IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 December 2024, accepted 27 January 2025, date of publication 5 February 2025, date of current version 25 February 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3538867

RESEARCH ARTICLE

An Improved Triangulation Oversampling Method for Processing Unbalanced Data

JINGJING LIU[1,2], YEFENG LIU[1,3], YANWEI MA[1,4], AND QICHUN ZHANG[5], (Senior Member, IEEE)

[1]Liaoning Key Laboratory of Information Physics Fusion and Intelligent Manufacturing for CNC Machine, Shenyang Institute of Technology, Fushun 113122, China
[2]Department of Basic Courses, Shenyang Institute of Technology, Fushun 113122, China
[3]School of Automation and Electrical Engineering, Shenyang Ligong University, Shenyang 110180, China
[4]School of Mechanical Engineering and Automation, Shenyang Institute of Technology, Fushun 113122, China
[5]School of Creative and Digital Industries, Buckinghamshire New University, HP11 2JZ High Wycombe, U.K.

Corresponding author: Yefeng Liu (liuyefeng@situ.edu.cn)

This work was supported in part by the Science and Technology Plan Project of Liaoning Province under Grant 2023JH2/101700066, in part by the National Science Foundation of China under Grant 62073226, in part by the State Key Laboratory of Synthetical Automation for Process Industries Program under Grant 2023-kfkt-03 and Grant SAPI-2024-KFKT-05, and in part by the Basic Scientific Research Project of Liaoning Province Department of Education under Grant JYTMS20231179.

ABSTRACT In classification tasks, the algorithms heavily depend on big data, yet data category imbalance hinders the model's ability to adequately learn from scarce class samples, impacting its overall learning capacity. This paper addresses the challenge of data imbalance in data-driven classification problems, which introduce an Improved Finite Elements-Synthetic Minority Oversampling Technique (IFE-SMOTE) for data balancing. This method triangulates the sample space leveraging FE-SMOTE, designates three strategic points within each split element based on defined rules, and synthetically generates samples within the linear vicinity of these points. The mathematical expectation of generated samples aligns with the triangle center of the original minority samples, while their variance closely mirrors that of the triangle center, ensuring statistical consistency with the original data. The corresponding theorem is provided and its validity is proved. Numerical experiments confirm the effectiveness of the proposed IFE-SMOTE method. The IFE-SMOTE algorithm classified eight datasets and compared them with other oversampling algorithms using G-mean, F-measure, and AUC. IFE-SMOTE's average scores were 0.9273, 0.9754, and 0.9309, respectively, outperforming FE-SMOTE by 0.0268, 0.0059, and 0.0253, and other algorithms' averages by 0.0493, 0.0169, and 0.0402.

INDEX TERMS Oversampling, SMOTE, triangulation, unbalanced data.

I. INTRODUCTION

In the age of artificial intelligence, leveraging data-driven approaches to tackle intricate industrial problems has emerged as a pivotal method. When confronted with abnormal or malfunctioning industrial processes, data gathering is frequently constrained by environmental factors, leading to imbalances across data types. Addressing data imbalance is a pressing research topic. Solutions to this challenge encompass data-level, algorithm-level, and hybrid methodologies [1]. Given the direct correlation between data quality and model performance, enhancing data quality often surpasses sole algorithm optimization in effectiveness. This paper specifically delves into data-level research endeavors.

Data-level techniques encompass oversampling and undersampling strategies. Notable undersampling methods are Random Undersampling [2], Condensed Nearest Neighbor (CNN) [3], Willson's Edited Nearest Neighbor (ENN) [4], and One-Sided Selection (OSS) [5]. In the oversampling method, Random Oversampling [6] is the simplest oversampling method, achieves balance by duplicating underrepresented classes. However, this may induce overfitting due to the absence of novel information. In order to generate new information, the SMOTE(synthetic minority

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

oversampling technique) [7] proposed by Chawla et al. generates new samples through interpolation, which provides a new direction for solving the imbalance problem. This technique enhances classifier generalization on test sets, proving effective as a preprocessing step for unbalanced data and finding widespread application across domains.

However, SMOTE faces challenges with high-dimensional data and outliers. Variations on the SMOTE method followed: SMOTEN [7], Borderline-SMOTE [8], ADASYN(Adaptive synthetic sampling approach) [9], Kmeans-SMOTE [10], CP-SMOTE(Center point SMOTE) [11], IO-SMOTE(Inner and outer SMOTE) [11], KMeans-SMOTE-ENN [12], Radius-SMOTE [13], BO-SMOTE(Bayesian-Optimization-Based-SMOTE) [14]. SMOTEN utilizes the Value Difference Metric (VDM) [15] to quantify distances between samples, leveraging classification data and distance metrics for oversampling. Borderline-SMOTE targets samples near classification boundaries for new sample generation. ADASYN dynamically adjusts sample weights based on data distribution, automatically determining the number of new samples to rectify skewness. Kmeans-SMOTE integrates K-means clustering with SMOTE, oversampling in clusters rich in majority class samples. CP-SMOTE crafts new samples by identifying central points and linearly combining them with a subset of samples. IO-SMOTE segregates minority samples into interior and exterior groups, prioritizing interior samples for new sample creation. K-Means-SMOTE-ENN combines resampling techniques to manage noisy unbalanced datasets. Radius-SMOTE emphasizes safe radius distances for composite data creation, minimizing overlap with opposing classes. Bo-SMOTE frames classifier prediction as a black-box optimization, iteratively generating samples via Bayesian Optimization (BO). However, the new data produced by these oversampling methods remains stochastic and may significantly deviate from the original data distribution.

In order to study the distribution of raw data, researchers have made some efforts, especially in the problem of dealing with high-dimensional unbalanced data. The problems of sparse distribution of high-dimensional unbalanced data, feature redundancy or feature uncorrelation affect the traditional learning algorithm to identify a few class samples, and feature selection or dimensionality reduction is a common method, such as LASSO-SMOTE [16], BRFE-PBKS-SVM [17], [18]. A combination of the stochastic forest method and SMOTE [19]. In addition, Polynom-fit-SMOTE [20] and DeepSMOTEA [21] explored the distribution of the data from a higher dimension. However, the former is not effective in dealing with outliers, while the latter is more computative. In recent years, researchers have explored the generation of new samples from the perspective of computational geometry. G-SMOTE [22] extended the SMOTE data range by generating artificial samples in the geometric area around each selected minority of class samples. DTO-SMOTE [23] is a data balance processing method based on the combination of Delaunay subdivision and SMOTE. It reduces the dimensionality of the original data and then generates samples based on simplex. FE-SMOTE [24] is also based on the finite element method, which does not reduce the dimensionality of the data, and its most important feature is to make the generated sample more consistent with the original distribution of the minority samples, that is, the expectation of the new sample is equal to the expectation of the center of the constructed basic unit. Optimization (BO) iteration generation. The new data generated by these oversampling methods are still random and have a large deviation from the distribution of the original data.

