the decision boundary are more apt to be misclassified than the ones far from the borderline, and thus more important for classification. On the contrary, those examples far from the borderline may contribute little to classification. Based on this observation, they proposed Borderline-SMOTE in which only the borderline examples of the minority class are over-sampled. He *et al.* [8] proposed a novel adaptive synthetic sampling approach named ADASYN which is based on the idea of adaptively generating minority samples according to their distributions: more synthetic data is generated for minority class samples that are harder to learn. Along this technical route, Barua *et al.* [9] propsoed MWMOTE which first identifies the hard-to-learn informative minority class samples and assigns them weights according to their Euclidean distance from the nearest majority class samples. It then generates the synthetic samples from the weighted informative minority class samples using a clustering approach. In the processing of generating synthetic minority samples, SMTOTE ignores majority instances. Bunkhumpornpat *et al.* [10] proposed Safe-Level-SMOTE to deal with this problem, it carefully samples minority instances along the same line with different weight degree, called safe level. The safe level is computed using nearest neighbour minority instances. Bunkhumpornpat *et al.* [11] proposed a density-based clustering over-sampling technique called DBSMOTE which generates synthetic instances along a shortest path from each positive instance to a pseudocentroid of a minority-class cluster. Similarly, Douzas *et al.* [12] proposed an oversampling method based on k-means clustering and SMOTE. Douzas and Bacao [13] proposed a geometric SMOTE which generates synthetic samples in a hypersphere around each selected minority instance. Mathew *et al.* [14] proposed a weighted kernel-based SMOTE (WK-SMOTE) approach which generates synthetic positive class samples in feature space, WK-SMOTE can overcome the limitation of linear interpolation of SMOTE. Based on WK-SMOTE, Raghuwanshi and Shukla [15] proposed a SMOTE based class-specific extreme learning machine, which exploits the benefit of both the minority oversampling and the class-specific regularization. Maldonado *et al.* [16] studied SMOTE oversampling strategy for high-dimensional datasets, and proposed an alternative distance metric for computing the neighbours for each minority sample. Tao *et al.* [17] proposed an over-sampling method which uses the real-value negative selection procedure to generate synthetic minority samples, its novelty is that it does not require minority class instance available, and only relies on majority class instances. Based on density peaks clustering with heuristic filtering, Tao *et al.* [18] also proposed an adaptive weighted over-sampling method for imbalanced data classification. The main advantages of this method are twofold: (1) the between-class and within-class imbalance issues can be simultaneously addressed; (2) the weights for synthetic instance generation can be adaptively determined. Shamsolmoali *et al.* [19] proposed an oversampling method which uses capsule adversarial networks to augment minority class samples. Three excellent survey papers on SMOTE can be found in [20]–[22].

SMOTE can be integrated into ensemble learning for imbalanced data classification, this strategy is very effective for high imbalanced data classification. Chawla *et al.* [23] combined SMOTE and boosting algorithm, and proposed SMOTEBoost. Different from the weight updating strategy in boosting, SMOTEBoost creates synthetic examples from minority class, and indirectly change the updating weights and compensating for skewed distributions. Wang and Yao [24] put SMOTE and bagging together, and proposed SMOTEBagging algorithm which can overcome over-fitting problem. But the way it produces compositional samples simply setting similar individuals' nearest neighbors to a uniform number, and without considering the samples' real distribution and their neighbors' distribution. For this reason, Zhang *et al.* [25] proposed an improved SMOTEBagging algorithm. Based on SMOTE combining with adaboost support vector machine ensemble, Sun *et al.* [26] proposed two class imbalanced dynamic financial distress prediction approaches. Zhai *et al.* [27] combined oversampling based on enemy nearest neighbor hypersphere and ensemble learning, and proposed a MapReduce based classification algorithm for large scale imbalance data. Inspired by the localized generalization error model, Chen *et al.* [28] propsoed an oversampling method, which generates some synthetic samples located within some local area of the training samples, and combined the oversampling method with ensemble learning for imbalanced data classification. Tao *et al.* [29] proposed a cost-sensitive ensemble approach for imbalanced data classification, it uses support vector machine as as basic weak leaner, and uses AdaBoost as ensemble mechanism. The novelty of this method lie in introducing a self-adaptive cost weights strategy. Wong *et al.* [30] proposed a cost-sensitive stacked denoising autoencoder ensemble method, and applied to address class imbalance problems in business domain. Zhu *et al.* [31] proposed a geometric structural ensemble learning framework, which partitions and eliminates redundant majority samples by generating hyper-sphere through the Euclidean metric and learns basic classifiers to enclose the minority samples. Yang *et al.* [32] proposed a hybrid ensemble classifier framework that combines density-based undersampling and cost-effective methods using multi-objective optimization algorithm to handle two issues: (1) undersampling methods suffer from losing important information; (2) cost-sensitive methods are sensitive to outliers and noise.

In recent years, generative adversarial network (GAN) [33] is a very hot research topic in deep learning, some researchers use the generation mechanism of GAN to generate synthetic positive class samples for balancing imbalanced data set. For instance, inspired by the idea of AC-GAN (Auxiliary Classifier-GAN) [34], Ali-Gombe and Elyan proposed an improved model MFC-GAN (Multiple Fake Class-GAN) [35] and used the MFC-GAN to handle imbalanced data classification problem. Zheng *et al.* [36] proposed a synthetic oversampling approach for imbalanced data sets.

In this article, we presented a binary imbalanced data classification algorithm D2GANDO which combines diversity oversampling and ensemble learning. The diversity oversampling is conducted by a modified D2GAN [37]. The ensemble incorporated the idea of fuzzy integral [38] which can well model the interaction among the base classifiers, as a result, the generalization ability of the classification algorithm can be effectively enhanced. Different from existing methods based on GAN, the novelty of D2GANDO lies in (a) introducing a classifier to D2GAN to guarantee diversity of the synthetic samples. (b) introducing MMD-score and Silhouette-score to measure diversity and separability, both of which have important influence on the performance of imbalanced data classification.

III. THE PROPOSED ALGORITHM

In this section, we present the proposed binary imbalanced data classification algorithm which includes a modified D2GAN based oversampling method and an ensemble approach by fuzzy integral for binary imbalanced data classification.

A. DIVERSITY OVERSAMPLING METHOD BASED ON MODIFIED D2GAN

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

Since GAN is a probabilistic generative model, it is a natural to use GAN to generate synthetic positive class samples for balancing imbalanced data set. But GAN is prone to mode collapse, whilst D2GAN is tailored for addressing the mode collapse problem. D2GAN has two discriminators, the one rewards high scores for samples from data distribution whilst the another favours data from the generator, and the generator produces data to fool both two discriminators. Furthermore, D2GAN combines the KL divergence and reverse KL divergence into a unified objective function, which insures to effectively diversify the estimated density in capturing multi-modes. As a result, the samples generated by D2GAN have good diversity. We found that although the samples generated by D2GAN have good diversity, the separability between classes is poor. To this end, we modified D2GAN model by introducing a classifier C (see Figure 2). Its output is a three-dimensional vector, the three components p_{pos}, p_{neg}, and p_g are the support degrees for positive class, negative class and generated samples respectively. In the adversarial training process, we want the samples generated by generator G to fool the classifier C, that is, when the samples are 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 can learn a good classification boundary between positive class and negative class.

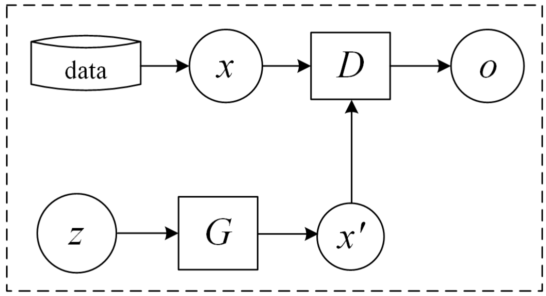

FIGURE 1. The architecture of generative adversarial network.

