Manifold neighboring envelope projection imbalanced ensemble algorithm with consistent FCM

Yiwen Wang[a], Yongming Li[a]*, Fan Li[a], Yinghua Shen[b], Pin Wang[a]*

(a. School of Microelectronics and Communication Engineering, Chongqing University, Chongqing, 400044, China;
b. School of Economics and Business Administration, Chongqing University, Chongqing, 400044, China)

Abstract

Class imbalanced is a salient and challenging problem in current research, but the existing imbalanced ensemble methods are based on the original samples and lack the mining of structural information between samples. To solve this problem, this paper proposes a manifold neighboring envelope projection imbalanced ensemble algorithm with consistent FCM (MNEFD_IE). The algorithm can effectively mine the nearest neighbors and hierarchical structure relationships among samples, thereby reconstructing the structural information into structured samples-envelope samples. First, the dataset is divided into several balanced subsets based on the undersampling method. Second, a manifold neighboring envelope sample reconstruction mechanism (MNESR) is designed to mine the structural relationships between the samples and their manifold nearest neighbors, thereby obtaining neighboring envelope samples. Then, a consistent fuzzy clustering algorithm (CFCMD) based on the minimum interlayer discrepancy mechanism (MIDMD) is designed to mine the hierarchical structure information among samples and map them to the clustering centers to generate high-quality neighboring cluster envelope samples. Finally, the base classifiers are trained based on the envelope samples, and their prediction results are fused by a 2D sparse fusion mechanism (2D-SFM). In the experimental section, the proposed MNEFD_IE is compared with representative classical and state-of-the-art imbalanced ensemble algorithms on 38 public datasets. The results show that the proposed algorithm performs significantly better.

Keywords

Class imbalanced problem; ensemble learning; envelope learning; fuzzy c-means clustering; domain adaptation.

1. Introduction

The challenging class imbalanced problem is widely present in many fields of data analysis [1] and data mining [2]. In an imbalanced dataset, the number of majority class samples is much larger than that of the minority class samples. The situation in which most traditional classification algorithms are proposed under the assumption of balanced data distribution and equal misclassification cost (balanced) leads to a classification model that biases toward the majority class and ignores the minority class for imbalanced datasets [3]. However, in most cases of class imbalanced problems, information from minority classes will be more important, such as in biomedical diagnosis, anomaly detection, fraud detection, etc [4-5]. The goal of imbalanced learning methods is to improve both the classification performance of the model for minority class samples and the overall classification result [6].

Existing imbalanced learning algorithms can be mainly classified into algorithm-level methods [7], data-level methods [8] and ensemble learning methods [9]. Algorithm-level methods directly enhance the sensitivity of classifiers to minority classes by modifying the model's structure [10]. Data-level methods balance the class distribution mainly through data resampling: this type of method can be further classified as undersampling [11], oversampling [12], or hybrid sampling [13]. Ensemble learning methods combine multiple base classifiers to obtain stable and robust classification results, and they are also often applied in combination with data-level methods [14].

Imbalanced ensemble methods are widely used to solve class imbalanced problems because of their superior accuracy, stability, and robustness [15-16]. However, these methods mainly utilize a single sample as the base unit for modeling, so the methods suffer from some limitations.

1) The training error is computed based on the training samples one by one and does not consider the structural information of similar samples. Therefore, the decision boundary is complex, thereby leading to low accuracy, low generalization, and high sensitivity to noise.

2) There exists high overlap for the divided subsets, especially for the minority samples. Therefore, the diversity among the subsets is low, thereby restricting ensemble learning.

2. Motivation and contribution

To ameliorate these limitations, this paper analyzes the possible reasons and provides a solution.

For the first limitation, the possible reason is that the existing methods do not consider the correlation relationship between similar samples, so they cannot extract the structural information (high-level information) among samples. The structured samples are more stable and less sensitive to noise. Fig. 1 shows the distribution of samples. Fig. 1 (a) shows the distribution of the original samples under feature spaces F1 and F2. As shown in Fig. 1 (a), the separability is low. Fig. 1 (b) shows the distribution of the sample envelopes by similarity. Fig. 1 (c) shows the distribution of the clustering centers of the samples. As shown in Fig. 1 (c), the separability is significantly improved without changing the features, and the noise is removed. The clustering centers can be viewed as structured samples by extracting the similarity structural information among sample envelopes.

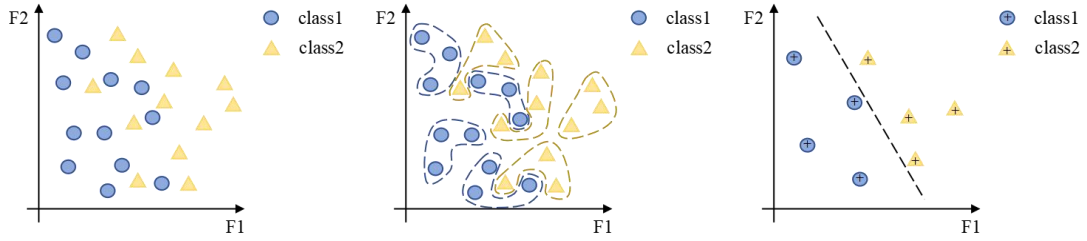


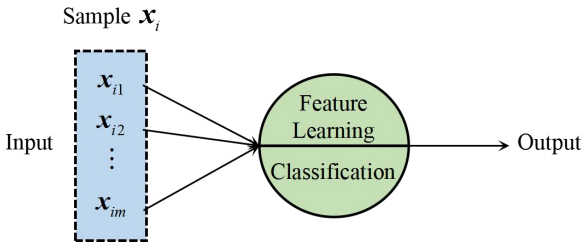
Fig. 1. Separability analysis of samples: (a) Separability of original samples; (b) Separability of sample envelopes ; (c)Separability of clustered samples (structured samples)

As we know, due to some reasons of data collection and subject sources, there exists local similarity among samples. However, the clustering just considers the global similarity of the samples, but ignore the local similarity of the samples. Therefore, to enhance the quality of the clustering, it is necessary to mine the local similarity among samples.

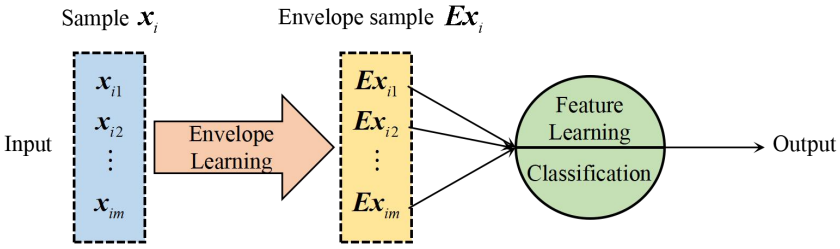
For the second limitation, the possible reason is that all the samples of every subset are obtained from the same dataset by resampling. When the imbalance ratio is very high or the resampling ratio is high, the overlap among subsets is severe, thereby leading to low diversity, even for sample weighting

and selection. Based on the analysis above, the original samples are homogeneous, but the original samples and structured samples are heterogeneous, so the overlap and diversity will be improved with the structured samples.

Therefore, to solve the limitations above, it is necessary to construct the structured samples by mining the local and global similarity among samples. The structured samples merge the similar samples and reflect their similarity structural information, so are called 'envelope samples'. The existing single-sample based imbalanced learning algorithm can be converted to the envelope-sample based one. The Fig.2 shows the major difference between the two kinds of methods. As the Fig. 2 shows, the structured samples can be better for modeling in terms of accuracy, anti-noise, and so on, as Xia and Wang said [17].



(a) The existing single-sample based method



(b) The envelope-sample based method

Fig. 2. The comparison of the envelope-sample based method and the single-sample based method

Based on the analysis above, this paper proposes a new imbalanced ensemble method that mines the correlation relationship between similar samples to construct envelope samples for subsequent modeling. This proposed method generates two kinds of envelope samples: neighboring envelope samples and neighboring cluster envelope samples. The former is for mining the local similarity of samples, and the latter is for mining the global similarity of samples. First, the training set is divided into balanced subsets. Then, we design the sample neighboring envelope mechanism to obtain the sample envelope for every original sample. The intersample nearest neighbor relationships are mined by transposition projection on the sample envelopes, and these nearest neighbor relationships are reconstructed into structured samples-neighboring envelope samples. After that, consistent fuzzy clustering is performed on the neighboring envelope samples to mine the similarity structural information between the samples and map this structural information to the clustering centers, which are the neighboring cluster envelope samples. The neighboring cluster envelope samples significantly improve diversity . The base classifier is trained for each layer of samples, and the results are fused by a two-dimensional sparse fusion mechanism (2D-SFM). Finally, the neighboring envelope projection imbalanced ensemble algorithm with consistent FCM (MNEFD_IE) is proposed. To guarantee the quality of the envelope samples, manifold neighboring envelope sample reconstruction and consistent

fuzzy clustering are combined by joint optimization. The main contributions of this paper are as follows.