The innovation of this paper lies in the following: To enhance the classification performance of the balanced treated sample, improvements have been made based on the FE-SMOTE algorithm. These improvements ensure that the distribution of the newly generated samples more closely resembles the original data, possessing similar features to the original samples, and thus improving the training and learning efficiency of the subsequent model. To enhance the similarity between the distribution of newly generated samples and the original data, this paper endeavors to align the expectation and variance of the new samples with those of the constructed triangular unit centers, leveraging the FE-SMOTE algorithm. Consequently, the research focuses on exploring and refining two key aspects.

1) Instead of generating new samples in the whole triangle region as FESMOTE did, a new generation method was defined after triangulating the sample space: generating samples in the neighborhood of the specified point on the middle line of the triangle unit.

2) The new sample's expectations and variances exhibit a closer approximation to those of the original minority sample. The positioning of designated points is parameterized, enabling the optimal parameter values for different midlines to be derived through a straightforward optimization process. This demonstrates that the new sample's statistical properties align more closely with those of the original minority sample.

This method not only explores the generation space of the new sample, but also limits its generation range regularly, retaining and extending the advantage of FE-SMOTE, so that the variance of the new sample is also approximately equal to the variance of the center of the constructed basic unit.

In the context of data-based classification applications, IFE-SMOTE is capable of addressing the issue of class data imbalance during the data preprocessing phase by augmenting the dataset with additional samples. Consequently, it enhances the learning efficiency of the model during the training stage, ultimately leading to improved classification performance.

The structure of this paper is as follows. The second part introduces the basic knowledge of SMOTE and FE-SMOTE algorithm. The third part gives the method of improving FE-SMOTE put forward in this paper. The fourth part gives

the numerical experimental results. The fifth part summarizes the full text.

II. PRELIMINARIES

A. SMOTE ALGORITHM

The basic idea of SMOTE is to analyze a few class samples and generate new composite samples based on those samples to increase the number of minority class samples and thus achieve a balance in the dataset. This method generates new samples by interpolating a few class samples. Specifically, SMOTE is implemented through the following steps.

1) Calculate K nearest neighbors. For each minority class sample, its K-nearest neighbor is calculated by calculating the Euclidean distance.

2) Choose the nearest neighbors at random. One sample is randomly selected from the K-nearest neighbors of each minority class sample.

3) Generate synthetic samples. Between the selected minority class sample and its randomly selected neighbors, a new composite sample is generated based on linear interpolation, which is added to the minority class as a new instance.

B. FE-SMOTE ALGORITHM

The FE-SMOTE method constructs a simplex within the original space for each positive sample and its neighboring points. This approach ensures that the generated new samples occupy both the interior and boundary regions of the triangulation, with their mathematical expectation matching that of the triangle's basic unit center. This alignment fosters a closer fit to the local density distribution of the positive samples. The subsequent details are sourced from literature [24].

1) BUILDING BASE UNIT

Let the unbalanced binary data set be $H = \{(\boldsymbol{X}_i, t_i)|X_i \in R^d, t_i \in \{1, -1\}, i = 1, \ldots, n\}$,$t_i$ is the label corresponding to the sample $\boldsymbol{X}_i$. Let $H^+ = \{(\boldsymbol{x}_i\}_{i=1}^{n_1}$ and $H^- = \{(\boldsymbol{x}_i\}_{i=1}^{n_2}$ are sample sets of minority and majority classes respectively, where $n_1 = |H^+|$, $n_2 = |H^-|$. A basic unit is constructed for each positive sample $\boldsymbol{x}_i$, according to the nearest neighbor algorithm, it is composed of $\boldsymbol{x}_i$ and d neighbors $Nerb_i = \{\boldsymbol{x}_i^l\}_{l=1}^d$ closest to $\boldsymbol{x}_i$, where x_i^l is the l sample closest to $\boldsymbol{x}_i$.

2) TRIANGULATION OF $(D+1)$-SIDED POLYHEDRON

For each positive sample $\boldsymbol{x}_i$, triangulate its $(d+1)$-sided polyhedron with its vertex V and center $\boldsymbol{v}_c$, as shown in formulas (1) and (2). If the center of a polyhedron and two vertices of its d+1 edges are arbitrarily combined to form C_{d+1}^2 triangles, each triangle is a triangulation.

$$V = Nerb_i \cup \{\boldsymbol{x}_i\} = \{\boldsymbol{v}_m\}_{m=1}^{d+1} \tag{1}$$

$$\boldsymbol{v}_c = (\sum_{m=1}^{d+1} \boldsymbol{v}_m)/(d+1) \tag{2}$$

3) DETERMINES THE NUMBER OF SAMPLES GENERATED ON THE TRIANGULAR UNIT

Let D_j represent the sum of the distance between the two vertices of the j-th triangle and the central vertex, and use it to measure the size of the triangle element, then

$$D_j = \frac{1}{2}(\|\boldsymbol{v}_1^j - \boldsymbol{v}_c^j\|_2 + \|\boldsymbol{v}_2^j - \boldsymbol{v}_c^j\|_2) \tag{3}$$

where v_1^j,v_2^j and v_c^j are the three vertices of the j-th triangular element. Calculate the number of new samples num_j according to formula (4)-(6).

$$O = \{e^{1-D_j}\}_{j=1}^{C_{d+1}^2 * n_1} \tag{4}$$

$$\bar{O} = \{\frac{e^{1-D_j}}{\sum_j e^{1-D_j}}\}|_{j=1}^{C_{d+1}^2 * n_1} = \{o_j\}|_{j=1}^{C_{d+1}^2 * n_1} \tag{5}$$

$$num_j = \lceil (n_2 - n_1) * o_j \rceil \tag{6}$$

4) SYNTHESIZE NEW SAMPLE

In the j-th triangular element after splitting, a new sample is generated according to formulas (7)-(8).

$$\boldsymbol{v}_{mid} = r_1 \boldsymbol{v}_1^j + (1 - r_2)\boldsymbol{v}_2^j \tag{7}$$

$$\begin{aligned} \boldsymbol{x}' &= r_0 \boldsymbol{v}_c^j + (1 - r_0)\boldsymbol{v}_{mid} \\ &= r_0 \boldsymbol{v}_c^j + (1 - r_0)(r_1 \boldsymbol{v}_1^j + (1 - r_2)\boldsymbol{v}_2^j) \end{aligned} \tag{8}$$

where r_0 and r_1 follow uniform distribution on [0, 1].

The conclusion has been given in literature [24]: The expectation of samples generated by (8) inside a basic unit constructed by any $d+1$ samples in the d-dimensional space is equal to the expectation of the center of the basic unit.

III. THE IMPROVED FE-SMOTE ALGORITHM

In order to make the generated sample have more similar features to the original sample, the numerical feature of variance is considered, that is, the variance and expectation of the generated sample are close to the variance and expectation of the center of the trigonometric basic unit. Step 4) in improving the FE-SMOTE method is as follows.