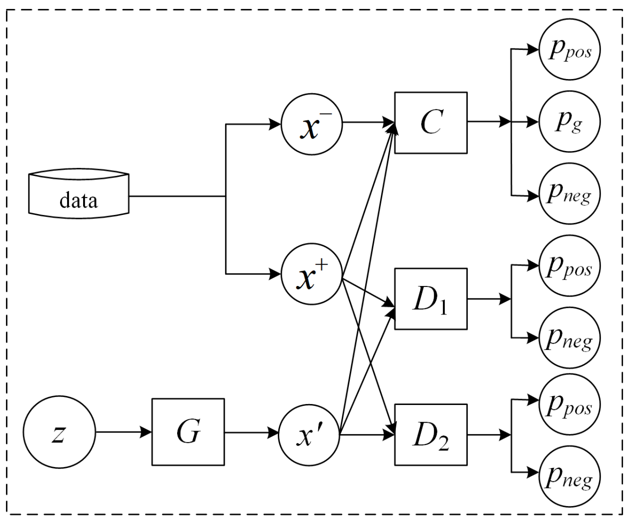

FIGURE 2. The architecture of the modified D2GAN.

The objective functions of the modified D2GAN model are given by Eq.(1), Eq.(2) and Eq.(3).

$$\begin{aligned}\max_{D_1,D_2} L(D_1,D_2) &= \alpha \times \mathbb{E}_{\boldsymbol{x}\sim p_{pos}}[\log D_1(\boldsymbol{x})] \\ &\quad + \mathbb{E}_{z\sim p_z}[-D_1(G(z))] \\ &\quad + \mathbb{E}_{\boldsymbol{x}\sim p_{pos}}[-D_2(\boldsymbol{x})] \\ &\quad + \beta \times \mathbb{E}_{z\sim p_z}[\log D_2(G(z))] \quad (1)\end{aligned}$$

$$\max_C L(C) = J_1 + J_2 + J_3 \quad (2)$$

$$\max_G L(G) = L(D_1,D_2) - J_4 \quad (3)$$

where $D_i(\cdot)(i=1,2)$ denote the output of discriminator D_i, $\mathbb{E}$ is expectation operator. α and β are two parameters, and $0 \le \alpha, \beta \le 1$.

$$\begin{aligned}J_1 &= \mathbb{E}_{\boldsymbol{x}\sim p_{neg}} \log C_1(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{x}\sim p_{neg}} \log(1-C_2(\boldsymbol{x})) \\ &\quad + \mathbb{E}_{\boldsymbol{x}\sim p_{neg}} \log(1-C_3(\boldsymbol{x})) \\ J_2 &= \mathbb{E}_{\boldsymbol{x}\sim p_{pos}} \log C_2(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{x}\sim p_{pos}} \log(1-C_1(\boldsymbol{x})) \\ &\quad + \mathbb{E}_{\boldsymbol{x}\sim p_{pos}} \log(1-C_3(\boldsymbol{x})) \\ J_3 &= \mathbb{E}_{\boldsymbol{x}\sim p_g} \log C_3(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{x}\sim p_g} \log(1-C_1(\boldsymbol{x})) \\ &\quad + \mathbb{E}_{\boldsymbol{x}\sim p_g} \log(1-C_2(\boldsymbol{x})) \\ J_4 &= \mathbb{E}_{\boldsymbol{x}\sim p_g} \log C_2(\boldsymbol{x}) \\ &\quad - \mathbb{E}_{\boldsymbol{x}\sim p_g} \log C_1(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{x}\sim p_g} \log C_3(\boldsymbol{x})\end{aligned}$$

where $C_i(\cdot)(i=1,2,3)$ represent the posterior probability that the output of the classifier C is negative, positive, and generated respectively.

On the premise of obtaining the two optimal discriminators and an optimal classifier, the optimization goal of the generator becomes:

$$\begin{aligned}\min_G L(G) &= \alpha(\log\alpha - 1) + \beta(\log\beta - 1) \\ &\quad + \alpha D_{KL}(p_{pos} \parallel p_g) \\ &\quad + (\beta+1)D_{KL}(p_g \parallel p_{pos}) - H(p_g, p_{neg}) \quad (4)\end{aligned}$$

where $D_{KL}(\cdot \parallel \cdot)$ is the KL divergence between two distributions, $H(\cdot,\cdot)$ is the cross entropy between two distributions.

In the following, we proof that Eq. (4) is hold. Take the partial derivative of $L(D_1,D_2)$ with respect to D_1 and D_2, and set them equal to zero, we can obtain.

$$D_1^*(\boldsymbol{x}) = \frac{\alpha p_{pos}(\boldsymbol{x})}{p_g(\boldsymbol{x})}$$

and

$$D_2^*(\boldsymbol{x}) = \frac{\beta p_g(\boldsymbol{x})}{p_{pos}(\boldsymbol{x})}$$

Take the partial derivative of $L(C)$ with respect to C_1, and set it equal to zero, we can obtain.

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

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 D_1^*, D_2^*, $C_1^*(\boldsymbol{x})$, $C_2^*(\boldsymbol{x})$ and $C_3^*(\boldsymbol{x})$ into (3), we have,

$$\begin{aligned}L(G) &= \alpha\mathbb{E}_{\boldsymbol{x}\sim p_{pos}}\left[\log\alpha + \log\frac{p_{pos}(\boldsymbol{x})}{p_g(\boldsymbol{x})}\right] \\ &\quad - \alpha\int_x p_g(\boldsymbol{x})\frac{p_{pos}(\boldsymbol{x})}{p_g(\boldsymbol{x})}d\boldsymbol{x} \\ &\quad - \beta\int_x p_{pos}(\boldsymbol{x})\frac{p_g(\boldsymbol{x})}{p_{pos}(\boldsymbol{x})}d\boldsymbol{x} \\ &\quad + \beta\mathbb{E}_{\boldsymbol{x}\sim p_g}\left[\log\beta + \log\frac{p_g(\boldsymbol{x})}{p_{pos}(\boldsymbol{x})}\right] \\ &\quad - \int_x p_g(\boldsymbol{x})\log\frac{p_{pos}(\boldsymbol{x})}{p_g(\boldsymbol{x})}d\boldsymbol{x} \\ &\quad + \int_x p_g(\boldsymbol{x})\log p_{neg}(\boldsymbol{x})d\boldsymbol{x} \\ &= \alpha(\log\alpha - 1) + \beta(\log\beta - 1) \\ &\quad + \alpha D_{KL}(p_{pos} \parallel p_g) \\ &\quad + (\beta+1)D_{KL}(p_g \parallel p_{pos}) - H(p_g, p_{neg})\end{aligned}$$

From Eq.(4), it is easy to find that compared with D2GAN, the optimization target of G in modified D2GAN increases the cross entropy loss between the generated distribution and the negative class distribution. In the process of optimizing G, the model introduces the information of negative class sample distribution to prevent the overlap between classes. Therefore, the modified D2GAN model can not only ensure the diversity of generated samples through the double discriminators, but also avoid the overlap between generated samples and negative class samples by introducing a classifier which can capture the negative class sample distribution information. The pseudo code of the proposed oversampling algorithm D2GANDO is given in algorithm 1.