1) A manifold neighboring envelope sample reconstruction mechanism (MNESR) is proposed to mine the intersample nearest neighbor relationships by manifold nearest neighbor envelope and transposition projection reconstruction to obtain structured samples-neighboring envelope samples with rich structural information.

2) A consistent fuzzy clustering algorithm (CFCMD) is proposed based on neighboring envelope samples. The CFCMD algorithm obtains neighboring cluster envelope samples by soft clustering of similar samples and designing a minimum interlayer discrepancy mechanism based on maximum mean discrepancy (MIDMD).

3) A new imbalanced ensemble algorithm (MNEFD_IE) is proposed based on joint MNESR and CFCMD to construct high-quality envelope samples to improve the classification performance and generalization ability of subsequent modeling. Unlike existing imbalanced ensemble algorithms, the proposed algorithm transforms the existing original-sample based method into the structured sample (envelope sample) based method by mining the structural information among samples.

4) Two kinds of envelope samples are generated in this paper. In addition, representative datasets with a high range of imbalance ratios are studied.

The remainder of this paper is organized as follows. The related work on imbalanced learning is reviewed in Section III. Section IV describes the proposed algorithm (MNEFD_IE). Section V presents the complexity analysis of MNEFD_IE. In Section VI, comparison experiments are performed and analyzed. Finally, conclusions are drawn in Section VII.

3. Related works

This section reviews related work on imbalanced learning, which can be mainly classified into data-level methods, algorithm-level methods, and ensemble learning methods.

Data-level methods, also called resampling methods, balance the dataset by modifying the distribution in the original data to improve the classification performance of the minority class. Resampling methods mainly use the strategies of undersampling the majority class samples or oversampling the minority class samples to balance the dataset. Among the undersampling methods, random undersampling (RUS) is the simplest method, randomly removing some of the majority class samples to balance the dataset [18]. The undersampling method ensures the reality of the data used and the efficiency of training but loses some valid information of the samples. Among the oversampling methods, the random oversampling method (ROS) is the simplest method: it randomly replicates the minority samples to balance the dataset. Although this method does not lose any data information, it can easily lead to overfitting [19]. To solve the above problems, many different resampling methods have been proposed in recent years. For example, the synthetic minority oversampling technique (SMOTE) [13] synthesizes new minority samples by random interpolation between minority class samples and their nearest neighbor samples. To improve the performance of the SMOTE algorithm, the adaptive synthetic sampling (ADASYN) [20] and safe-level-SMOTE [21] algorithms have been proposed to fine-tune the newly synthesized samples. Another typical resampling method is the clustering-based resampling method, which is used to identify and preserve the class space [22].

Algorithm-level methods aim to address the class imbalanced problem from an algorithm design perspective by modifying the learning algorithm and thus directly enhancing the sensitivity of the classifier to minority classes. Among them, cost-sensitive learning [23] is one of the most commonly

used methods, which optimizes the overall misclassification cost by assigning different misclassification costs to different classes. Kukar and Kononenko [24] proposed a cost-sensitive neural network. Inspired by AdaBoost, the AdaCost [25] algorithm inherits the framework of AdaBoost while modifying the weight update strategy so that minority class samples receive more attention. Specifically, for the imbalanced problem, the minority class is very easy to misclassify and its cost deserves to be studied in further detail. The reason why this method is not as popular as the resampling method is that, in most cases, the assignment of misclassification costs is not available from data centers or is determined directly by experts [26].

The ensemble learning method is a widely used method to solve the class imbalanced problem. Its main idea is to train multiple classifiers based on several class-balanced subsets and to obtain higher classification accuracy, stability, and robustness than a single classifier by combining these classifiers. Algorithm-level and data-level methods are often combined with ensemble methods to further improve performance, which can usually be classified into cost-sensitive boosting approaches [27] and ensemble learning algorithms with embedded data preprocessing techniques [28]. The cost-sensitive boosting approaches also classify different classes into different misclassification costs, differing from the nonensemble cost-sensitive approach in that the optimization of the overall misclassification cost of the approach is guided by the boosting algorithm. In contrast, ensemble learning algorithms with embedded data preprocessing techniques are more widely used to address the imbalanced problem. These methods typically use a data-level method to balance the dataset distribution, train multiple base classifiers based on the balanced subsets, and then fuse classification results. A large number of studies [29] have shown that bagging [30] and boosting [31] combined with data-level methods achieve good performance, such as SMOTEBagging [32], UnderBagging [33], SMOTEBoost [34], and RUSBoost [35]. In addition, Galar et al. [36] proposed an ensemble learning algorithm based on the evolutionary undersampling approach (EUSBoost), which selects different subsets of majority class samples to train base classifiers based on the evolutionary undersampling approach, thus improving the diversity and generalization performance of base classifiers. Liu et al. [37] proposed EasyEnsemble and BalanceCascade. EasyEnsemble obtains multiple training subsets by randomly dividing multiple majority-class subsets and fusing them with minority-class samples by using AdaBoost as the base classifier for training based on these training subsets. The basic framework of BalanceCascade is the same as EasyEnsemble, except that the algorithm uses an AdaBoost classifier to predict all majority samples after each training. The false positive rate is controlled by adjusting the threshold to remove all correct samples and iterating until the number of remaining majority samples equals the number of minority samples.

4. Method

4.1. Notation

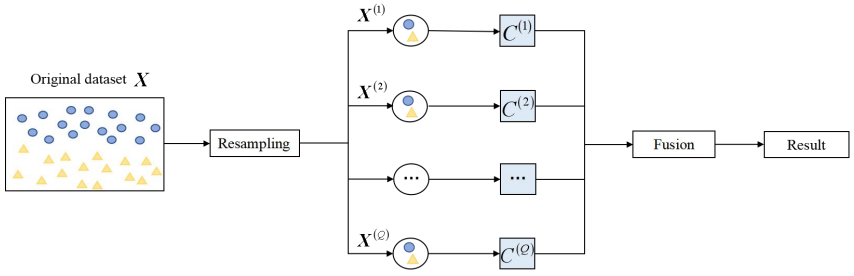
Given a training set $\{\boldsymbol{X},\boldsymbol{Y}\}=\{(\boldsymbol{x}_1,\boldsymbol{y}_1),(\boldsymbol{x}_2,\boldsymbol{y}_2),...,(\boldsymbol{x}_N,\boldsymbol{y}_N)\}$, where $\boldsymbol{y}_n$ expresses the true label of a sample $\boldsymbol{x}_n\in\mathbb{R}^d, n=1,2,...,N$. The numbers of majority samples and minority samples in the original dataset $\boldsymbol{X}$ are denoted as N_{maj} and N_{min}, respectively. Random undersampling of majority class samples is performed to obtain Q subsets of majority class samples with the number of samples N_{min}. The Q subsets of majority class samples are fused with minority class samples to obtain a

balanced set of original sample subsets $\boldsymbol{X}_S = \left\{ \boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, ..., \boldsymbol{X}^{(Q)} \right\}$, and each subset $\boldsymbol{X}^{(q)}, q = 1, 2, ..., Q$ includes $N^{'} = 2N_{min}$ samples. Based on each original sample subset $\boldsymbol{X}^{(q)}, q = 1, 2, ..., Q$, the MNEFD algorithm is implemented to obtain two kinds of structured sample subsets: the neighboring envelope sample subset $\tilde{\boldsymbol{X}}^{(q)} \in \mathbb{R}^{N \times d}$ and the neighboring cluster envelope sample subset $\boldsymbol{V}^{(q)} \in \mathbb{R}^{C \times d}$. Based on the training of $\tilde{\boldsymbol{X}}^{(q)}$ and $\boldsymbol{V}^{(q)}$, $2 \times Q$ base classifiers are obtained, and the final label $\hat{\boldsymbol{y}}$ is obtained by fusion of the predictive label matrix $\boldsymbol{E}$ through a 2D sparse fusion mechanism (2D-SFM).

4.2. Brief description of the proposed method

The main part of the proposed method (MNEFD_IE) is the envelope learning algorithm (MNEFD) which constructs two kinds of structured samples: neighboring envelope samples and neighboring cluster envelope samples. Fig. 3 shows the flowchart of the two kinds of imbalanced ensemble algorithms. Fig. 3 (a) shows the framework of the existing imbalanced ensemble algorithms. As shown in Fig. 3 (a), the kind of algorithms are single-sample based methods, since the subsets are derived from the same datasets by resampling. Fig. 3 (b) shows the framework of the proposed method, and the method is based on envelope sample. As shown in Fig. 3 (b), the proposed method is an envelope-sample based method since it generates structured samples for subsequent modeling. Fig. 3 (c) shows the flowchart of the MNEFD algorithm for constructing two kinds of envelope samples.