In the j-th triangular element after splitting, a new sample x' is generated according to formulas (9)-(11).

$$\boldsymbol{x}' = (1 - \delta)\boldsymbol{v}_c + \delta(\frac{\boldsymbol{v}_1 + \boldsymbol{v}_2}{2}), \delta \in U^+(0) \tag{9}$$

$$\boldsymbol{x}' = (1 - \delta)\boldsymbol{v}_1 + \delta(\frac{\boldsymbol{v}_c + \boldsymbol{v}_2}{2}), \delta \in U(0.6638) \tag{10}$$

$$\boldsymbol{x}' = (1 - \delta)\boldsymbol{v}_2 + \delta(\frac{\boldsymbol{v}_1 + \boldsymbol{v}_c}{2}), \delta \in U(0.6638) \tag{11}$$

where $\boldsymbol{v}_c$ is the center of gravity of the generalized triangle, $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, $\boldsymbol{v}_c$ is the three vertices of the triangular element after subdivision, and U represents the neighborhood.

As shown in Figure 1 and Figure 2, the vertices of a triangle or tetrahedron are sample points, which form a basic unit of a triangle, and the red points are newly synthesized points. If the sample is newly synthesized according to this method, then proposition 1 is true.

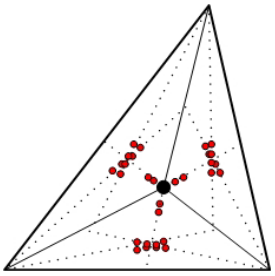

FIGURE 1. Synthesis of new samples in two-dimensional space.

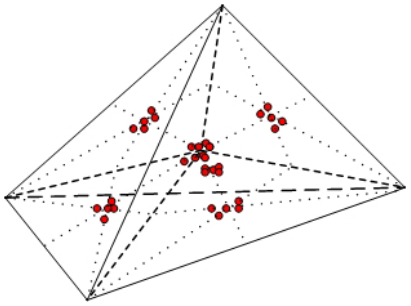

FIGURE 2. Synthesis of new samples in 3D space.

Proposition 1: A generalized triangle can be formed by $d+1$ samples in the d-dimensional space, and a new sample $\boldsymbol{x}^*$ is synthesized within each triangle element according to formula (9) -(11) after triangulation (assuming that the parameter δ is constant), the mathematical expectation of the new sample is equal to the expectation of the generalized triangle center, and the variance of the new sample is approximately equal to the variance of the generalized triangle center, $E(\boldsymbol{x}^*) = E(\boldsymbol{v}_{center})$, $D(\boldsymbol{x}^*) \approx D(\boldsymbol{v}_{center})$.

Proof: Represented by $d = 3$, the same can be proved in other cases. Let $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, $\boldsymbol{v}_3$ be the vertex of the generalized triangle before subdivision, and also the three sample points, which are Independence follows the same distribution. Let $\boldsymbol{v}_c$ is the center of gravity of the generalized triangle, after triangulating the generalized triangle, obviously, $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, $\boldsymbol{v}_c$ are the vertices of the triangular element after subdivision, let $\boldsymbol{v}_n$ is the center of gravity of the triangular element, then

$$E(\boldsymbol{v}_1) = E(\boldsymbol{v}_2) = E(\boldsymbol{v}_3) \tag{12}$$

$$\boldsymbol{v}_c = \frac{1}{3}(\boldsymbol{v}_1 + \boldsymbol{v}_2 + \boldsymbol{v}_3) \tag{13}$$

$$\boldsymbol{v}_n = \frac{1}{3}(\boldsymbol{v}_1 + \boldsymbol{v}_2 + \boldsymbol{v}_c) \tag{14}$$

$$E(\boldsymbol{v}_c) = \frac{1}{3}(E(\boldsymbol{v}_1) + E(\boldsymbol{v}_2) + E(\boldsymbol{v}_3)) = E(\boldsymbol{v}_1) \tag{15}$$

$$\begin{aligned} E(\boldsymbol{v}_n) &= \frac{1}{3}(E(\boldsymbol{v}_1) + E(\boldsymbol{v}_2) + E(\boldsymbol{v}_c)) \\ &= \frac{1}{3}(E(\boldsymbol{v}_1) + E(\boldsymbol{v}_2) + E(\boldsymbol{v}_1)) = E(\boldsymbol{v}_1) \end{aligned} \tag{16}$$

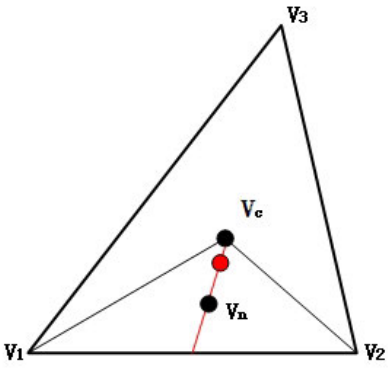

FIGURE 3. Newly synthesized sample (1).

therefore,

$$E(\boldsymbol{v}_1) = E(\boldsymbol{v}_2) = E(\boldsymbol{v}_3) = E(\boldsymbol{v}_c) = E(\boldsymbol{v}_n) \tag{17}$$

$$D(\boldsymbol{v}_c) = \frac{1}{9}(D(\boldsymbol{v}_1) + D(\boldsymbol{v}_2) + D(\boldsymbol{v}_3)) = \frac{1}{3}(D(\boldsymbol{v}_1)) \tag{18}$$

$$\begin{aligned} D(\boldsymbol{v}_n) &= D(\frac{1}{3}(\boldsymbol{v}_1 + \boldsymbol{v}_2 + \boldsymbol{v}_c)) \\ &= D(\frac{1}{3}\boldsymbol{v}_1 + \frac{1}{3}\boldsymbol{v}_2 + \frac{1}{9}(\boldsymbol{v}_1 + \boldsymbol{v}_2 + \boldsymbol{v}_3)) \\ &= \frac{11}{9}D(\boldsymbol{v}_c) \end{aligned} \tag{19}$$

(1) If $\boldsymbol{x}^* = (1-\delta)\boldsymbol{v}_c + \delta(\frac{\boldsymbol{v}_1+\boldsymbol{v}_2}{2})$, then

$$E(\boldsymbol{x}^*) = (1-\delta)E(\boldsymbol{v}_c) + \delta E(\frac{\boldsymbol{v}_1 + \boldsymbol{v}_2}{2}) = E(\boldsymbol{v}_c) \tag{20}$$

$$\begin{aligned} D(\boldsymbol{x}^*) &= (1-\delta)^2 D(\boldsymbol{v}_c) + \frac{\delta^2}{4}(D(\boldsymbol{v}_1) + D(\boldsymbol{v}_2)) \\ &= (\frac{5}{2}\delta^2 - 2\delta + 1)D(\boldsymbol{v}_c) \end{aligned} \tag{21}$$

$\delta = 0$ if $\frac{5}{2}\delta^2 - 2\delta + 1 = 1$, therefore, δ should be selected in the right neighborhood of 0 if $D(\boldsymbol{x}^*)$ is close to $D(\boldsymbol{v}_c)$, and the position of the synthesized point is shown in Figure 3 as the solid red point.