B. BINARY IMBALANCED DATA CLASSIFICATION APPROACH BASED ON CLASSIFIER FUSION BY FUZZY INTEGRAL

On the basis of the above oversampling method, we proposed a binary imbalanced data classification approach based on classifier fusion by fuzzy integral. 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 $S_1^-, S_2^-, \cdots, S_l^-$, 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 $C = \{C_1, C_2, \cdots, C_l\}$ on the l balance training sets. The l classifiers are fused for imbalanced data classification by fuzzy integral in the next stage.

Algorithm 1 Oversampling Algorithm D2GANDO

Input: Imbalanced data set $S = S^{+} \cup S^{-}$, the size of batch m, the iterative number n, and the number of training t.

Output: S_{up}^{+}.

Initialize the parameter $\boldsymbol{\theta}_g$ of generator G, the parameter $\boldsymbol{\theta}_{d_1}$ of discriminator D_1, the parameter $\boldsymbol{\theta}_{d_2}$ of discriminator D_2, and the parameter $\boldsymbol{\theta}_c$ of classifier C with small random numbers.

for $(i = 1; i \leq n; i = i + 1)$ **do**

  for $(j = 1; j \leq t; j = j + 1)$ **do**

    Sample m samples from noise prior distribution p_z, and input them to G, obtain m generated samples $\{\boldsymbol{x}_1^g, \boldsymbol{x}_2^g, \cdots, \boldsymbol{x}_m^g\}$;

    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 $\boldsymbol{\theta}_g$, $\boldsymbol{\theta}_{d_1}$, and $\boldsymbol{\theta}_{d_2}$, update $\boldsymbol{\theta}_c$ by ascending its stochastic gradient;

    Sample m samples from noise prior distribution p_z, and input them to G, obtain m generated samples $\{\boldsymbol{x}_1^g, \boldsymbol{x}_2^g, \cdots, \boldsymbol{x}_m^g\}$;

    Fix $\boldsymbol{\theta}_g$ and $\boldsymbol{\theta}_c$, update $\boldsymbol{\theta}_{d_1}$, and $\boldsymbol{\theta}_{d_2}$ by ascending its stochastic gradient;

    Sample m samples from noise prior distribution p_z, and input them to G, obtain m generated samples $\{\boldsymbol{x}_1^g, \boldsymbol{x}_2^g, \cdots, \boldsymbol{x}_m^g\}$;

    Fix $\boldsymbol{\theta}_{d_1}$, $\boldsymbol{\theta}_{d_2}$, and $\boldsymbol{\theta}_c$, update $\boldsymbol{\theta}_g$ by ascending its stochastic gradient;

  end

  Sample m samples from noise prior distribution p_z, and input them to G, obtain a set of m generated samples $S_g = \{\boldsymbol{x}_1^g, \boldsymbol{x}_2^g, \cdots, \boldsymbol{x}_m^g\}$;

  Let $S_{up}^{+} = S_{up}^{+} \cup S_g$;

end

Return S_{up}^{+}.

(2) Fuse the trained base classifiers by fuzzy integral

As a classifier fusion method, fuzzy integral is distinguished from other fusion methods due to its an intriguing property, that is it can well model the interaction 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, $1 \leq i \leq n$, $Y = \{\omega_1, \omega_2, \cdots, \omega_k\}$ be a set of class labels, $C = \{C_1, C_2, \cdots, C_l\}$ 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 $(p_{i1}(\boldsymbol{x}), p_{i2}(\boldsymbol{x}), \ldots, p_{ik}(\boldsymbol{x}))$. The $p_{ij}(\boldsymbol{x}) \in [0, 1](1 \leq i \leq l; 1 \leq j \leq k)$ 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 $C = \{C_1, C_2, \cdots, C_l\}$, $Y = \{\omega_1, \omega_2, \cdots, \omega_k\}$, and arbitrary testing sample $\boldsymbol{x}$. The following matrix is called decision matrix with respect to $\boldsymbol{x}$.

$$DM(\boldsymbol{x}) = \begin{bmatrix} p_{11}(\boldsymbol{x}) & \cdots & p_{1j}(\boldsymbol{x}) & \cdots & p_{1k}(\boldsymbol{x}) \\ \vdots & & \vdots & & \vdots \\ p_{i1}(\boldsymbol{x}) & \cdots & p_{ij}(\boldsymbol{x}) & \cdots & p_{ik}(\boldsymbol{x}) \\ \vdots & & \vdots & & \vdots \\ p_{l1}(\boldsymbol{x}) & \cdots & p_{lj}(\boldsymbol{x}) & \cdots & p_{lk}(\boldsymbol{x}) \end{bmatrix} \tag{5}$$

In the matrix $DM(\boldsymbol{x})$, the i^{th} row of the matrix is the output of classifier L_i, the j^{th} column of the matrix are the support degrees from classifiers $C_1, C_2, \ldots, C_l$ for class ω_j.

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

(1) $g(\varnothing) = 1$, $g(C) = 1$;

(2) For $\forall C_i, C_j \subseteq C$, if $C_i \subset C_j$, then $g(C_i) \leq g(C_j)$.

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

$$g(C_i \cup C_j) = g(C_i) + g(C_j) + \lambda g(C_i)g(C_j) \tag{6}$$

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

The value of λ can be determined by solving the following equation.

$$\lambda + 1 = \prod_{i=1}^{l}(1 + \lambda g_i) \tag{7}$$

where $g_i = g(\{C_i\})$, it is usually determined by the following formula [38]:

$$g_i = \frac{p_i}{\sum_{j=1}^{l} p_j}\delta. \tag{8}$$

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

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

$$(C)\int h d\mu = \sum_{i=2}^{l+1} (h(C_{i-1}) - h(C_i))\, g(F_{i-1}) \tag{9}$$

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

Given a testing instance $\boldsymbol{x}$, when we use fuzzy integral to fuse l base classifiers $C_1, C_2, \cdots, C_l$ for classifying $\boldsymbol{x}$, the process includes three step: Firstly, compute decision matrix $DM(\boldsymbol{x})$. Secondly, sort $j^{th}(1 \leq j \leq k)$ column of $DM(\boldsymbol{x})$ in descending order and obtain $(p_{i_1j}, p_{i_2j}, \cdots, p_{i_lj})$. Finally, calculate the support degree $p_j(\boldsymbol{x})$ by the following formula.

$$p_j(\boldsymbol{x}) = \sum_{t=2}^{l+1} \left(p_{i_{t-1}j}(\boldsymbol{x}) \text{-} p_{i_tj}(\boldsymbol{x})\right) g(F_{t\text{-}1}) \tag{10}$$

The pseudo code of the proposed binary imbalanced data classification algorithm based on classifier fusion by fuzzy integral is given in algorithm 2.