The main procedures are as follows: first, this algorithm uses an undersampling method to obtain Q balanced subsets. Second, based on each balanced subset, the intersample nearest neighbor relationship is mined by the manifold neighboring envelope sample reconstruction mechanism (MNESR). Each sample is enveloped with its manifold nearest neighbors to reconstruct it as the structured sample-neighboring envelope sample. Then, based on the neighboring envelope samples, a consistent fuzzy clustering algorithm (CFCMD) is designed to mine the structural information among samples and map it to the neighboring cluster envelope samples. After that, the base classifiers are separately trained on the obtained envelope samples subsets. For the Q balanced subsets, the $2 \times Q$ envelope samples subsets are obtained, so the $2 \times Q$ base classifier array is trained. Finally, for each sample, the $2Q$ classification results are obtained and fused by the 2D sparse fusion mechanism (2D-SFM).



(a)

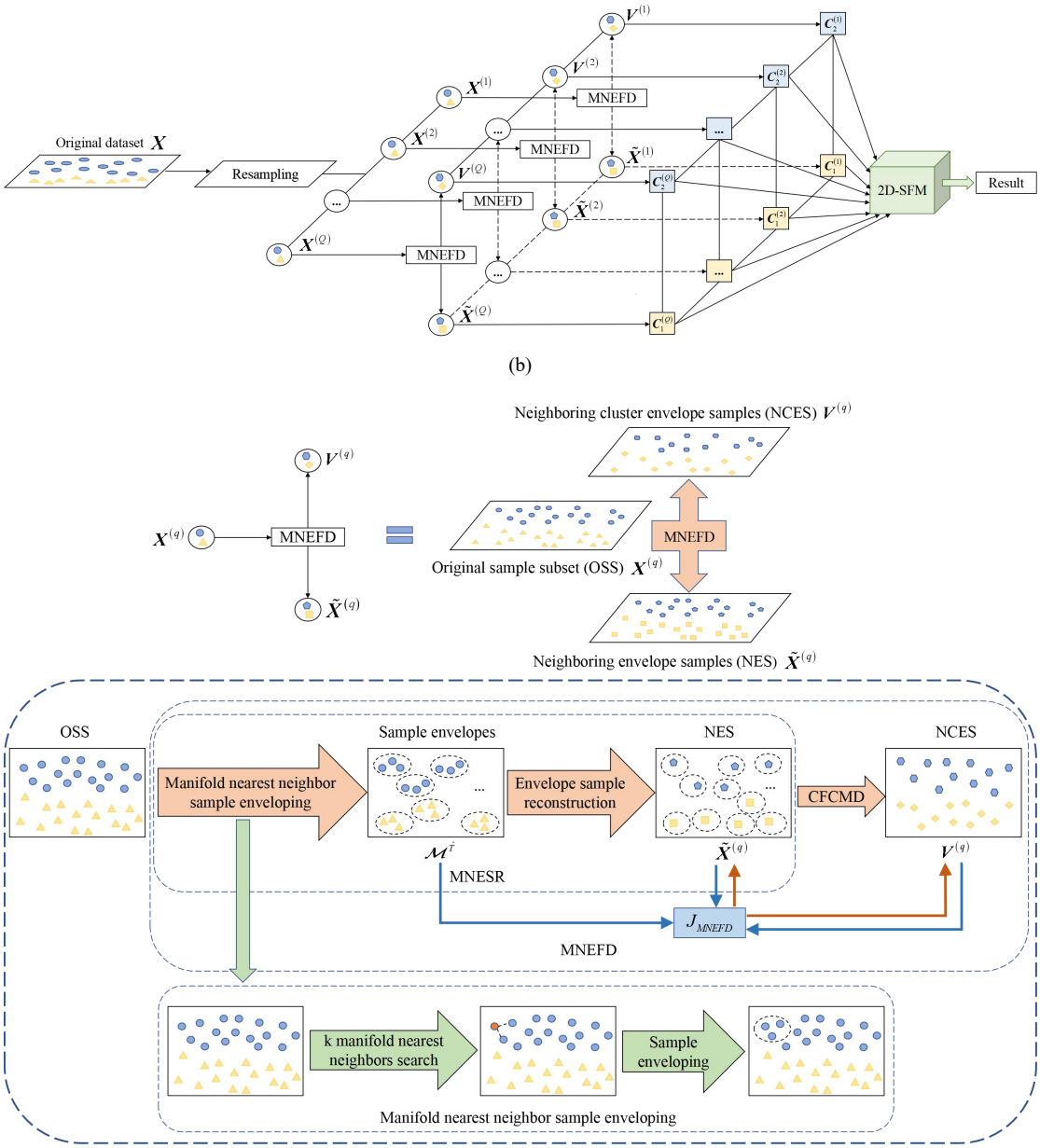


Fig. 3. Flow chart of the proposed algorithm: (a) Classical imbalanced ensemble algorithm framework; (b) The proposed algorithm (MNEFD_IE) framework; (c) The MNEFD algorithm flow chart

4.3. MNEFD

The MNEFD algorithm combines MNESR and CFCMD to explore the structural information among samples by joint optimization. MNESR aims to mine the structural relationships between the sample and its k manifold nearest neighbors and to reconstruct these structural relationships into structured sample-neighboring envelope samples by enveloping transposition projection. CFCMD aims to explore the structural information among samples and to map the hierarchical structure information to clustering centers, thereby generating neighboring cluster envelope samples.

4.3.1. MNESR-manifold neighboring envelope sample reconstruction

The MNESR conducts sample transformation by enveloping the sample and its k neighbors. Taking $k=1$ as an example, for sample $\boldsymbol{x}_i \in \mathbb{R}^d$, its nearest neighbor is $\boldsymbol{x}_j$, and their transpositions

are $\boldsymbol{x}_i^T$ and $\boldsymbol{x}_j^T$. Then, these two samples are combined to form the neighboring sample matrix (sample enveloping) $\left[\boldsymbol{x}_i^T \quad \boldsymbol{x}_j^T\right] \in \mathbb{R}^{d\times 2}$. Assuming a principal component projection based upon this, we can obtain $\left[\boldsymbol{x}_i^T \quad \boldsymbol{x}_j^T\right]\cdot \boldsymbol{P} = \left[\boldsymbol{x}_i^T \quad \boldsymbol{x}_j^T\right]\cdot\begin{bmatrix} p_{11} & p_{21} \\ p_{12} & p_{22}\end{bmatrix} = \left[p_{11}\boldsymbol{x}_i^T + p_{12}\boldsymbol{x}_j^T \quad p_{21}\boldsymbol{x}_i^T + p_{22}\boldsymbol{x}_j^T\right]$. Generally, the principal components are concentrated in the first component, so taking the first principal component, we can obtain $\tilde{\boldsymbol{x}}_1 = \left[\boldsymbol{x}_i^T \quad \boldsymbol{x}_j^T\right]\cdot\begin{bmatrix} p_{11} \\ p_{12}\end{bmatrix}$; then, the new sample is $\tilde{\boldsymbol{x}}_1 = p_{11}\boldsymbol{x}_i^T + p_{12}\boldsymbol{x}_j^T$. Therefore, it can be determined that the intersample correlation relationship can be mined by nearest neighbor enveloping. This not only helps to obtain intersample structural information but also helps to improve sample separability, as discussed above.

For distance measures, Euclidean distance is the most commonly used distance method, but when the dataset does not have a global linear structure, Euclidean distance is not a reasonable data distance measure. The topological manifold structure is generally used to measure high-dimensional nonlinear data, and this data distance metric is called the manifold distance. Based on the manifold distance, we find k nearest neighbors of each sample in the dataset and envelop the sample with the neighboring samples into a sample envelope. Based on this sample envelope, a transposition projection is performed to obtain a structured sample-neighboring envelope sample with structural information. This sample transformation process is called the manifold neighboring envelope sample reconstruction mechanism (MNESR).

The principle of MNESR to extract the structural information of similar samples around the original sample $\boldsymbol{x}_i$ can be viewed as extracting the principal sample of similar samples. This new sample not only contains most of the information of the original sample but also contains the structural information of the similar samples around the original sample $\boldsymbol{x}_i$, which is more helpful for modeling. The mathematical description is as follows.

Consider a subset $\boldsymbol{X}^{(q)}, q = 1,2,...,Q$ of $\boldsymbol{X}_S = \left\{\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, ..., \boldsymbol{X}^{(Q)}\right\}$ containing N' samples denoted as the original sample subset $\boldsymbol{X}^{(q)} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_{N'}\end{bmatrix} \in \mathbb{R}^{N'\times d}$. Based on the principle of the Isomap algorithm [38], the manifold distance between the samples is calculated.