(2) If $\boldsymbol{x}^* = (1-\delta)\boldsymbol{v}_1 + \delta(\frac{\boldsymbol{v}_c+\boldsymbol{v}_2}{2})$, then

$$E(\boldsymbol{x}^*) = (1-\delta)E(\boldsymbol{v}_1) + \delta E(\frac{\boldsymbol{v}_c + \boldsymbol{v}_2}{2}) = E(\boldsymbol{v}_c) \tag{22}$$

$$\begin{aligned} D(\boldsymbol{x}^*) &= (1-\delta)^2 D(\boldsymbol{v}_1) + \frac{\delta^2}{4}(D(\boldsymbol{v}_2 + \frac{\boldsymbol{v}_1 + \boldsymbol{v}_2 + \boldsymbol{v}_3}{3})) \\ &= (\frac{9}{2}\delta^2 - 6\delta + 3)D(\boldsymbol{v}_c) \end{aligned} \tag{23}$$

then $h(\delta) = \frac{9}{2}\delta^2 - 6\delta + 3 \to 1 (\delta \to 0.6638)$, as a consequenc, δ should be near 0.6638 if $D(\boldsymbol{x}^*)$ is close to $D(\boldsymbol{v}_c)$, and the position of the synthesized point is shown in Figure 4 as the solid red point.

(3) If $\boldsymbol{x}^* = (1-\delta)\boldsymbol{v}_2 + \delta(\frac{\boldsymbol{v}_c+\boldsymbol{v}_1}{2})$, similarly to (2), $h(\delta) = \frac{9}{2}\delta^2 - 6\delta + 3 \to 1 (\delta \to 0.6638)$, as a consequenc, δ should be near 0.6638 if $D(\boldsymbol{x}')$ is close to $D(\boldsymbol{v}_c)$, and the position of the synthesized point is shown in Figure 5 as the solid red point.

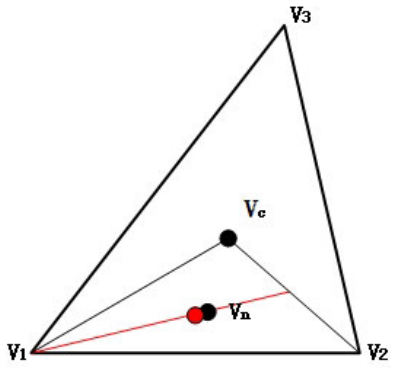

FIGURE 4. Newly synthesized sample (2).

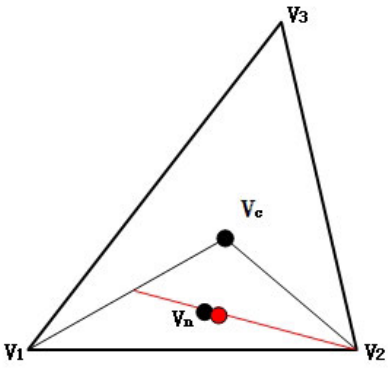

FIGURE 5. Newly synthesized sample (3).

Algorithm 1

Input: Minority samples H^+ and majority samples H^-,The number of two types of samples n_1, n_2 the features d.
Output: Balanced minority samples, majority samples. Balanced minority samples H^+, majority samples H^-_{new}.
Find the d points closest to the vertex $\boldsymbol{x}_i$, and then construct the generalized triangular element $Nerb_i = \{\boldsymbol{x}_i^l\}_{l=1}^d$;
For each sample $\boldsymbol{x}_i$, calculate the coordinates $\boldsymbol{v}_c$ of the center point of the generalized triangle according to formula (2);
Construct triangulation elements within each generalized triangle;
for $i = 1 \rightarrow n_1$ **do**
  for $k = 1 \rightarrow d-1$ **do**
    for $j = k+1 \rightarrow d$ **do**
      Calculate the size of each triangulation element according to formula (3);
      Calculate the number of samples generated within the triangulation element according to formula (4) - (6);
    end for
  end for
end for
Synthesize new samples:
for $i = 1 \rightarrow n_1$ **do**
  for $k = 1 \rightarrow d-1$ **do**
    for $j = k+1 \rightarrow d$ **do**
      Generate a new sample according to formula (7)-(8);
    end for
  end for
end for

IV. NUMERICAL SIMULATION EXPERIMENTS

A. DATASETS

For the experiment, datasets from the UCI database were selected, including Balance-scale, Yeast, Ecoli, Page-blocks, Testpad, Glass and Wine. The datasets can be accessed at the website 'https://archive.ics.uci.edu/'. The attributes of each set are detailed in Table 1.

TABLE 1. Dataset information.

Dataset	Majority	Minority	Feature	Unbalanced ratio	Binary classification
Balance-scale	337	288	4	1.17	1,2vs3
Yeast	1484	35	8	42.4	1-4,6-10vs5
Page-blocks1	4913	329	10	14.93	1vs2
Page-blocks2	4913	231	10	21.27	1vs3,4,5
Testpad	4000	100	5	4	1vs2
Ecoli	180	20	7	9	145vs6
Wine	107	71	13	1.51	1,3vs2
Glass	175	17	9	10.29	127vs3

B. COMPARISON EXPERIMENT DESIGN

1) CLASSIFIER

Stochastic configuration network(SCN) [25] is selected for data training and classification, owing to its advantages of minimal adjustment parameters, simplicity of model, and ease of convergence.

SCN is a machine learning model designed to tackle regression and classification tasks based on data. Its architectural topology consists of input layer, hidden layer, and output layer, organized in the form of a fully connected network. The neurons within the hidden layer dynamically expand until the model attains an acceptable error threshold or reaches the maximum number of iterations. Predominantly, it employs the least squares method to update its parameters in the trainning process, incorporating an 'inequality supervision mechanism' to guarantee the convergence of the model during this process. Compared to other machine learning models, it guarantees theoretical convergence, a characteristic that is not universally present. Furthermore, in contrast to deep learning models, it combines simplicity of structure with a relatively modest computational requirement, thereby justifying its selection as the classifier for this study.

The structure of SCN is shown as Figure 6, x_i represents the i-th feature of the input data, y_j represents the j-dimensional data of the model output, and S_l represents the activation function, where $i = 1, 2, \ldots, d$, $j = 1, 2$, $l =$

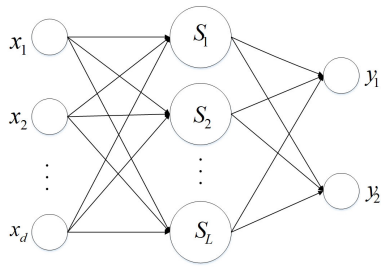

FIGURE 6. SCN structure diagram.