Algorithm 2 The Binary Imbalanced Data Classification Algorithm Based on Classifier Fusion by Fuzzy Integral

Input: Imbalanced data set $S = S^+ \cup S^-$, testing 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^-, ..., S_l^-, where
l = |S^-| / |S_up^+|;
for (i = 1; i ≤ l; i = i + 1) do
    Construct balance training sets S_i = S_i^- ∪ S_up^+;
    Train base classifier C_i on S_i, and soft-maximize its
    outputs, obtain a probability distribution
    (p_i1(x), p_i2(x), ..., p_ik(x));
end
// The second stage: fuse the trained
   base classifiers by fuzzy
   integral;
Calculate fuzzy densities g_i (1 ≤ i ≤ l) by (8);
Calculate parameter λ by (7);
Calculate DM(x) by (6);
for (j = 1; j ≤ k; j = j + 1) do
    Sort j^th column of DM(x) in descending order and
    obtain (d_{i_1 j}, d_{i_2 j}, ..., d_{i_l j});
    Set g(F_1) = g_{i_1};
    for (t = 2; t ≤ l; t = t + 1) do
        Calculate g(F_t) = g_{i_t} + g(F_{t-1}) + λ g_{i_t} g(F_{t-1});
    end
    Calculate p_j(x) = Σ_{t=2}^{l+1} [d_{i_{t-1} j}(x) - d_{i_t j}(x)] g(F_{t-1});
end
Calculate p_{j*}(x) = argmax_{1≤j≤k} {p_j(x)};
Return j*.
```

IV. EXPERIMENTAL RESULTS AND ANALYSES

A. DATA SETS AND EXPERIMENTAL ENVIRONMENTS

In order to demonstrate the superiority of the proposed algorithm also denoted by D2GANDO for simplicity, we conducted extensive experiments on 8 data sets to compare D2GANDO with 7 state-of-the-art methods, including 4 SMOTE related approaches, and 3 GAN related approaches. The 4 SMOTE related approaches are SMOTE [6], B-SMOTE [7], ADASYN [8], and K-SMOTE [12], the 3 GAN related approaches are GAN [33], AC-GAN [34], and MFC-GAN [35]. The 8 data sets include 1 artificial data set, 4 KEEL data sets [39], 3 liver data sets [40]. The basic information of the 8 data sets is given in table 1. All experiments were carried out on the same hardware platform with Intel(R) Core(TM) i7-6600k CPU @ 3.10GHz, 16.0G 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 1. **The basic information of the 8 data sets.**

Data sets	♯Attribute	♯Instances	IR	Note
Gaussian	2	10000	100	1 artificial data set
Blocks0	10	5472	8.79	4 KEEL data sets
Segment0	19	2308	6.02	
Yeast1	8	1484	2.46	
Vowel0	13	988	9.98	
Liver1	5	12400	61	3 liver data sets
Liver2	5	14000	13	
Liver3	5	13000	25	

In table 1, IR $= \frac{|S^-|}{|S^+|}$. Gaussian is an artificial data set which is a two-dimensional data set 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 data set Gaussian is used for illustrating the feasibility of the proposed approach and visualizing the generated synthetic samples.

TABLE 2. **The mean vectors and covariance matrices of two Gaussian distributions.**

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

B. PERFORMANCE EVALUATION MEASURES

The used performance evaluation measures include MMD-score [41], Silhouette-score [42], F-measure [43], G-mean [43], and AUC-area [43]. The MMD is a statistics for measuring the mean squared difference of two sets of samples. Given two sets of samples $\mathbf{X} = \{\mathbf{x}_i\}$, $1 \leq i \leq n$ and $\mathbf{Y} = \{\mathbf{y}_i\}$, $1 \leq i \leq m$, the MMD of $\mathbf{X}$ and $\mathbf{Y}$ is defined by Eq.(11).

$$
\begin{aligned}
\mathrm{MMD} &= \left\| \frac{1}{n} \sum_{i=1}^{n} \phi(\mathbf{x}_i) - \frac{1}{m} \sum_{j=1}^{m} \phi(\mathbf{y}_i) \right\|^2 \\
&= \frac{1}{n^2} \sum_{i=1}^{n} \sum_{i'=1}^{n} \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_{i'}) \\
&\quad - \frac{2}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \phi(\mathbf{x}_i)^T \phi(\mathbf{y}_j) \\
&\quad + \frac{1}{m^2} \sum_{j=1}^{m} \sum_{j'=1}^{m} \phi(\mathbf{y}_j)^T \phi(\mathbf{y}_{j'}) \qquad (11)
\end{aligned}
$$

TABLE 3. Model parameter settings used for 8 data sets.

Data sets	♯HNodesG	♯HNodes	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

In Eq.(11), $\phi(\cdot)$ is a kernel mapping, using kernel trick, Eq.(11) can be written as Eq.(12).

$$\begin{aligned}\mathrm{MMD} = &\frac{1}{n^2}\sum_{i=1}^{n}\sum_{i'=1}^{n}k(\mathbf{x}_i, \mathbf{x}_{i'}) \\ &- \frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}k(\mathbf{x}_i, \mathbf{y}_j) \\ &+ \frac{1}{m^2}\sum_{j=1}^{m}\sum_{j'=1}^{m}k(\mathbf{y}_j, \mathbf{y}_{j'}) \end{aligned} \tag{12}$$

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.(13).

$$s(\mathbf{x}) = \frac{b(\mathbf{x}) - a(\mathbf{x})}{\max\{a(\mathbf{x}), b(\mathbf{x})\}} \tag{13}$$

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 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.(13), 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.

C. NETWORK ARCHITECTURE AND PARAMETER SETTINGS

In the modified D2GAN, the generator, the two discriminators and the classifier are all single hidden layer feedforward neural networks, and the two discriminators and the classifier have same architecture, i.e. they have same the number of hidden nodes denoted by ♯HNodes, and the number of hidden nodes of generator is denoted by ♯HNodesG. The dimension of noise z is uniformly set to 100. All parameters including 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 (SVM) [44], decision tree [45], and extreme learning machine (ELM) [46] as base classifiers for fusion to demonstrate that the superiority of the proposed method is less relevant to base classifier selection. For SVM, we set $C = 1.0$ and the kernel function is Gaussian, whose coefficient γ is the reciprocal of the dimension of the feature. For decision tree, we use the Gini index as a heuristic, and there is no limit to the depth of the decision tree. For ELM, the activation function is sigmoid, but for different data sets, the number of hidden layer nodes denoted also by ♯HNodes is different, which are given in Table 4.

D. COMPARISONS WITH 3 GAN RELATED METHODS

We use 5-fold cross validation to experimentally compare the proposed method D2GANDO with the 7 state-of-the-art approaches on 5 aspects: MMD-score, Silhouette-score, F-measure, G-means, and AUC-area, and visualize the generated synthetic samples on the artificial data set to demonstrate effectiveness and superiority of the proposed approach D2GANDO. The experimental results of MMD-score and Silhouette-score on the 8 data sets are given in Table 5 and Table 6, respectively.