Step 1: Computing the neighborhood graph matrix

In the input space X, the Euclidean distance between samples i, j is calculated in turn $d_{\mathrm{X}}(i,j)$ to determine which samples are neighbors on manifold M. Based on $d_{\mathrm{X}}(i,j)$, the k nearest neighbors are taken as neighbors of sample i. These neighborhood relations are represented as a weighted graph G over the samples, with edges of weight $d_{\mathrm{X}}(i,j)$ between neighboring samples.

Step 2: Computing the matrix of shortest paths

The geodesic distances (manifold distances) $d_M(i,j)$ between all sample pairs on the

manifold are estimated by computing the shortest path distances $d_G(i,j)$ in graph G. For the neighboring samples, Euclidean distance can better reflect the geodesic distance; the distant samples are approximated by adding the distances between neighboring samples to find the shortest path to the geodesic distance. Therefore, if the sample pairs i, j are linked by an edge, then $d_G(i,j) = d_X(i,j)$; otherwise, let $d_G(i,j) = \infty$, thus obtaining the weighted graph after initialization G. Based on G, $\min\{d_G(i,j), d_G(i,t) + d_G(t,j)\}, t = 1,2,...,N'$ is calculated in turn to update the shortest path of all sample pairs in $d_G(i,j)$. As the number of samples increases, the graph distances $d_G(i,j)$ can better approximate the manifold distances $d_M(i,j)$.

Step 3: Construction of the manifold neighboring sample envelope

Based on manifold distances $d_M(i,j)$, find the k nearest neighbors of sample $\boldsymbol{x}_i \in \mathbb{R}^d, i = 1,2,...,N'$; then envelop them into a manifold neighboring sample envelope $\mathcal{M}_i = \begin{bmatrix} \boldsymbol{x}_{i,1} \\ \hat{\boldsymbol{x}}_{i,2} \\ \vdots \\ \hat{\boldsymbol{x}}_{i,j} \\ \vdots \\ \hat{\boldsymbol{x}}_{i,k+1} \end{bmatrix} \in \mathbb{R}^{(k+1)\times d}$, where the subscript i indicates the sequential number of the original sample and the subscript j denotes the sequential number of the sample in the sample envelope; after that, the sample envelope is transposed as $\mathcal{M}_i^T = \begin{bmatrix} \boldsymbol{x}_{i,1}^T & \hat{\boldsymbol{x}}_{i,2}^T & ... & \hat{\boldsymbol{x}}_{i,k+1}^T \end{bmatrix} \in \mathbb{R}^{d\times(k+1)}$. Similar processing is conducted on all the original samples; then, for dataset $\boldsymbol{X}$, N' transposed manifold neighboring sample envelopes are obtained. These sample envelopes are combined by columns ($\hat{T}$) to obtain a manifold neighboring sample envelope matrix $\boldsymbol{\mathcal{M}}^{\hat{T}} = \begin{bmatrix} \mathcal{M}_1^T \\ \mathcal{M}_2^T \\ \vdots \\ \mathcal{M}_{N'}^T \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_{1,1}^T & \hat{\boldsymbol{x}}_{1,2}^T & ... & \hat{\boldsymbol{x}}_{1,k+1}^T \\ \boldsymbol{x}_{2,1}^T & \hat{\boldsymbol{x}}_{2,2}^T & ... & \hat{\boldsymbol{x}}_{2,k+1}^T \\ \vdots & \vdots & ... & \vdots \\ \boldsymbol{x}_{N',1}^T & \hat{\boldsymbol{x}}_{N',2}^T & ... & \hat{\boldsymbol{x}}_{N',k+1}^T \end{bmatrix} \in \mathbb{R}^{(N'\times d)\times(k+1)}$.

Step 4: Computing the manifold neighboring envelope sample

Centralize $\boldsymbol{\mathcal{M}}^{\hat{T}}$, i.e., $\sum_{i=1}^{N'\times d} \boldsymbol{\mathcal{M}}_i^{\hat{T}} = 0$, and then perform sample projection based on the centralized $\boldsymbol{\mathcal{M}}^{\hat{T}}$ to obtain $\tilde{\boldsymbol{\mathcal{M}}} = \boldsymbol{\mathcal{M}}^{\hat{T}} \boldsymbol{P} = \begin{bmatrix} \mathcal{M}_1^T \boldsymbol{P} \\ \mathcal{M}_2^T \boldsymbol{P} \\ \vdots \\ \mathcal{M}_{N'}^T \boldsymbol{P} \end{bmatrix} = \begin{bmatrix} \tilde{\mathcal{M}}_1 \\ \tilde{\mathcal{M}}_2 \\ \vdots \\ \tilde{\mathcal{M}}_{N'} \end{bmatrix} \in \mathbb{R}^{(N'\times d)\times 1}$, where $\tilde{\mathcal{M}}_i \in \mathbb{R}^{d\times 1}$ is the neighboring envelope sample, and $\boldsymbol{P} \in \mathbb{R}^{(k+1)\times 1}$ is the projection vector. Based on the transposition of each

envelope sample $\tilde{\mathcal{M}}_i$, we can obtain the final envelope sample $\tilde{\boldsymbol{x}}_i = \tilde{\mathcal{M}}_i^T = \boldsymbol{P}^{\mathrm{T}} \mathcal{M}_i$. Finally, the original sample subset $\boldsymbol{X}^{(q)} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_{N'} \end{bmatrix} \in \mathbb{R}^{N' \times d}$ is transformed to the neighboring envelope sample subset $\tilde{\boldsymbol{X}}^{(q)} = \begin{bmatrix} \tilde{\boldsymbol{x}}_1 \\ \tilde{\boldsymbol{x}}_2 \\ \vdots \\ \tilde{\boldsymbol{x}}_N \end{bmatrix} \in \mathbb{R}^{N' \times d}$.

To obtain the optimal $\boldsymbol{P}$, an objective function is constructed to minimize the reconstruction error as follows.

$$\begin{aligned} & J_{MNSER}(\boldsymbol{P}) = \min_{\boldsymbol{P}} \sum_{i=1}^{N'} \left\| \mathcal{M}_i^T - \tilde{\mathcal{M}}_i \boldsymbol{P}^{\mathrm{T}} \right\|_2^2 \\ & = \min_{\boldsymbol{P}} \sum_{i=1}^{N'} \left\| \mathcal{M}_i^T - \mathcal{M}_i^T \boldsymbol{P}\boldsymbol{P}^{\mathrm{T}} \right\|_2^2 \\ & = \min_{\boldsymbol{P}} \sum_{i=1}^{N'} \left\| \mathcal{M}_i - \boldsymbol{P}\boldsymbol{P}^{\mathrm{T}} \mathcal{M}_i \right\|_2^2 \\ & s.t. \boldsymbol{P}^{\mathrm{T}} \boldsymbol{P} = \boldsymbol{I} \end{aligned} \tag{1}$$

Eq. (1) can be organized to obtain Eq. (2):

$$\begin{aligned} & J_{MNSER}(\boldsymbol{P}) = \min_{\boldsymbol{P}} \sum_{i=1}^{N'} \left\| \mathcal{M}_i - \boldsymbol{P}\boldsymbol{P}^{\mathrm{T}} \mathcal{M}_i \right\|_2^2 \\ & = \min_{\boldsymbol{P}} \sum_{i=1}^{N'} \mathcal{M}_i^T \mathcal{M}_i - 2\sum_{i=1}^{N'} \mathcal{M}_i^T \boldsymbol{P}\boldsymbol{P}^{\mathrm{T}} \mathcal{M}_i + \sum_{i=1}^{N'} \mathcal{M}_i^T \boldsymbol{P}\boldsymbol{P}^{\mathrm{T}} \boldsymbol{P}\boldsymbol{P}^{\mathrm{T}} \mathcal{M}_i \\ & = \min_{\boldsymbol{P}} \sum_{i=1}^{N'} \mathcal{M}_i^T \mathcal{M}_i - \sum_{i=1}^{N'} \mathcal{M}_i^T \boldsymbol{P}\boldsymbol{P}^{\mathrm{T}} \mathcal{M}_i \\ & = \min_{\boldsymbol{P}} \sum_{i=1}^{N'} \mathcal{M}_i^T \mathcal{M}_i - tr\left[\boldsymbol{P}^{\mathrm{T}} \left(\sum_{i=1}^{N'} \mathcal{M}_i \mathcal{M}_i^T \right) \boldsymbol{P} \right] \\ & = \min_{\boldsymbol{P}} \sum_{i=1}^{N'} \mathcal{M}_i^T \mathcal{M}_i - tr\left[\boldsymbol{P}^{\mathrm{T}} \left(\boldsymbol{\mathcal{M}}^{\hat{T}} \right)^T \boldsymbol{\mathcal{M}}^{\hat{T}} \boldsymbol{P} \right] \end{aligned} \tag{2}$$