$1, 2, \ldots, L$, L is the number of hidden layer units. The network output is as follows.

$$y = (y_1, y_2) = (\sum_{l=1}^{L} \beta_{l1} S_l(w^T x + b), \sum_{l=1}^{L} \beta_{l2} S_l(w^T x + b)) \tag{24}$$

where w and b denote the weight and bias, respectively, associated with the L-th hidden node relative to the inputs, β_{l1} and β_{l2} represent the weights connecting the L-th hidden node to y_1 and y_2. The optimal parameters w and b should be selected to satisfy condition:

$$\langle E, S_L \rangle^2 \geq b_g^2 (1 - c - u_L) \|e_{L-1}\|^2 \tag{25}$$

Here, E represents the error of the network configured with L -1 hidden layer nodes, b denotes the upper bound of the function S, c is a constant close to 1, and u_L satisfying $\lim_{L \to \infty} u_L = 0$. The calculation of β_{l1} and β_{l2} is as follows.

$$(\beta_{L1}, \beta_{L2}) = argmin_\beta E \tag{26}$$

Within each category, 70% of the samples were randomly allocated as training samples, while the remaining 30% were designated as test samples. Subsequent to balancing the sample distribution using IFE-SMOTE, SCN was trained. Following this, a classification test was conducted on the test set to ascertain and validate the efficacy of the algorithm.

2) COMPARISON ALGORITHM

The proposed method is compared with Random Oversampling, SMOTE, SMOTEN, Bordline-SMOTE, KmeansSMOTE, ASDSYN, FE-SMOTE, etc. The above methods include classical oversampling methods and new oversampling methods in recent years.

3) EXPERIMENTAL METHOD AND EVALUATION METRICS

The evaluation criteria for classification encompass Geometric Mean (G-Mean), F-measure, and Area Under the Curve (AUC), with a closer proximity to 1 indicating superior performance. For the implementation of various data balancing methodologies, an initial step involves constructing identically specified training and test sets. Subsequently, the balancing process is exclusively applied to the training set. This balanced training data is then fed into the classifier for the purpose of learning. Once the classifier has undergone training, the test set is introduced for evaluation. Ultimately, a comparative analysis is conducted by averaging the outcomes of 50 classifier runs on the test set.

TABLE 2. Experimental Comparison Results (Balance-scale).

Algorithm	G-mean	F-measure	AUC
Raw data	0.9484	0.9470	0.9489
SMOTE	0.9541	0.9541	0.9542
Borderline-SMOTE	0.9652	0.9646	0.9652
ADASYN	0.9545	0.9543	0.9546
RandomOverSampler	0.9413	0.9414	0.9415
KMeansSMOTE	0.9280	0.9275	0.9280
SMOTENC	0.9413	0.9411	0.9414
FE-SMOTE	0.9813	**0.9813**	0.9815
IFE-SMOTE	**0.9851**	0.9785	**0.9852**

TABLE 3. Experimental Comparison Results (Yeast).

Algorithm	G-mean	F-measure	AUC
Raw data	0.6380	0.9800	0.6855
SMOTE	0.7915	0.9640	0.8036
Borderline-SMOTE	0.7791	0.9697	0.7947
ADASYN	0.7784	0.9688	0.7939
RandomOverSampler	0.7923	0.9752	0.8071
KMeansSMOTE	0.7782	0.9435	0.8022
SMOTENC	0.8168	0.9497	0.8217
FE-SMOTE	0.7813	0.9801	0.8058
IFE-SMOTE	**0.8500**	**0.9802**	**0.8606**

TABLE 4. Experimental Comparison Results (Page-blocks1).

Algorithm	G-mean	F-measure	AUC
Raw data	0.9324	0.9925	0.9344
SMOTE	0.9502	0.9914	0.9522
Borderline-SMOTE	0.9504	0.9902	0.9526
ADASYN	0.9320	0.9910	0.9423
RandomOverSampler	0.9505	0.9901	0.9508
KMeansSMOTE	0.9028	0.9930	0.9072
SMOTENC	0.9362	0.9909	0.9375
FE-SMOTE	0.9394	0.9930	0.9316
IFE-SMOTE	**0.9525**	**0.9932**	**0.9532**

TABLE 5. Experimental Comparison Results (Page-Blocks2).

Algorithm	G-mean	F-measure	AUC
Raw data	0.7029	0.9717	0.7467
SMOTE	0.9026	0.9756	0.9044
Borderline-SMOTE	0.9021	0.9712	0.9034
ADASYN	0.9103	0.9624	0.9106
RandomOverSampler	0.8916	0.9751	0.8945
KMeansSMOTE	0.7601	0.9858	0.7864
SMOTENC	0.8287	0.9880	0.8415
FE-SMOTE	0.9108	0.9881	0.9112
IFE-SMOTE	**0.9139**	**0.9885**	**0.9169**

C. EXPERIMENTAL RESULTS

D. RESULT ANALYSIS OF THE EXPERIMENT

Tables 2 to Table 9 comprehensively present the G-mean, F-measure, and AUC values of nine oversampling techniques

TABLE 6. Experimental Comparison Results (TestPad).

Algorithm	G-mean	F-measure	AUC
Raw data	0.9961	0.9962	0.9961
SMOTE	0.9958	0.9958	0.9959
Borderline-SMOTE	0.9961	0.9962	0.9961
ADASYN	0.9957	0.9960	0.9954
RandomOverSampler	0.9957	0.9957	0.9957
KMeansSMOTE	0.9957	0.9957	0.9957
SMOTENC	0.9953	0.9954	0.9953
FE-SMOTE	0.9954	0.9954	0.9954
IFE-SMOTE	**0.9979**	**0.9979**	**0.9980**

TABLE 7. Experimental Comparison Results (Ecoli).

Algorithm	G-mean	F-measure	AUC
Raw data	0.9621	0.9883	0.9633
SMOTE	0.9632	**0.9899**	0.9650
Borderline-SMOTE	0.9513	0.9884	0.9550
ADASYN	0.9029	0.9768	0.9100
RandomOverSampler	0.8835	0.9769	0.8933
KMeansSMOTE	0.9461	0.9815	0.9483
SMOTENC	0.8696	0.9737	0.8816
FE-SMOTE	0.9399	0.9747	0.9416
IFE-SMOTE	**0.9776**	0.9864	**0.9783**

TABLE 8. Experimental Comparison Results (Wine).

Algorithm	G-mean	F-measure	AUC
Raw data	0.9117	0.9245	0.9035
SMOTE	0.9017	0.9245	0.9061
Borderline-SMOTE	0.9267	0.9441	0.9295
ADASYN	0.9093	0.9320	0.9134
RandomOverSampler	0.8970	0.9222	0.9015
KMeansSMOTE	0.9025	0.9222	0.9064
SMOTENC	0.9220	0.9222	0.9252
FE-SMOTE	0.9572	0.9625	0.9574
IFE-SMOTE	**0.9657**	**0.9707**	**0.9660**

TABLE 9. Experimental Comparison Results (Glass).