From the experimental results listed in Table 5, the MMD-scores of the proposed method D2GANDO on the 7 data sets are greater than the ones of the 6 related approaches. In the data set Blocks0, the MMD-scores is greater than the one of D2GANDO. Overall, the positive class samples generated by D2GANDO have better diversities, this observation can be further confirmed by the visualization of the generated synthetic positive class samples on the artificial data set (see figure 3), the visualization was conducted by matplotlib package of Python. In the figure 3, the yellow "−" represents the negative class sample, the blue "+" represents the positive class sample, while red "+" represents the generated positive class sample. It can be seen from the figure 3 that the samples generated by the proposed method D2GANDO has better diversity than the 7 state-of-the-art approaches, including K-SMOTE which is an exception that K-SMOTE can not generate synthetic positive class samples on the artificial data set. This is due to its oversampling mechanism, K-SMOTE first use K-means to cluster the artificial data set, 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. Although MFC-GAN has good diversity, it has bad separability, i.e. the generated synthetic

TABLE 4. Hidden node settings of ELM networks used for 8 data sets.

Data sets	Gaussian	Blocks0	Segment0	Yeast1	Vowel0	Liver1	Liver2	Liver3
♯HNodes	15	50	55	25	35	50	50	50

TABLE 5. Experimental comparison of MMD-score on the 8 data sets.

Data sets	SMOTE	B-SMOTE	ADASYN	K-SMOTE	GAN	AC-GAN	MFC-GAN	D2GANDO
Gaussian	0.026	0.394	0.397	0.024	0.580	1.181	1.436	**1.646**
Blocks0	0.006	0.348	0.322	**2.441**	0.125	0.437	1.124	1.650
Segment0	0.012	1.179	0.624	0.406	0.408	1.371	1.495	**3.172**
Yeast1	0.011	0.040	0.023	0.285	0.067	0.653	0.805	**2.614**
Vowel0	0.042	0.425	0.206	0.038	0.398	1.139	1.217	**3.225**
Liver1	0.018	0.049	0.046	0.014	0.238	0.959	0.353	**3.512**
Liver2	0.004	0.056	0.049	0.003	0.203	0.391	0.033	**1.700**
Liver3	0.008	0.051	0.038	0.060	0.277	1.321	1.357	**2.631**

TABLE 6. Experimental comparison of Silhouette-score on the 8 data sets.

Data sets	SMOTE	B-SMOTE	ADASYN	K-SMOTE	GAN	AC-GAN	MFC-GAN	D2GANDO
Gaussian	0.449	0.394	0.380	0.438	0.441	0.382	0.425	**0.566**
Blocks0	0.171	0.098	0.091	**0.520**	0.216	0.231	0.407	0.438
Segment0	0.186	0.229	0.214	0.218	0.080	0.439	0.233	**0.665**
Yeast1	0.051	0.042	0.042	0.133	0.066	0.391	0.360	**0.678**
Vowel0	0.094	0.254	0.106	0.038	0.124	0.556	0.471	**0.637**
Liver1	0.081	0.049	0.057	-0.130	-0.150	0.280	0.670	**0.728**
Liver2	0.069	0.049	0.042	-0.090	-0.080	0.233	0.151	**0.406**
Liver3	0.074	0.053	0.049	-0.120	-0.110	0.207	0.187	**0.470**

positive class samples overlap with the original negative samples.

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. The good quality of the generated synthetic positive class samples can effectively expand the training field of positive class samples, and then effectively improve the performance of the proposed classification algorithm, this point can be confirmed by the experimental results on three classification performance metrics: F-measure, G-means, and AUC-area. The experimental results of F-measure, G-means, and AUC-area by support vector machine are given in Table 7-9. The experimental results of F-measure, G-means, and AUC-area by decision tree are given in Table 10-12. The experimental results of F-measure, G-means, and AUC-area by extreme learning machine are given in Table 13-15.

From the experimental results listed in Table 6, the Silhouette-scores of the proposed method D2GANDO on the 8 data sets are also greater than the ones of the 7 state-of-the-art approaches, which demonstrates that the oversampled positive class samples by D2GANDO have also better separability than the 7 state-of-the-art approaches, this conclusion can also be further confirmed by the visualization of the generated synthetic positive class samples on the artificial data set (see figure 3).

From the experimental results of F-measure, G-means and AUC-area by support vector machine given in table 7, 8 and 9, it can be found that (a) D2GANDO obtained 5 maximum values of F-measure, the other 3 maxima were obtained by SMOTE, K-SMOTE and MFC-GAN, respectively; (b) D2GANDO obtained 7 maximum values of G-means, another maximum was obtained by MFC-GAN; (c) D2GANDO obtained 6 maximum values of AUC-area, the other 2 maximum was obtained by SMOTE and MFC-GAN. From the experimental results of F-measure, G-means and AUC-area by decision tree and extreme learning machine given in Table 10-15, similar or even better results can be found. Overall, the performance of the proposed method D2GANDO outperforms the 7 approaches in terms of 5 aspects. We think that the reasons include the following three points:

(1) In contrast to D2GAN which insures to effectively diversify the estimated density in capturing multi-modes, D2GANDO adopts the MMD to ensure good diversity of the generated synthetic positive class samples, the good diversity can effectively expand the training field of the positive class samples.

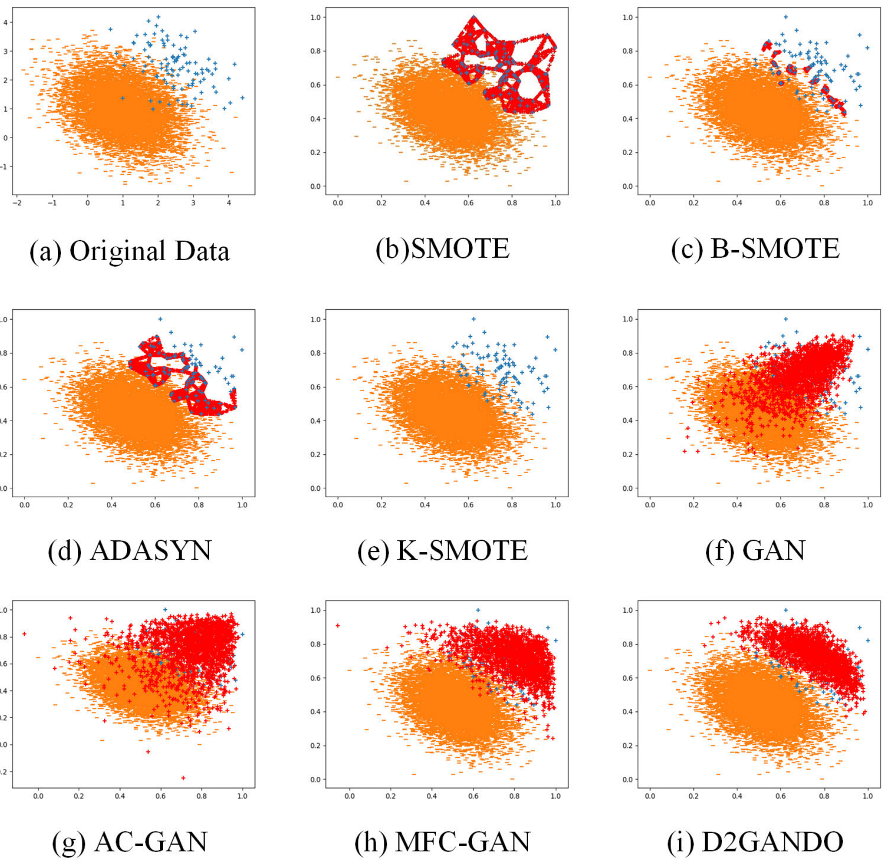

FIGURE 3. The visualization of the generated synthetic positive class samples of the artificial data set.

TABLE 7. Experimental comparison of F-measure by support vector machine on the 8 data sets.

Data sets	SMOTE	B-SMOTE	ADASYN	K-SMOTE	GAN	AC-GAN	MFC-GAN	D2GANDO
Gaussian	0.34	0.33	0.26	0.14	0.61	0.62	0.68	**0.73**
Blocks0	0.65	0.56	0.56	0.64	0.63	0.54	0.60	**0.74**
Segment0	0.88	0.63	0.66	0.90	0.89	0.86	**0.96**	0.93
Yeast1	0.46	0.45	0.40	0.50	0.49	0.39	0.40	**0.57**
Vowel0	0.93	0.92	0.93	0.93	0.93	0.88	0.90	**0.94**
Liver1	**0.85**	0.05	0.05	0.05	0.59	0.63	0.71	0.76
Liver2	0.86	0.87	0.89	0.90	0.92	0.91	0.87	**0.96**
Liver3	0.86	0.86	0.78	**0.91**	0.81	0.76	0.82	0.88

(2) Introducing a classifier to D2GAN can not only learn the distribution of samples, but also can learn a good classification boundary between positive class and negative class. In addition, the Silhouette-score can well measure separability between the generated synthetic positive class samples and negative, the combination of MMD-score and Silhouette-score can further effectively improve the quality of the generated synthetic positive class samples, and then finally effectively improve the performance of the proposed method D2GANDO.