In Eq. (2), $\sum_{i=1}^{N'} \mathcal{M}_i^T \mathcal{M}$ is a constant, so Eq. (2) is equivalent to Eq. (3):

$$\begin{aligned} & J_{MNSER}(\boldsymbol{P}) = \min_{\boldsymbol{P}} - tr\left(\boldsymbol{P}^{\mathrm{T}} \left(\boldsymbol{\mathcal{M}}^{\hat{T}} \right)^T \boldsymbol{\mathcal{M}}^{\hat{T}} \boldsymbol{P} \right) \\ & s.t. \boldsymbol{P}^{\mathrm{T}} \boldsymbol{P} = \boldsymbol{I} \end{aligned} \tag{3}$$

The objective function $J_{MNSER}(\boldsymbol{P})$ can be optimized by the Lagrange multiplier method to obtain Eq. (4):

$$J_{MNSER}(\boldsymbol{P}) = -tr\left[\boldsymbol{P}^{\mathrm{T}} \left(\boldsymbol{\mathcal{M}}^{\hat{T}} \right)^T \boldsymbol{\mathcal{M}}^{\hat{T}} \boldsymbol{P} \right] + \zeta \left(\boldsymbol{P}^{\mathrm{T}} \boldsymbol{P} - \boldsymbol{I} \right) \tag{4}$$

Solve for the minimalist solution of Eq. (4) to obtain Eq. (5) :

$$\left(\boldsymbol{\mathcal{M}}^{\hat{T}} \right)^T \boldsymbol{\mathcal{M}}^{\hat{T}} \boldsymbol{P} = \zeta \boldsymbol{P} \tag{5}$$

From Eq. (5), we can solve that $\boldsymbol{P}$ is a matrix composed of the eigenvectors of $\left(\boldsymbol{\mathcal{M}}^{\hat{T}}\right)^{T}\boldsymbol{\mathcal{M}}^{\hat{T}}$ and $\boldsymbol{\zeta}$ is a diagonal matrix composed of the eigenvalues of $\left(\boldsymbol{\mathcal{M}}^{\hat{T}}\right)^{T}\boldsymbol{\mathcal{M}}^{\hat{T}}$. Therefore, when we reconstruct the envelope sample consisting of $k+1$ samples into a structured neighboring envelope sample by envelope sample projection reconstruction, we need to find the eigenvector $\boldsymbol{P}$ corresponding to the largest eigenvalue of $\left(\boldsymbol{\mathcal{M}}^{\hat{T}}\right)^{T}\boldsymbol{\mathcal{M}}^{\hat{T}}$ as the projection vector.

The whole process of the MNESR algorithm is as follows.

Algorithm 1: MNESR

Input: Original sample subset $\boldsymbol{X}^{(q)}$, Number of samples N', Number of manifold nearest neighbors k.

Procedure:

1：Computing the neighborhood graph matrix G;

2：Computing the matrix of shortest paths to approximate the manifold distances $d_M(i,j)$;

3：For i-th sample, construct its sample envelope $\mathcal{M}_i = \begin{bmatrix} \boldsymbol{x}_i \\ \hat{\boldsymbol{x}}_{i1} \\ \vdots \\ \hat{\boldsymbol{x}}_{ik} \end{bmatrix}$, and the transposed sample envelope $\mathcal{M}_i^T = \begin{bmatrix} \boldsymbol{x}_i^T & \hat{\boldsymbol{x}}_{i1}^T & \cdots & \hat{\boldsymbol{x}}_{ik}^T \end{bmatrix}$;

4：Repeat step 3 until all the samples are processed. After that, the original samples are transformed into the sample envelope matrix $\boldsymbol{\mathcal{M}}^{\hat{T}} = \begin{bmatrix} \mathcal{M}_1^T \\ \mathcal{M}_2^T \\ \vdots \\ \mathcal{M}_{N'}^T \end{bmatrix}$, and this sample envelope matrix is centralized;

5：For the sample envelope matrix $\boldsymbol{\mathcal{M}}^{\hat{T}}$, the principal samples of every sample envelope are extracted by Eqs. (1)-(5). Then, the principal samples are the reconstructed envelope samples $\tilde{\boldsymbol{X}}^{(q)} = \begin{bmatrix} \tilde{\boldsymbol{x}}_1 \\ \tilde{\boldsymbol{x}}_2 \\ \vdots \\ \tilde{\boldsymbol{x}}_{N'} \end{bmatrix} \in \mathbb{R}^{N' \times d}$;

6：Return $\tilde{\boldsymbol{X}}^{(q)}$;

Output: Neighboring envelope sample subset $\tilde{\boldsymbol{X}}^{(q)}$.

4.3.2. CFCMD-consistent fuzzy clustering algorithm

Based on the neighboring envelope sample subset $\tilde{\boldsymbol{X}}^{(q)} = \begin{bmatrix} \tilde{\boldsymbol{x}}_1 \\ \tilde{\boldsymbol{x}}_2 \\ \vdots \\ \tilde{\boldsymbol{x}}_{N'} \end{bmatrix} \in \mathbb{R}^{N' \times d}$, the clustered samples

obtained by fuzzy c-means (FCM) are denoted as $\boldsymbol{V}^{(q)}=\begin{bmatrix}\boldsymbol{v}_1\\ \boldsymbol{v}_2\\ \vdots\\ \boldsymbol{v}_C\end{bmatrix}\in\mathbb{R}^{C\times d}$. The objective function is as follows:

$$\begin{aligned} &J(\boldsymbol{U},\boldsymbol{V}^{(q)})=\min_{\boldsymbol{U},\boldsymbol{V}^{(q)}}\sum_{i=1}^{N'}\sum_{j=1}^{C}\left(u_{ij}\right)^m\left\|\tilde{\boldsymbol{x}}_i-\boldsymbol{v}_j\right\|^2 \\ &s.t.\boldsymbol{U}\mathbf{1}=\mathbf{1},\ \boldsymbol{U}\geq 0 \end{aligned} \tag{6}$$

where $\boldsymbol{U}=\begin{bmatrix}u_{11} & u_{12} & \cdots & u_{1C}\\ u_{21} & u_{22} & \cdots & u_{2C}\\ \cdots & \cdots & \cdots & \cdots\\ u_{N'1} & u_{N'2} & \cdots & u_{N'C}\end{bmatrix}\in\mathbb{R}^{N'\times C}$ is the affiliation matrix; $\boldsymbol{V}^{(q)}=\begin{bmatrix}\boldsymbol{v}_1\\ \boldsymbol{v}_2\\ \vdots\\ \boldsymbol{v}_C\end{bmatrix}\in\mathbb{R}^{C\times d}$ is the cluster center matrix; $\left\|\tilde{\boldsymbol{x}}_i-\boldsymbol{v}_j\right\|_2^2$ denotes the distance between sample $\tilde{\boldsymbol{x}}_i\in\tilde{\boldsymbol{X}}^{(q)}$ and cluster center $\boldsymbol{v}_j$ under Euclidean distance; u_{ij} refers to the affiliation value, which indicates the degree of affiliation of sample $\tilde{\boldsymbol{x}}_i$ to cluster center $\boldsymbol{v}_j$; C is the number of clusters; $m>1$ is called the fuzzification coefficient and is usually set to 2; and $\mathbf{1}$ denotes a column vector with all elements being equal to one.

However, in the process of clustering, the distribution differences between the samples before and after clustering are not considered. Therefore, to enhance the representative capability of the clusters for the input samples (neighboring envelope samples) $\tilde{\boldsymbol{X}}^{(q)}$, a mechanism is designed to maintain the consistency of the distribution differences. This mechanism is called the minimum interlayer discrepancy mechanism based on maximum mean discrepancy-MIDMD. The objective function framework can be obtained:

$$\begin{aligned} &J_{FC\&IDMD}(\boldsymbol{U},\boldsymbol{V}^{(q)})=\gamma J_{FCM}(\boldsymbol{U},\boldsymbol{V}^{(q)})+\mu J_{MIDMD}(\tilde{\boldsymbol{X}}^{(q)},\boldsymbol{V}^{(q)})\\ &=\min_{\boldsymbol{U},\boldsymbol{V}^{(q)}}\gamma\sum_{i=1}^{N'}\sum_{j=1}^{C}\left(u_{ij}\right)^m\left\|\tilde{\boldsymbol{x}}_i-\boldsymbol{v}_j\right\|^2+\mu loss\left(\mathfrak{F},\tilde{\boldsymbol{X}}^{(q)},\boldsymbol{V}^{(q)}\right)\\ &s.t.\boldsymbol{U}\mathbf{1}=\mathbf{1},\ \boldsymbol{U}\geq 0 \end{aligned} \tag{7}$$

where $loss\left(\mathfrak{F},\tilde{\boldsymbol{X}}^{(q)},\boldsymbol{V}^{(q)}\right)$ is a measure of the differences in sample distribution between $\tilde{\boldsymbol{X}}^{(q)}$ and $\boldsymbol{V}^{(q)}$, γ and μ are weights of different items, and $\mathfrak{F}$ is the set of functions that are continuous on the topological space $f:\aleph\to\mathbb{R}$.