Algorithm	G-mean	F-measure	AUC
Raw data	0.4469	0.8548	0.5682
SMOTE	0.7253	0.8638	0.7342
Borderline-SMOTE	0.7640	0.8806	0.7708
ADASYN	**0.7925**	0.8853	**0.7974**
RandomOverSampler	0.6953	0.8638	0.7117
KMeansSMOTE	0.7242	0.8632	0.7342
SMOTENC	0.0732	0.8518	0.4862
FE-SMOTE	0.6980	0.8803	0.7205
IFE-SMOTE	0.7757	**0.9078**	0.7897

applied to eight unbalanced datasets. With regard to G-mean, IFE-SMOTE emerges as the dominant approach across seven datasets, albeit yielding a lower performance than ADASYN on dataset Glass, thus securing the second position. In the realm of F-measure, IFE-SMOTE holds the top spot on six datasets, while ranking second on dataset Balance-scale and Ecoli, where it falls slightly behind FE-SMOTE and SMOTE, respectively. Turning to G-AUC, IFE-SMOTE maintains its dominance on seven datasets, yet on dataset Glass, it is surpassed by ADASYN, once again occupying the second rank. Figures 7 to 9 offer a vivid illustration of the performance exhibited by IIFE SMOTE. The results demonstrated that the average G-mean, F-measure, and AUC for the classification of the eight datasets by IFE-SMOTE were 0.9273, 0.9754, and 0.9309, respectively. These values were 0.0268, 0.0059, and 0.0253 higher than those of FE-SMOTE, and 0.0493, 0.0169, and 0.0402 higher than the average performance indicators of the other algorithms, respectively. In summary, the proposed IFE SMOTE algorithm effectively balances the datasets and subsequently attains a superior classification performance during the training and learning phases.

This result is related to the principle of various oversampling methods. Random oversampling is the simplest and most blind method, but it can also achieve good results in individual experiments, which is related to the feature distribution of the original data and the small size of the dataset, which increases the visibility of a few class samples, so that the model can strengthen the learning effect through repeated learning. SMOTE and SMOTEN methods expand the sample generation space linearly, but still have some randomness, and this method only performs well on a few datasets. ADASYN tries to generate more samples in a few class sample areas that are difficult to classify, yet it still has limitations. BorderlineSMOTE, on the other hand, strategically targets samples that significantly influence classification boundaries, potentially enhancing class separability but also introducing a risk of increased class overlap. In comparative experiments, BorderlineSMOTE has demonstrated relatively favorable performance. Meanwhile, KMeansSMOTE suffers from sensitivity to initial cluster center selection and outliers, rendering the algorithm inherently unstable. Specifically, when dealing with exceedingly small sample sizes within certain classes, KMeansSMOTE may struggle to form meaningful clusters. Lastly, FE-SMOTE boasts an innovative approach that extensively explores the generation space for new samples, resulting in a distribution that closely mirrors the original sample distribution. This characteristic imparts FE-SMOTE with certain advantages in comparative evaluations.

IFE-SMOTE worked better because the point to keep the statistical characteristics (expectation and variance) of the raw data as close as possible to the newly generated data. And it retains the advantages of FE-SMOTE not lowering the dimensionality of the data, handling outliers efficiently, and using all few class samples to generate new samples.

Judging from the experimental results, IFE-SMOTE is ahead of other methods on dataset Yeast,Page-blocks1,Page-blocks2,TestPad and Wine. The imbalance ratio of the four dataset(Yeast,Page-blocks1,Page-blocks2,Glass) is all greater than 10, and dataset Page-blocks1,Page-blocks2 and Wine have a higher dimension, IFE-SMOTE continues to hold the overall leading position. The above results indicate that in addition to the use of the general data set, IFE-SMOTE still has good applicability when the imbalance is

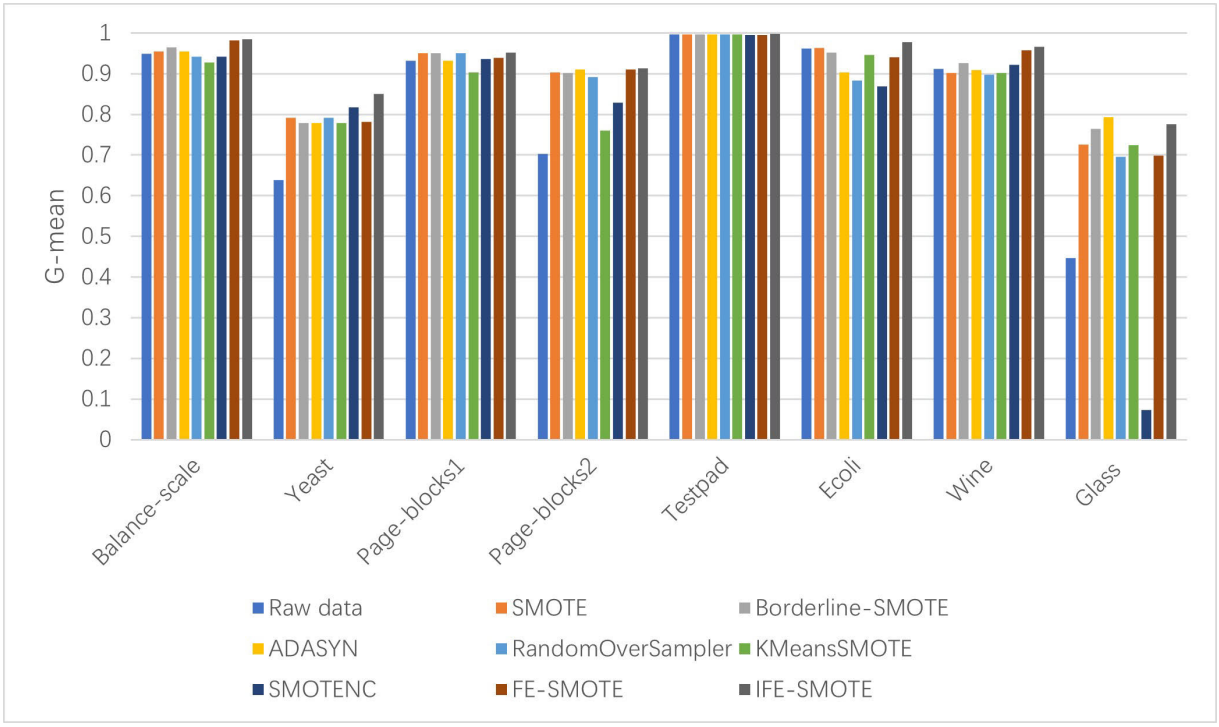

FIGURE 7. G-mean of 9 methods on 8 datasets.