(3) Because the base classifiers are trained on balanced training sets which all contain the same set of oversampling positive class samples, there exist intrinsic interactions among different base classifiers, the interactions may be positive correlated, in this case, the base classifiers enhance each other. The interactions also may be negative correlated, in this situation, 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.

TABLE 8. Experimental comparison of G-mean by support vector machine on the 8 data sets.

Data sets	SMOTE	B-SMOTE	ADASYN	K-SMOTE	GAN	AC-GAN	MFC-GAN	D2GANDO
Gaussian	0.45	0.46	0.34	0.54	0.69	0.69	0.70	**0.82**
Blocks0	0.72	0.63	0.63	0.73	0.70	0.63	0.69	**0.86**
Segment0	0.90	0.72	0.70	0.91	0.91	0.92	0.93	**0.94**
Yeast1	0.58	0.59	0.59	0.63	0.58	0.62	0.62	**0.66**
Vowel0	0.95	0.93	0.94	0.96	0.93	0.96	**0.97**	0.95
Liver1	0.85	0.67	0.16	0.00	0.70	0.84	0.75	**0.98**
Liver2	0.95	0.95	0.95	0.98	0.91	0.98	0.98	**0.99**
Liver3	0.93	0.95	0.89	0.97	0.91	0.90	0.95	**0.98**

TABLE 9. Experimental comparison of AUC-area by support vector machine on the 8 data sets.

Data sets	SMOTE	B-SMOTE	ADASYN	K-SMOTE	GAN	AC-GAN	MFC-GAN	D2GANDO
Gaussian	0.60	0.61	0.56	0.27	0.74	0.67	0.70	**0.83**
Blocks0	0.76	0.70	0.70	0.85	0.74	0.82	0.84	**0.86**
Segment0	0.90	0.76	0.75	0.91	0.92	0.95	**0.96**	0.93
Yeast1	0.62	0.59	0.62	0.65	0.62	0.59	0.60	**0.68**
Vowel0	0.95	0.98	0.98	0.97	0.92	0.94	0.93	**0.99**
Liver1	**0.90**	0.67	0.67	0.50	0.58	0.77	0.76	0.81
Liver2	0.91	0.92	0.93	0.92	0.90	0.93	0.89	**0.94**
Liver3	**0.93**	0.91	0.89	0.90	0.91	0.84	0.86	**0.93**

TABLE 10. Experimental comparison of F-measure by decision tree on the 8 data sets.

Data sets	SMOTE	B-SMOTE	ADASYN	K-SMOTE	GAN	AC-GAN	MFC-GAN	D2GANDO
Gaussian	0.37	0.28	0.26	0.17	0.64	0.65	0.66	**0.73**
Blocks0	0.58	0.46	0.52	0.61	0.64	0.57	0.63	**0.76**
Segment0	0.76	0.66	0.63	0.87	0.86	0.82	**0.94**	0.92
Yeast1	0.49	0.51	0.43	0.53	0.53	0.43	0.45	**0.58**
Vowel0	0.84	0.83	0.87	0.89	0.89	0.86	0.86	**0.93**
Liver1	**0.80**	0.23	0.12	0.09	0.64	0.66	0.72	0.74
Liver2	0.83	0.87	0.86	0.86	0.81	0.88	0.86	**0.96**
Liver3	0.88	0.85	0.80	0.88	0.83	0.85	0.81	**0.89**

TABLE 11. Experimental comparison of G-mean by decision tree on the 8 data sets.

Data sets	SMOTE	B-SMOTE	ADASYN	K-SMOTE	GAN	AC-GAN	MFC-GAN	D2GANDO
Gaussian	0.43	0.45	0.34	0.53	0.67	0.69	0.68	**0.82**
Blocks0	0.72	0.67	0.64	0.73	0.72	0.62	0.69	**0.87**
Segment0	0.91	0.71	0.71	0.89	0.92	0.93	0.92	**0.96**
Yeast1	0.57	0.63	0.58	0.63	0.6	0.63	0.63	**0.66**
Vowel0	0.93	0.91	0.91	0.95	0.93	**0.96**	**0.96**	**0.96**
Liver1	0.84	0.67	0.24	0.01	0.71	0.84	0.77	**0.98**
Liver2	0.93	0.94	0.95	0.95	0.93	0.98	0.93	**0.99**
Liver3	0.93	0.94	0.91	0.95	0.93	0.93	0.95	**0.99**

To further confirm the superiorities of the proposed algorithm to the 7 state-of-the-art methods, we statistically analyzed the experimental results of F-measures by three base classifiers with paired T-test in confidence level 0.05 [47]. For the limitation of pages, we do not provide the statistical analysis on G-mean and AUC-area by three

TABLE 12. Experimental comparison of AUC-area by decision tree on the 8 data sets.

Data sets	SMOTE	B-SMOTE	ADASYN	K-SMOTE	GAN	AC-GAN	MFC-GAN	D2GANDO
Gaussian	0.6	0.59	0.56	0.27	0.71	0.68	0.71	**0.83**
Blocks0	0.75	0.64	0.7	0.83	0.71	0.81	0.83	**0.86**
Segment0	0.91	0.71	0.73	0.9	0.86	**0.95**	**0.95**	0.93
Yeast1	0.61	0.58	0.62	0.66	0.58	0.59	0.61	**0.69**
Vowel0	0.94	0.94	0.97	0.96	0.91	0.93	0.91	**0.99**
Liver1	**0.88**	0.63	0.69	0.53	0.57	0.77	0.75	0.82
Liver2	0.88	0.91	0.93	0.87	0.91	0.93	0.88	**0.94**
Liver3	0.88	0.91	0.92	0.87	0.91	0.88	0.86	**0.94**

TABLE 13. Experimental comparison of F-measure by extreme learning machine on the 8 data sets.

Data sets	SMOTE	B-SMOTE	ADASYN	K-SMOTE	GAN	AC-GAN	MFC-GAN	D2GANDO
Gaussian	0.34	0.35	0.26	0.16	0.62	0.62	0.72	**0.74**
Blocks0	0.66	0.57	0.55	0.64	0.72	0.53	0.62	**0.74**
Segment0	0.88	0.64	0.68	0.91	0.88	0.86	**0.96**	0.95
Yeast1	0.47	0.46	0.43	0.53	0.48	0.42	0.45	**0.58**
Vowel0	0.93	0.91	0.91	0.92	0.93	0.87	0.92	**0.95**
Liver1	**0.86**	0.14	0.07	0.05	0.59	0.67	0.73	0.77
Liver2	0.86	0.86	0.88	0.91	0.92	0.91	0.91	**0.96**
Liver3	0.86	0.86	0.86	0.88	0.82	0.75	0.86	**0.89**

TABLE 14. Experimental comparison of G-mean by extreme learning machine on the 8 data sets.

Data sets	SMOTE	B-SMOTE	ADASYN	K-SMOTE	GAN	AC-GAN	MFC-GAN	D2GANDO
Gaussian	0.46	0.45	0.36	0.55	0.73	0.71	0.71	**0.85**
Blocks0	0.73	0.63	0.64	0.74	0.72	0.62	0.69	**0.88**
Segment0	0.91	0.72	0.73	0.92	0.91	0.91	0.94	**0.96**
Yeast1	0.58	0.58	0.64	0.64	0.63	0.63	0.65	**0.67**
Vowel0	0.95	0.92	0.93	0.89	0.92	0.96	**0.99**	0.98
Liver1	0.86	0.67	0.18	0.02	0.72	0.88	0.75	**0.98**
Liver2	0.95	0.95	0.94	0.95	0.94	0.96	**0.99**	**0.99**
Liver3	0.94	0.96	0.91	0.94	0.94	0.96	0.96	**0.99**

TABLE 15. Experimental comparison of AUC-area by extreme learning machine on the 8 data sets.

Data sets	SMOTE	B-SMOTE	ADASYN	K-SMOTE	GAN	AC-GAN	MFC-GAN	D2GANDO
Gaussian	0.62	0.63	0.57	0.31	0.73	0.67	0.68	**0.86**
Blocks0	0.76	0.72	0.71	0.81	0.74	0.83	0.81	**0.86**
Segment0	0.92	0.76	0.78	0.88	0.91	**0.96**	0.95	0.94
Yeast1	0.64	0.63	0.64	0.66	0.65	0.62	0.72	**0.79**
Vowel0	0.96	0.95	0.95	0.96	0.93	0.96	0.96	**0.99**
Liver1	0.87	0.67	0.68	0.52	0.60	0.77	0.79	**0.91**
Liver2	0.88	0.91	0.94	0.91	0.91	0.92	0.89	**0.94**
Liver3	0.89	0.91	0.89	0.90	0.91	0.83	0.88	**0.94**

base classifiers. Specifically, for each data set and for each method, we run the 5-fold cross-validation 5 times and obtain eight 25-dimensional statistics denoted by X_1, X_2, X_3, X_4, X_5, X_6, X_7 and X_8 corresponding to SMOTE, B-SMOTE, ADASYN, K-SMOTE, GAN, AC-GAN, MFC-GAN and D2GANDO respectively. Next the paired T-test is applied

TABLE 16. Statistical analysis on F-measure by support vector machine.

Data sets	p-value1	p-value2	p-value3	p-value4	p-value5	p-value6	p-value7
Gaussian	5.92E-06	1.72E-05	5.04E-05	3.91E-05	1.99E-06	5.72E-08	4.21E-06