$$loss\left(\mathfrak{F},\tilde{\boldsymbol{X}}^{(q)},\boldsymbol{V}^{(q)}\right)=\sup_{f\in\mathfrak{F}}\left(\frac{1}{N'}\sum_{k=1}^{N'}f\left(\tilde{\boldsymbol{x}}_i\right)-\frac{1}{C}\sum_{i=1}^{C}f\left(\boldsymbol{v}_j\right)\right) \tag{8}$$

We propose the unit ball in the reproducing kernel Hilbert space (RKHS) H as the MIDMD function class $\mathfrak{F}$. MIDMD is designed to determine the consistency of the distribution between $\tilde{\boldsymbol{X}}^{(q)}$ and $\boldsymbol{V}^{(q)}$, as in Eq. (9).

$$J_{MIDMD}\left(\tilde{\boldsymbol{X}}^{(q)},\boldsymbol{V}^{(q)}\right)=\left\|\frac{1}{N'}\sum_{k=1}^{N'}f\left(\tilde{\boldsymbol{x}}_i\right)-\frac{1}{C}\sum_{i=1}^{C}f\left(\boldsymbol{v}_j\right)\right\|_{\mathrm{H}}^2 \tag{9}$$

Using the characteristic kernels to construct the reproducing Hilbert space and to make the distribution between $\tilde{\boldsymbol{X}}^{(q)}$ and $\boldsymbol{V}^{(q)}$ consistent, we optimize $J_{MIDMD}\left(\tilde{\boldsymbol{X}}^{(q)},\boldsymbol{V}^{(q)}\right)$ with the following objective function.

$$J_{MIDMD}\left(\tilde{\boldsymbol{X}}^{(q)},\boldsymbol{V}^{(q)}\right)=\min_{\boldsymbol{U},\boldsymbol{V}^{(q)}}\frac{1}{N'^2}\sum_{i=1}^{N'}\sum_{i'=1}^{N'}\kappa\left(\tilde{\boldsymbol{x}}_i,\tilde{\boldsymbol{x}}_{i'}\right)-\frac{2}{N'C}\sum_{i=1}^{N'}\sum_{j=1}^{C}\kappa\left(\tilde{\boldsymbol{x}}_i,\boldsymbol{v}_j\right)+\frac{1}{C^2}\sum_{j=1}^{C}\sum_{j'=1}^{C}\kappa\left(\boldsymbol{v}_j,\boldsymbol{v}_{j'}\right) \quad (10)$$

Thus, the objective function (7) can be written as Eq. (11).

$$\begin{aligned}&J_{FC\&IDMD}(\boldsymbol{U},\boldsymbol{V}^{(q)})=\gamma J_{FCM}(\boldsymbol{U},\boldsymbol{V}^{(q)})+\mu J_{MIDMD}(\tilde{\boldsymbol{X}}^{(q)},\boldsymbol{V}^{(q)})\\&=\min_{\boldsymbol{U},\boldsymbol{V}^{(q)}}\gamma\sum_{i=1}^{N'}\sum_{j=1}^{C}\left(u_{ij}\right)^m\left\|\tilde{\boldsymbol{x}}_i-\boldsymbol{v}_j\right\|^2+\mu\left[\frac{1}{N'^2}\sum_{i=1}^{N'}\sum_{i'=1}^{N'}\kappa\left(\tilde{\boldsymbol{x}}_i,\tilde{\boldsymbol{x}}_{i'}\right)-\frac{2}{N'C}\sum_{i=1}^{N'}\sum_{j=1}^{C}\kappa\left(\tilde{\boldsymbol{x}}_i,\boldsymbol{v}_j\right)+\frac{1}{C^2}\sum_{j=1}^{C}\sum_{j'=1}^{C}\kappa\left(\boldsymbol{v}_j,\boldsymbol{v}_{j'}\right)\right]\\&s.t.\boldsymbol{U}\boldsymbol{1}=\boldsymbol{1},\ \boldsymbol{U}\geq 0\end{aligned}$$

(11)

4.3.3. MNEFD based on joint optimization of MNESR and CFCMD

To better search for the neighboring envelope sample and the neighboring cluster envelope sample, the MNESR and CFCMD are combined by joint optimization. That is, the projection vector, affiliation value, cluster center, and weights of different items $\boldsymbol{P},\boldsymbol{U},\boldsymbol{V}^{(q)},\eta,\gamma,\mu$ are optimized jointly.

(1) Objective function

The objective function of the proposed algorithm (MNEFD) can be obtained as follows.

$$\begin{aligned}&J_{MNEFD}\left(\boldsymbol{P},\boldsymbol{U},\boldsymbol{V}^{(q)},\eta,\gamma,\mu\right)=\eta J_{MNSER}\left(\boldsymbol{P}\right)+J_{FC\ \&I\ DMD}(\boldsymbol{U},\boldsymbol{V}^{(q)})\\&=\eta J_{MNSER}\left(\boldsymbol{P}\right)+\gamma J_{FCM}\left(\boldsymbol{U},\boldsymbol{V}^{(q)}\right)+\mu J_{MIDMD}\left(\tilde{\boldsymbol{X}}^{(q)},\boldsymbol{V}^{(q)}\right)\\&=\min_{\boldsymbol{P},\boldsymbol{U},\boldsymbol{V}^{(q)}}\eta\sum_{i=1}^{N'}\left\|\mathcal{M}_i-\boldsymbol{P}\boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i\right\|_2^2+\gamma\sum_{i=1}^{N'}\sum_{j=1}^{C}\left(u_{ij}\right)^m\left\|\tilde{\boldsymbol{x}}_i-\boldsymbol{v}_j\right\|^2\\&+\mu\left[\frac{1}{N'^2}\sum_{i=1}^{N'}\sum_{i'=1}^{N'}\kappa\left(\tilde{\boldsymbol{x}}_i,\tilde{\boldsymbol{x}}_{i'}\right)-\frac{2}{N'C}\sum_{i=1}^{N'}\sum_{j=1}^{C}\kappa\left(\tilde{\boldsymbol{x}}_i,\boldsymbol{v}_j\right)+\frac{1}{C^2}\sum_{j=1}^{C}\sum_{j'=1}^{C}\kappa\left(\boldsymbol{v}_j,\boldsymbol{v}_{j'}\right)\right]\end{aligned} \quad (12)$$

Since $\tilde{\boldsymbol{x}}_i=\boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i$, then the objective function can be transformed into Eq. (13):

$$\begin{aligned}&J_{MNEFD}\left(\boldsymbol{P},\boldsymbol{U},\boldsymbol{V}^{(q)},\eta,\gamma,\mu\right)=\eta J_{MNSER}\left(\boldsymbol{P}\right)+J_{FC\ \&IDMD}(\boldsymbol{P},\boldsymbol{U},\boldsymbol{V}^{(q)})\\&=\eta J_{MNSER}\left(\boldsymbol{P}\right)+\gamma J_{FCM}\left(\boldsymbol{P},\boldsymbol{U},\boldsymbol{V}^{(q)}\right)+\mu J_{MIDMD}\left(\boldsymbol{P},\boldsymbol{V}^{(q)}\right)\\&=\min_{\boldsymbol{P},\boldsymbol{U},\boldsymbol{V}^{(q)}}\eta\sum_{i=1}^{N'}\left\|\mathcal{M}_i-\boldsymbol{P}\boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i\right\|_2^2+\gamma\sum_{i=1}^{N'}\sum_{j=1}^{C}\left(u_{ij}\right)^m\left\|\boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i-\boldsymbol{v}_j\right\|_2^2\\&+\mu\left[\frac{1}{N'^2}\sum_{i=1}^{N'}\sum_{i'=1}^{N'}\kappa\left(\boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i,\boldsymbol{P}^{\mathrm{T}}\mathcal{M}_{i'}\right)-\frac{2}{N'C}\sum_{i=1}^{N'}\sum_{j=1}^{C}\kappa\left(\boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i,\boldsymbol{v}_j\right)+\frac{1}{C^2}\sum_{j=1}^{C}\sum_{j'=1}^{C}\kappa\left(\boldsymbol{v}_j,\boldsymbol{v}_{j'}\right)\right]\\&s.t.\boldsymbol{U}\boldsymbol{1}=\boldsymbol{1},\ \boldsymbol{U}\geq 0,\ \boldsymbol{P}^{\mathrm{T}}\boldsymbol{P}=\boldsymbol{I}\end{aligned} \quad (13)$$

Among them, η,γ,μ are three hyperparameters, and this paper optimizes the hyperparameters based on the grid search method. The right end of Eq. (13) consists of three parts. The first part describes the loss of the manifold neighboring envelope sample reconstruction consisting of $\mathcal{M}_i$ and $\boldsymbol{P}$. The second part describes the clustering loss measure consisting of $\boldsymbol{V}^{(q)}$ and $\boldsymbol{U}$. The third part describes the distribution differences between $\tilde{\boldsymbol{X}}^{(q)}$ and $\boldsymbol{V}^{(q)}$.