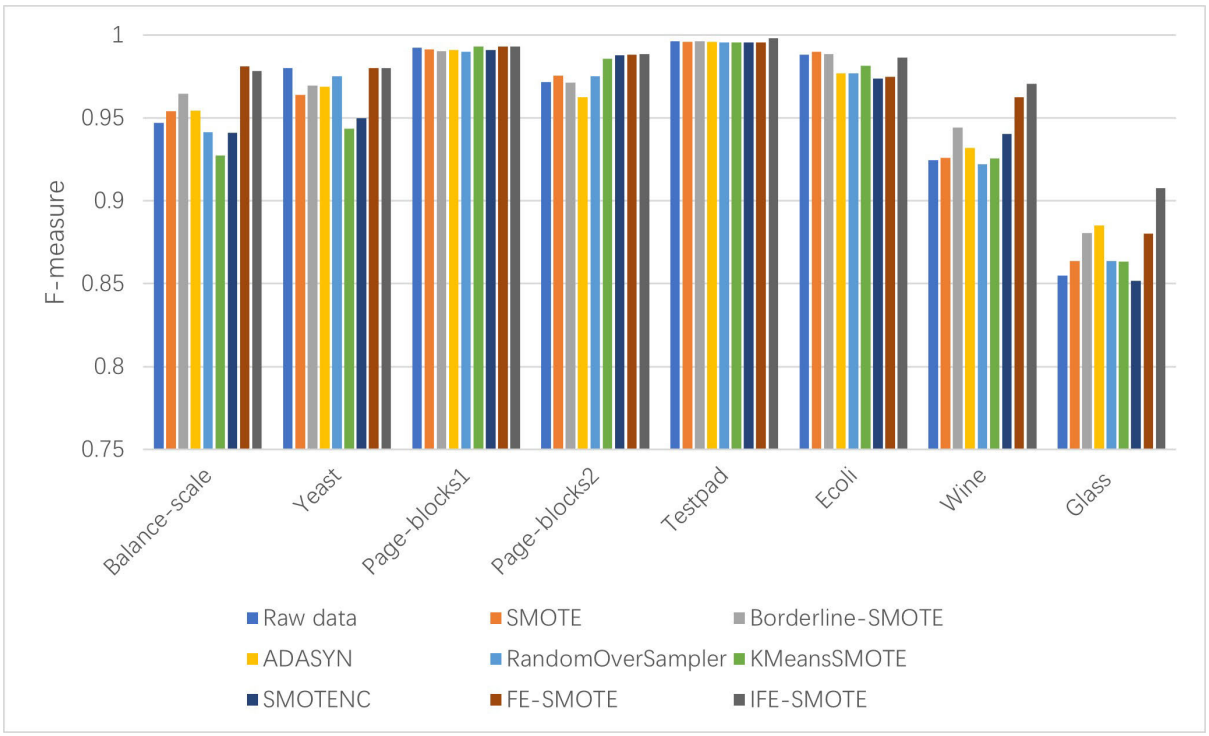

FIGURE 8. GF-measure of 9 methods on 8 datasets.

large and the data dimension is high. The results suggest that IFE-SMOTE exhibits robust applicability in scenarios characterized by significant class imbalance and high data dimensionality.

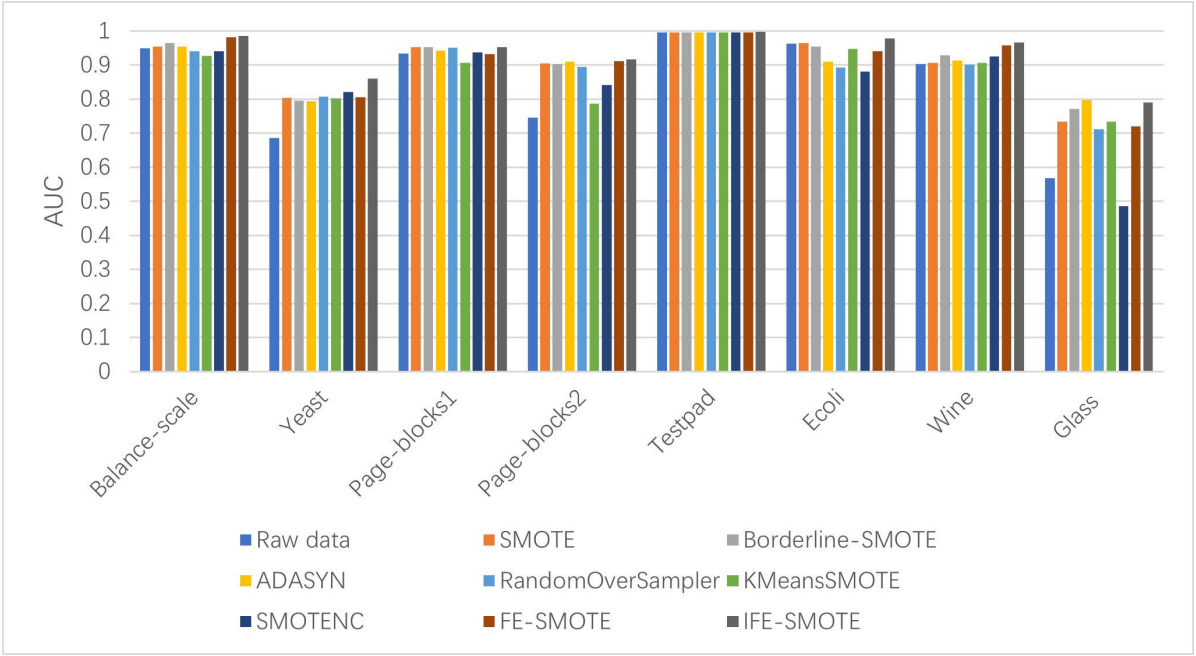

FIGURE 9. AUC of 9 methods on 8 datasets.

V. CONCLUSION

This paper introduces IFE-SMOTE, an oversampling method for unbalanced data in data-driven models. As an enhancement of FE-SMOTE, IFE-SMOTE employs triangulation to selectively expand the generation space of new samples, ensuring that the expectations and variances of new samples align closely with those of the triangle centers of original minority samples. This promotes statistical consistency between new and original data, bolstering model learning performance. The space to be studied is as follows: The statistical feature consistency conclusion presented herein assumes constant parameters. However, parameter variations within a neighborhood may subtly alter the expectations and variances of new samples in practical experiments, necessitating further investigation of the statistical feature consistency method. The current analysis assumes constant parameters, yet future research should explore parameter variability's impact on sample statistics. Furthermore, optimization algorithms [26], [27] offer a means to further enhance the parameter settings. Additionally, incorporating other sample numerical characteristics could further refine new sample distribution. In addition to expectation and variance, other numerical characteristics of the sample can be further considered to make the distribution of the new sample more reasonable.

REFERENCES

[1] M. Nahas, M. Hussein, and A. Keshk, "Imbalanced data oversampling technique based on convex combination method," *Int. J. Comput. Inf.*, vol. 9, no. 1, pp. 15–28, Mar. 2022.

[2] I. Mani and I. Zhang, "KNN approach to unbalanced data distributions: A case study involving information extraction," in *Proc. ICML Workshop Learn. Imbalanced Datasets*, 2003, pp. 1–7.