Blocks0	8.18E-06	5.84E-08	4.91E-07	1.44E-07	3.65E-08	1.97E-06	3.30E-06
Segment0	1.73E-07	4.76E-06	1.23E-06	5.79E-07	7.07E-08	6.30E-08	6.31E-01
Yeast1	5.21E-07	7.79E-09	4.28E-06	1.11E-06	1.59E-05	1.71E-06	4.18E-07
Vowel0	1.05E-03	2.09E-05	1.52E-03	3.53E-02	6.15E-03	1.07E-04	1.16E-05
Liver1	3.45E-01	1.92E-04	8.08E-05	1.25E-04	1.69E-05	2.90E-06	1.65E-05
Liver2	4.47E-11	1.07E-08	2.43E-11	4.18E-08	3.06E-05	1.51E-06	2.70E-05
Liver3	3.34E-05	7.33E-04	9.96E-09	9.81E-01	7.50E-05	1.36E-06	6.00E-08

TABLE 17. Statistical analysis on F-measure by decision tree.

Data sets	p-value1	p-value2	p-value3	p-value4	p-value5	p-value6	p-value7
Gaussian	3.24E-14	9.01E-15	8.78E-15	7.63E-08	5.86E-07	7.65E-07	2.81E-06
Blocks0	1.07E-09	3.86E-13	1.29E-11	1.03E-08	7.64E-08	4.83E-08	1.32E-08
Segment0	8.87E-10	3.74E-09	2.54E-08	1.45E-06	2.20E-07	5.32E-11	5.89E-03
Yeast1	8.81E-11	4.16E-08	6.92E-13	4.91E-07	9.38E-07	1.62E-15	2.68E-10
Vowel0	5.14E-10	5.92E-13	4.77E-08	4.26E-06	2.74E-06	2.76E-09	3.31E-07
Liver1	1.49E-01	6.60E-07	2.99E-06	5.00E-06	2.28E-12	3.28E-09	1.57E-04
Liver2	1.09E-13	3.56E-09	8.96E-14	1.85E-14	5.53E-11	1.49E-10	1.58E-12
Liver3	1.76E-12	1.06E-12	4.74E-12	8.30E-10	3.56E-17	7.69E-11	5.50E-15

TABLE 18. Statistical analysis on F-measure by extreme learning machine.

Data sets	p-value1	p-value2	p-value3	p-value4	p-value5	p-value6	p-value7
Gaussian	5.09E-05	3.43E-06	1.51E-05	1.26E-04	7.01E-08	5.44E-07	6.00E-05
Blocks0	2.23E-10	5.56E-07	3.04E-06	5.29E-10	9.38E-03	1.15E-06	7.99E-10
Segment0	5.18E-08	9.37E-05	1.06E-05	3.83E-05	3.04E-07	8.33E-08	1.48E-01
Yeast1	2.53E-05	6.43E-08	6.46E-06	1.93E-09	7.80E-10	1.67E-07	9.16E-09
Vowel0	9.74E-04	1.59E-04	2.50E-06	1.60E-04	1.18E-05	4.18E-09	1.89E-04
Liver1	4.10E-01	3.49E-05	9.42E-05	6.50E-05	3.49E-06	1.94E-06	7.55E-05
Liver2	5.56E-10	1.42E-07	1.16E-07	1.07E-07	1.29E-07	2.51E-07	1.48E-06
Liver3	9.70E-05	8.55E-04	8.64E-05	8.29E-02	3.42E-07	1.41E-06	3.91E-05

to the experimental results by calling the Python library function ttest_rel(·, ·). The results of the statistical analysis on F-measures by three base classifiers are listed in table 16, 17 and 18 respectively. From the p-values listed in the three tables, we can undoubtedly confirm that D2GANDO statistically outperforms the 7 state-of-the-art methods.

V. CONCLUSION

Based on a modified D2GAN model and classifier fusion mechanism, an approach for classifying binary imbalanced data was proposed in this article. The proposed method contains a diversity oversampling method and an ensemble classification approach for classification of binary imbalanced data. The oversampling method is based on the modified D2GAN model by introducing a classifier into the D2GAN model. The ensemble classification approach is based on fuzzy integral, because that the base classifiers are trained on balanced training sets containing same positive class set, there are intrinsic interactions among the base classifiers, fuzzy integral can well the interactions which can effectively enhance the classification performance. The proposed method has four advantages: (1) it can generate synthetic positive class samples with good diversity and good separability. (2) the modified D2GAN model can effectively avoid mode collapse. (3) it has good classification generalization ability due to diverse oversampling and controllable separability. (4) it is effective not only for data sets with medium imbalanced ratio, but also for data sets with very high imbalanced ratio. The promising future works of this study include (1) The extension of D2GANDO to multi-class imbalacned data classification; (2) The scalability of D2GANDO in imbalanced big data scenario.

REFERENCES

[1] N. Japkowicz, "The class imbalance problem: Significance and strategies," in *Proc. Int. Conf. Artif. Intell.*, 2000, pp. 111–117.

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

[3] W. Zhang, X. Li, X.-D. Jia, H. Ma, Z. Luo, and X. Li, "Machinery fault diagnosis with imbalanced data using deep generative adversarial networks," *Measurement*, vol. 152, Feb. 2020, Art. no. 107377.

[4] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from Imbalanced Data Sets*. Cham, Switzerland: Springer, 2018.

[5] D. Wang and W. Ding, "A hierarchical pattern learning framework for forecasting extreme weather events," in *Proc. IEEE Int. Conf. Data Mining*, Atlantic City, NJ, USA, Nov. 2015, pp. 1021–1026.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[7] H. Han, W. Y. Wang, and B. H. Mao, "Borderline-SMOTE: A new oversampling method in imbalanced data sets learning," in *Proc. Int. Conf. Adv. Intell. Comput.* Berlin, Germany: Springer-Verlag, 2005, pp. 878–887.

[8] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw., IEEE World Congr. Comput. Intell.*, Hong Kong, Jun. 2008, pp. 1322–1328.

[9] S. Barua, M. M. Islam, X. Yao, and K. Murase, "MWMOTE-majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 2, pp. 405–425, Feb. 2014.

[10] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem," in *Proc. 13th Pacific-Asia Conf. Adv. Knowl. Discovery Data Mining*, 2009, pp. 475–482.

[11] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "DBSMOTE: Density based synthetic minority over-sampling technique," *Appl. Intell.*, vol. 36, no. 3, pp. 664–684, 2012.

[12] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE," *Inf. Sci.*, vol. 465, pp. 1–20, Oct. 2018.

[13] G. Douzas and F. Bacao, "Geometric SMOTE a geometrically enhanced drop-in replacement for SMOTE," *Inf. Sci.*, vol. 501, pp. 118–135, Oct. 2019.

[14] J. Mathew, C. K. Pang, M. Luo, and W. H. Leong, "Classification of imbalanced data by oversampling in kernel space of support vector machines," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4065–4076, Sep. 2018.

[15] B. S. Raghuwanshi and S. Shukla, "SMOTE based class-specific extreme learning machine for imbalanced learning," *Knowl.-Based Syst.*, vol. 187, Jan. 2020, Art. no. 104814.

[16] S. Maldonado, J. López, and C. Vairetti, "An alternative SMOTE oversampling strategy for high-dimensional datasets," *Appl. Soft Comput.*, vol. 76, pp. 380–389, Mar. 2019.