(2) Optimization

In the MNEFD model, there are three variables $\boldsymbol{P}$, $\boldsymbol{U}$ and $\boldsymbol{V}^{(q)}$ that need to be optimized, so an effective alternating variable optimization strategy can be considered to optimize the solution, i.e., to solve for one variable while fixing the rest of the variables as constants. Therefore, in solving objective function (13), $\boldsymbol{P}$, $\boldsymbol{U}$ and $\boldsymbol{V}^{(q)}$ can be solved in turn using the gradient descent method, and the optimization is described as follows.

1) Fixing $\boldsymbol{V}^{(q)}$ and $\boldsymbol{U}$ to solve $\boldsymbol{P}$.

By fixing $\boldsymbol{V}^{(q)}$ and $\boldsymbol{U}$, the problem is solved with respect to $\boldsymbol{P}$. After removing the terms unrelated to $\boldsymbol{P}$, the objective function (13) is transformed into Eq. (14).

$$\begin{aligned} J_1\left(\boldsymbol{P},\boldsymbol{U},\boldsymbol{V}^{(q)}\right) &= \min_{\boldsymbol{P}} \eta \sum_{i=1}^{N'} \left\| \mathcal{M}_i - \boldsymbol{P}\boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i \right\|_2^2 + \gamma \sum_{i=1}^{N'}\sum_{j=1}^{C} \left(u_{ij}\right)^m \left\| \boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i - \boldsymbol{v}_j \right\|_2^2 \\ &+ \mu\left[\frac{1}{N'^2} \sum_{i=1}^{N'}\sum_{i'=1}^{N'} \kappa\left(\boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i, \boldsymbol{P}^{\mathrm{T}}\mathcal{M}_{i'}\right) - \frac{2}{N'C} \sum_{i=1}^{N'}\sum_{j=1}^{C} \kappa\left(\boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i, \boldsymbol{v}_j\right) \right] + \lambda\left(\boldsymbol{P}^{\mathrm{T}}\boldsymbol{P} - \boldsymbol{I}\right) \end{aligned} \tag{14}$$

As shown in Eq. (14), it is difficult to obtain the closed-form solution of $\boldsymbol{P}$. Therefore, the gradient descent method is used to update $\boldsymbol{P}$. Then the iterative solution of $\boldsymbol{P}$ can be expressed as Eq. (15).

$$\boldsymbol{P}_{K+1} = \boldsymbol{P}_K - \theta \cdot \nabla\left(\boldsymbol{P}\right) \tag{15}$$

$$\begin{aligned} \nabla\left(\boldsymbol{P}\right) &= -2\eta \sum_{i=1}^{N'} \left[\left(\mathcal{M}_i - \boldsymbol{P}\boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i\right)\mathcal{M}_i^{\mathrm{T}}\boldsymbol{P} + \mathcal{M}_i\left(\mathcal{M}_i - \boldsymbol{P}\boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i\right)^{\mathrm{T}}\boldsymbol{P} \right] \\ &+ 2\sum_{i=1}^{N'}\sum_{j=1}^{C} \left[\gamma\left(u_{ij}\right)^m + \frac{\mu}{N'C\sigma^2} \kappa\left(\boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i, \boldsymbol{v}_j\right) \right] \mathcal{M}_i\left(\boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i - \boldsymbol{v}_j\right)^T \\ &- \frac{\mu}{N'^2\sigma^2} \sum_{i=1}^{N'}\sum_{i'=1}^{N'} \kappa\left(\boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i, \boldsymbol{P}^{\mathrm{T}}\mathcal{M}_{i'}\right)\left(\mathcal{M}_i - \mathcal{M}_{i'}\right)\left(\mathcal{M}_i - \mathcal{M}_{i'}\right)^{\mathrm{T}}\boldsymbol{P} + 2\lambda\boldsymbol{P} \end{aligned} \tag{16}$$

2) Fixing $\boldsymbol{P}$ and $\boldsymbol{V}^{(q)}$ to solve $\boldsymbol{U}$.

By fixing $\boldsymbol{P}$ and $\boldsymbol{V}^{(q)}$, the problem is solved with respect to $\boldsymbol{U}$. After removing the terms unrelated to $\boldsymbol{U}$, the objective function (13) is transformed into Eq. (17).

$$J_2(\boldsymbol{P},\boldsymbol{U},\boldsymbol{V}^{(q)}) = \min_{\boldsymbol{U}} \gamma \sum_{i=1}^{N'}\sum_{j=1}^{C} \left(u_{ij}\right)^m \left\| \boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i - \boldsymbol{v}_j \right\|_2^2 + \rho\left(\sum_{j=1}^{C} u_{ij} - 1 \right) \tag{17}$$

Solve for the minimalist solution of Eq. (17).

$$\frac{\partial J_2(\boldsymbol{P},\boldsymbol{U},\boldsymbol{V}^{(q)})}{\partial u_{ij}} = m\gamma\left(u_{ij}\right)^{m-1} \left\| \boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i - \boldsymbol{v}_j \right\|_2^2 + \rho = 0 \tag{18}$$

By calculation, the iterative formula of the affiliation matrix is obtained as follows.

$$u_{ij} = \frac{\left(1 / \left\| \boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i - \boldsymbol{v}_j \right\|_2^2\right)^{\frac{1}{m-1}}}{\sum_{w=1}^{C} \left(1 / \left\| \boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i - \boldsymbol{v}_w \right\|_2^2\right)^{\frac{1}{m-1}}} \tag{19}$$

3) Fixing $\boldsymbol{U}$ and $\boldsymbol{P}$ to solve $\boldsymbol{V}^{(q)}$.

By fixing $\boldsymbol{U}$ and $\boldsymbol{P}$, the problem is solved with respect to $\boldsymbol{V}^{(q)}$. After removing the terms unrelated to $\boldsymbol{V}^{(q)}$, the objective function (13) is transformed into Eq. (20).

$$J_3(\boldsymbol{P},\boldsymbol{U},\boldsymbol{V}^{(q)})=\min_{\boldsymbol{V}^{(q)}}\gamma\sum_{i=1}^{C}\sum_{j=1}^{N'}\left(u_{ij}\right)^m\left\|\boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i-\boldsymbol{v}_j\right\|_2^2+\mu\left[\frac{1}{C^2}\sum_{j=1}^{C}\sum_{j'=1}^{C}\kappa\left(\boldsymbol{v}_j,\boldsymbol{v}_{j'}\right)-\frac{2}{N'C}\sum_{i=1}^{N'}\sum_{j=1}^{C}\kappa\left(\boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i,\boldsymbol{v}_j\right)\right] \tag{20}$$

Based on the kernel function $\kappa(\boldsymbol{x},\boldsymbol{y})=\exp\left(-\|\boldsymbol{x}-\boldsymbol{y}\|^2/2\sigma^2\right)$, the objective function (20) is transformed into Eq. (21).

$$\frac{\partial J_3(\boldsymbol{P},\boldsymbol{U},\boldsymbol{V}^{(q)})}{\partial\boldsymbol{v}_j}=-2\sum_{i=1}^{N'}\left[\gamma\left(u_{ij}\right)^m+\frac{\mu}{N'C\sigma^2}\kappa\left(\boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i,\boldsymbol{v}_j\right)\right]\left(\boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i-\boldsymbol{v}_j\right)+\frac{2\mu}{C^2\sigma^2}\sum_{j'=1}^{C}\left(\boldsymbol{v}_{j'}-\boldsymbol{v}_j\right)\kappa\left(\boldsymbol{v}_{j'},\boldsymbol{v}_j\right) \tag{21}$$

Solve for the minimum solution of Eq. (21). $\boldsymbol{V}^{(q)}$ is obtained from Eq. (22).

$$\boldsymbol{V}^{(q)}=\boldsymbol{A}^{-1}\begin{bmatrix}b_1\\b_2\\\vdots\\b_C\end{bmatrix} \tag{22}$$

Among them:

$$\boldsymbol{A}=diag(a_1,a_2,...,a_C)+\frac{\mu}{C^2\sigma^2}\hat{\boldsymbol{K}}$$

$$\hat{\boldsymbol{K}}=\begin{bmatrix}\kappa(\boldsymbol{v}_1,\boldsymbol{v}_1) & \kappa(\boldsymbol{v}_2,\boldsymbol{v}_1) & ... & \kappa(\boldsymbol{v}_C,\boldsymbol{v}_1)\\ \kappa(\boldsymbol{v}_1,\boldsymbol{v}_2) & \kappa(\boldsymbol{v}_2,\boldsymbol{v}_2) & ... & \kappa(\boldsymbol{v}_C,\boldsymbol{v}_2)\\ ... & ... & ... & ...\\ \kappa(\boldsymbol{v}_1,\boldsymbol{v}_C) & \kappa(\boldsymbol{v}_2,\boldsymbol{v}_C) & ... & \kappa(\boldsymbol{v}_C,\boldsymbol{v}_C)\end{bmatrix}$$

$$a_w=\sum_{i=1}^{N'}\left[\gamma(u_{iw})^m+\frac{\mu}{N'C\sigma^2}\kappa\left(\boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i,\boldsymbol{v}_w\right)\right]-\frac{\mu}{C^2\sigma^2}\sum_{j=1}^{C}\kappa\left(\boldsymbol{v}_j,\boldsymbol{v}_w\right),w=1,2,...,C$$

$$b_j=\sum_{i=1}^{N'}\left[\gamma\left(u_{ij}\right)^m+\frac{\mu}{N'C\sigma^2}\kappa\left(\boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i,\boldsymbol{v}_j\right)\right]\boldsymbol{P}^{\mathrm{T}}\mathcal{M}_i,\ j=1,2,...,C$$

The whole process of the MNEFD algorithm is as follows.