[3] P. Hart, "The condensed nearest neighbor rule," *IEEE Trans. Inf. Theory*, vol. IT-14, no. 3, pp. 515–516, May 1968.

[4] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Trans. Syst., Man, Cybern.*, vols. SMC–2, no. 3, pp. 408–421, Jul. 1972.

[5] M. Kubát and S. Matwin, "Addressing the curse of imbalanced training sets: One-sided selection," in *Proc. 14th Int. Conf. Mach. Learn.* Nashville, TN, USA: Morgan Kaufmann, Jan. 1997, pp. 179–186.

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

[7] 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.

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

[9] 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.*, Jun. 2008, pp. 1322–1328.

[10] 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.

[11] Y. Bao and S. Yang, "Two novel SMOTE methods for solving imbalanced classification problems," *IEEE Access*, vol. 11, pp. 5816–5823, 2023.

[12] A. Puri and M. K. Gupta, "Improved hybrid bag-boost ensemble with K-means-SMOTE–ENN technique for handling noisy class imbalanced data," *Comput. J.*, vol. 65, no. 1, pp. 124–138, Jan. 2020.

[13] G. A. Pradipta, R. Wardoyo, A. Musdholifah, and I. N. H. Sanjaya, "Radius-SMOTE: A new oversampling technique of minority samples based on radius distance for learning from imbalanced data," *IEEE Access*, vol. 9, pp. 74763–74777, 2021.

[14] S. Yan, Z. Zhao, S. Liu, and M. Zhou, "BO-SMOTE: A novel Bayesian-optimization-based synthetic minority oversampling technique," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 54, no. 4, pp. 2079–2091, Apr. 2024.

[15] S. Cost and S. Salzberg, "A weighted nearest neighbor algorithm for learning with symbolic features," *Mach. Learn.*, vol. 10, no. 1, pp. 57–78, Jan. 1993.

[16] X. Li, B. Zou, L. Wang, M. Zeng, K. Yue, and F. Wei, "A novel LASSO-based feature weighting selection method for microarray data classification," in *Proc. IET Int. Conf. Biomed. Image Signal Process. (ICBISP)*, Beijing, China, Nov. 2015, pp. 1–5.
[17] C. Zhang, J. Guo, and J. Lu, "Research on classification method of high-dimensional class-imbalanced data sets based on SVM," in *Proc. IEEE 2nd Int. Conf. Data Sci. Cyberspace*, Shenzhen, China, Jun. 2017, pp. 60–67.
[18] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, no. 1, pp. 389–422, 2002.
[19] L. Guo and S. Wang, "Membrane protein type prediction for high-dimensional imbalanced datasets," in *Proc. 9th Int. Conf. Inf. Technol. Med. Educ. (ITME)*, Hangzhou, China, Oct. 2018, pp. 847–851.
[20] S. Gazzah and N. E. B. Amara, "New oversampling approaches based on polynomial fitting for imbalanced data sets," in *Proc. 8th IAPR Int. Workshop Document Anal. Syst.*, Sep. 2008, pp. 677–684.
[21] D. Dablain, B. Krawczyk, and N. V. Chawla, "DeepSMOTE: Fusing deep learning and SMOTE for imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 9, pp. 6390–6404, Sep. 2023.
[22] G. Douzas and F. Bacao, "Geometric SMOTE a geometrically enhanced drop-in replacement for SMOTE," *Inf. Sci.*, vol. 501, pp. 118–135, Oct. 2019.
[23] A. M. de Carvalho and R. C. Prati, "DTO-SMOTE: Delaunay tessellation oversampling for imbalanced data sets," *Information*, vol. 11, no. 12, p. 557, Nov. 2020.
[24] Y. Chen, W. Pedrycz, J. Wang, C. Zhang, and J. Yang, "A new oversampling method based on triangulation of sample space," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 54, no. 2, pp. 774–786, Feb. 2024.
[25] D. Wang and M. Li, "Stochastic configuration networks: Fundamentals and algorithms," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3466–3479, Oct. 2017.
[26] Z. Zhao, J. Cheng, J. Liang, S. Liu, M. Zhou, and Y. Al-Turki, "Order picking optimization in smart warehouses with human–robot collaboration," *IEEE Internet Things J.*, vol. 11, no. 9, pp. 16314–16324, Mar. 2024.
[27] Z. Zhao, Q. Jiang, S. Liu, M. Zhou, X. Yang, and X. Guo, "Energy, cost and job-tardiness-minimized scheduling of energy-intensive and high-cost industrial production systems," *Eng. Appl. Artif. Intell.*, vol. 133, Jul. 2024, Art. no. 108477.

JINGJING LIU received the B.S. degree in mathematics and applied mathematics from Shaanxi Normal University, Xi'an, China, in 2009, and the M.S. degree in computational mathematics from Dalian University of Technology, Dalian, China, in 2012. She is currently an Associate Professor with Shenyang Institute of Technology, China. She has published five articles in SCI and EI, hosted and participated in more than ten projects, and authorized one Chinese invention patents. Her research interests include datadriven intelligent control and machine learning algorithms.

YEFENG LIU received the B.S. degree in automation from Qingdao University, Qingdao, China, in 2005, and the Ph.D. degree from Northeastern University, Shenyang, China, in 2015. He is currently a Professor with Liaoning Key Laboratory of Information Physics Fusion and Intelligent Manufacturing for CNC Machine, Shenyang Institute of Technology, Shenfu New District, China. He has published 60 peer-reviewed international journal and conference papers and more than 30 articles in SCI and EI, hosted and participated in more than 20 projects, and authorized 15 Chinese invention patents. His current research interests include the development of manufacturing execution systems, production planning and scheduling, and intelligent optimization methods.

YANWEI MA received the master's degree in computational mathematics from Dalian University of Technology. She is currently an Associate Professor with Shenyang Institute of Technology. Her current research interests include the research and development of numerical control technology for digital manufacturing, artificial intelligence, and intelligent production management. She has participated in the application for three national invention patents and the research and development of multiple national, provincial, and ministerial level scientific research projects.

QICHUN ZHANG (Senior Member, IEEE) received the B.Eng. degree in automation and the M.Sc. degree in control theory and control engineering from Northeastern University, China, and the Ph.D. degree in electrical and electronic engineering from The University of Manchester, U.K. He is currently the Chair Professor of inclusive AI and the Head of Research at the School of Creative and Digital Industries, Buckinghamshire New University. Prior to this position, he held roles as an Associate Professor with the University of Bradford, a Senior Lecturer with De Montfort University, and a Senior Research Officer with the University of Essex. His current research interests include stochastic dynamic systems, artificial intelligence, data-driven modeling, and design. He has published around 100 publications widely over the above areas. He is a Chartered Engineer. He is also serving as an Associate Editor for the *Journal of Intelligent Manufacturing*, *Cluster Computing*, and IEEE Access, and is a member of the EPSRC Peer-Review College.