[17] X. Tao, Q. Li, C. Ren, W. Guo, C. Li, Q. He, R. Liu, and J. Zou, "Real-value negative selection over-sampling for imbalanced data set learning," *Expert Syst. Appl.*, vol. 129, pp. 118–134, Sep. 2019.

[18] X. Tao, Q. Li, W. Guo, C. Ren, Q. He, R. Liu, and J. Zou, "Adaptive weighted over-sampling for imbalanced datasets based on density peaks clustering with heuristic filtering," *Inf. Sci.*, vol. 519, pp. 43–73, May 2020.

[19] P. Shamsolmoali, M. Zareapoor, L. Shen, A. H. Sadka, and J. Yang, "Imbalanced data learning by minority class augmentation using capsule adversarial networks," *Neurocomputing*, Jul. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0925231220312091?via%3Dihub, doi: 10.1016/j.neucom.2020.01.119.

[20] G. Kovács, "An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets," *Appl. Soft Comput.*, vol. 83, Oct. 2019, Art. no. 105662.

[21] D. Elreedy and A. F. Atiya, "A comprehensive analysis of synthetic minority oversampling technique (SMOTE) for handling class imbalance," *Inf. Sci.*, vol. 505, pp. 32–64, Dec. 2019.

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

[23] N. N. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting," in *Proc. 7th Eur. Conf. Princ. Pract. Knowl. Discovery Databases*, 2003, pp. 107–119.

[24] S. Wang and X. Yao, "Diversity analysis on imbalanced data sets by using ensemble models," in *Proc. IEEE Symp. Comput. Intell. Data Mining (CIDM)*, Mar. 2009, pp. 324–331.

[25] Z. Yongqing, Z. Min, Z. Danling, M. Gang, and M. Daichuan, "Improved SMOTEBagging and its application in imbalanced data classification," in *Proc. IEEE Conf. Anthol.*, Jan. 2013, pp. 1–6.

[26] J. Sun, H. Li, H. Fujita, B. Fu, and W. Ai, "Class-imbalanced dynamic financial distress prediction based on AdaBoost-SVM ensemble combined with SMOTE and time weighting," *Inf. Fusion*, vol. 54, pp. 128–144, Feb. 2020.

[27] J. Zhai, S. Zhang, and C. Wang, "The classification of imbalanced large data sets based on MapReduce and ensemble of ELM classifiers," *Int. J. Mach. Learn. Cybern.*, vol. 8, no. 3, pp. 1009–1017, Jun. 2017.

[28] Z. Chen, T. Lin, X. Xia, H. Xu, and S. Ding, "A synthetic neighborhood generation based ensemble learning for the imbalanced data classification," *Appl. Intell.*, vol. 48, no. 8, pp. 2441–2457, 2018.

[29] X. Tao, Q. Li, W. Guo, C. Ren, C. Li, R. Liu, and J. Zou, "Self-adaptive cost weights-based support vector machine cost-sensitive ensemble for imbalanced data classification," *Inf. Sci.*, vol. 487, pp. 31–56, Jun. 2019.

[30] M. L. Wong, K. Seng, and P. K. Wong, "Cost-sensitive ensemble of stacked denoising autoencoders for class imbalance problems in business domain," *Expert Syst. Appl.*, vol. 141, Mar. 2020, Art. no. 112918.

[31] Z. Zhu, Z. Wang, D. Li, Y. Zhu, and W. Du, "Geometric structural ensemble learning for imbalanced problems," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1617–1629, Apr. 2020.

[32] K. Yang, Z. Yu, X. Wen, W. Cao, C. L. P. Chen, H.-S. Wong, and J. You, "Hybrid classifier ensemble for imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 4, pp. 1387–1400, Apr. 2020.

[33] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 1, 2014, pp. 2672–2680.

[34] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 2642–2651.

[35] A. Ali-Gombe and E. Elyan, "MFC-GAN: Class-imbalanced dataset classification using multiple fake class generative adversarial network," *Neurocomputing*, vol. 361, pp. 212–221, Oct. 2019.

[36] M. Zheng, T. Li, R. Zhu, Y. Tang, M. Tang, L. Lin, and Z. Ma, "Conditional wasserstein generative adversarial network-gradient penalty-based approach to alleviating imbalanced data classification," *Inf. Sci.*, vol. 512, pp. 1009–1023, Feb. 2020.

[37] T. Nguyen, T. Le, H. Vu, and D. Phung, "Dual discriminator generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2670–2680.

[38] A. C. B. Abdallah, H. Frigui, and P. Gader, "Adaptive local fusion with fuzzy integrals," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 5, pp. 849–864, Oct. 2012.

[39] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, and S. García, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *J. Multiple-Valued Logic Soft Comput.*, vol. 17, nos. 2–3, pp. 255–287, 2011.

[40] J. Zhai, S. Zhang, M. Zhang, and X. Liu, "Fuzzy integral-based ELM ensemble for imbalanced big data classification," *Soft Comput.*, vol. 22, no. 11, pp. 3519–3531, Jun. 2018.

[41] A. Gretton, K. M. Borgwardt, and M. Rasch, "A kernel method for the two-sample problem," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2006, pp. 1672–1679.

[42] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.

[43] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[44] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer, 1995.

[45] J. R. Quinlan, *C4.5: Program for Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann, 1993.

[46] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006.

[47] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.

JUNHAI ZHAI (Member, IEEE) received the B.S. degree in mathematics and the M.S. degree in computing mathematics from Lanzhou University, Lanzhou, China, in 1988 and 2000, respectively, and the Ph.D. degree in optical engineering from Hebei University, Baoding, China, in 2010. He is currently a Professor with the College of Mathematics and Information Science, Hebei University. His main research interests include machine learning, deep learning, and big data processing.

JIAXING QI received the B.S. degree in software engineering from the Industrial and Commercial College, Hebei University, Baoding, China, in 2017. He is currently pursuing the M.S. degree with the College of Mathematics and Information Science, Hebei University. His main research interests include machine learning and deep learning.

SUFANG ZHANG received the B.S. degree in mathematics from Lanzhou University, Lanzhou, China, in 1988, and the M.S. degree in mathematics and applied mathematics from Hebei University, Baoding, China, in 2006. She is currently a Faculty Member with the Hebei Branch of China Meteorological Administration Training Center, China Meteorological Administration, China. Her research interest includes machine learning.