Algorithm 2: MNEFD

Input: Original sample subset $\boldsymbol{X}^{(q)}$, Number of manifold nearest neighbors k, Number of clusters C, Iteration number t, Iteration threshold ε.

Procedure:

1：Based on the original sample subset $\boldsymbol{X}^{(q)}$, obtain the initialized neighboring envelope sample subset $\tilde{\boldsymbol{X}}^{(q)}$ and the initialized projection vector $\boldsymbol{P}$ by Algorithm 1 (MNESR);

2：Initialize $\boldsymbol{V}^{(q)}$ and $\boldsymbol{U}$ based on $\tilde{\boldsymbol{X}}^{(q)}$ by FCM algorithm;

3：Optimize the $\boldsymbol{U}$, $\boldsymbol{V}^{(q)}$, $\boldsymbol{P}$ by Eqs. (15), (19), and (22), respectively until $\left|J_{MNEFD}{}^{(t+1)}-J_{MNEFD}{}^{(t)}\right|<\varepsilon$;

4：Return final $\boldsymbol{U}$, $\boldsymbol{V}^{(q)}$, $\boldsymbol{P}$;

5：Based on $\boldsymbol{P}$, obtain optimized $\tilde{\boldsymbol{X}}^{(q)}$;

Output: Neighboring envelope sample subset $\tilde{\boldsymbol{X}}^{(q)}$, Neighboring cluster envelope sample subset $\boldsymbol{V}^{(q)}$.

4.4. 2D Sparse fusion mechanism (2D-SFM)

According to the envelope samples, the Q subsets of original samples are transformed to $2\times Q$ subsets of envelope samples. Therefore, the $2\times Q$ prediction results are obtained. Because these prediction results are homogeneous, they can be transformed into a fusion of $1\times 2Q$ prediction labels. Therefore, the fusion of the $2\times Q$ prediction results is transformed into a fusion of $1\times 2Q$ prediction results. The matrix composed of the prediction labels is denoted as $\boldsymbol{E}=\begin{bmatrix}\boldsymbol{e}_1\\ \boldsymbol{e}_2\\ \vdots\\ \boldsymbol{e}_{N_t}\end{bmatrix}=\begin{bmatrix}e_{11} & e_{12} & \cdots & e_{1(2Q)}\\ e_{21} & e_{22} & \cdots & e_{2(2Q)}\\ \vdots & \vdots & \cdots & \vdots\\ e_{N_t1} & e_{N_t2} & \cdots & e_{N_t(2Q)}\end{bmatrix}\in\mathbb{R}^{N_t\times 2Q}$, and N_t is the number of test samples in the test set.

The objective function of this sparse fusion mechanism is:

$$\min_{\boldsymbol{\beta}}\left(\left\|\boldsymbol{y}\text{-}\boldsymbol{E}\boldsymbol{\beta}\right\|_2^2+\mu\left\|\boldsymbol{\beta}\right\|_1\right)=\min_{\boldsymbol{\beta}}\left[\sum_{i=1}^{N_t}\left(y_i-\sum_{j=1}^{2Q}e_{ij}\beta_j\right)^2+\omega\sum_{j=1}^{2Q}\left|\beta_j\right|\right] \tag{23}$$

In Eq. (23), $\boldsymbol{y}=\begin{bmatrix}y_1\\ y_2\\ \vdots\\ y_{N_t}\end{bmatrix}\in\mathbb{R}^{N_t\times 1}$ is the true label, $\boldsymbol{\beta}=\begin{bmatrix}\beta_1\\ \beta_2\\ \vdots\\ \beta_{2Q}\end{bmatrix}\in\mathbb{R}^{2Q\times 1}$ is the weight vector, and ω is the penalty coefficient for parameter estimation. The weight of each classifier is calculated by minimizing the objective function, and the estimated weight vector is $\boldsymbol{\beta}'=\begin{bmatrix}\beta_1'\\ \beta_2'\\ \vdots\\ \beta_{2Q}'\end{bmatrix}\in\mathbb{R}^{2Q\times 1}$. Based on this weight vector, the final predicted labels are obtained by weighting and summing the labels by Eq. (24).

$$\hat{\boldsymbol{y}}=\Phi\left(\boldsymbol{E}\boldsymbol{\beta}'\right)\in\mathbb{R}^{N_t\times 1} \tag{24}$$

In Eq. (24), $\Phi\left(a\right)=\begin{cases}0, a\le 0.5\\ 1, a>0.5\end{cases}$.

Algorithm 3: MNEFD_IE

Input: Original training set $\boldsymbol{X}$, Number of minority class samples in original training set N_{min}, Number of sample subsets Q, Number of manifold nearest neighbors k, Iteration number t, Iteration threshold ε.

Procedure:

1：Based on the majority class samples in $\boldsymbol{X}$, we obtain Q majority class sample subsets with the number of samples N_{min} by random undersampling. In addition, the Q majority class sample subsets are fused with minority class samples, respectively, to obtain Q balanced original sample subsets $\boldsymbol{X}_S=\left\{\boldsymbol{X}^{(1)},\boldsymbol{X}^{(2)},...,\boldsymbol{X}^{(Q)}\right\}$;

2：For $q=1:Q$

3： Based on the original sample subset $\boldsymbol{X}^{(q)}$, obtain neighboring envelope sample subset $\tilde{\boldsymbol{X}}^{(q)}$ and neighboring cluster envelope sample subset $\boldsymbol{V}^{(q)}$ by MNEFD algorithm;

4： Train based on $\tilde{\boldsymbol{X}}^{(q)}$ to obtain classifier $C_1^{(q)}$;

5： Train based on $\boldsymbol{V}^{(q)}$ to obtain classifier $C_2^{(q)}$;

6：End

7：The prediction label matrix $\boldsymbol{E}$ is obtained by predicting the test set based on the $2Q$ classifiers;

8：The 2D sparse fusion mechanism (2D SFM) is used to obtain the final labels $\hat{\boldsymbol{y}}$;

9：Accuracy, AUC , F-measure and G-mean are obtained.

Output: Accuracy, AUC , F-measure , G-mean.

5. Time complexity analysis

The computational complexity of MNEFD_IE consists of the following four components. (1) The Q sample subsets are divided based on the random undersampling method. (2) The initialization of $\boldsymbol{P}$, $\boldsymbol{U}$, and $\boldsymbol{V}$ in the MNEFD algorithm is performed by the MNESR and FCM algorithms. (3) Iterative updating of $\boldsymbol{P}$, $\boldsymbol{U}$, and $\boldsymbol{V}$ is conducted by the MNEFD algorithm. (4) 2D sparse fusion is performed based on the prediction results.

The computational complexity of the first part is related to the number of sample subsets Q. The second part of initializing $\boldsymbol{P}$ by MNESR involves manifold distance calculation and eigendecomposition, so the complexity is $O\left(N_{min}^3\right)$. Based on the FCM algorithm to initialize $\boldsymbol{U}$ and $\boldsymbol{V}$, the complexity is $O\left(N_{min}C^2dt\right)$. N_{min} is the number of minority class samples, C is the number of clustering centers, d is the sample dimension, and t is the number of iterations in the FCM algorithm. In the third part, the complexities of updating $\boldsymbol{P}$, $\boldsymbol{U}$, and $\boldsymbol{V}$ are $O\left(N_{min}^2\right)$, $O\left(N_{min}Cd\right)$, and $O\left(C^2d\right)$, respectively. The computational complexity of the fourth part of the 2D-SFM is related to the number of sample subsets Q and the number of test samples N_t, which can be expressed as $O\left(QN_t\right)$. Assuming that the number of iterations is T, the total computational complexity of the proposed MNEFD_IE algorithm can be expressed as: $Q+Q\cdot\left(O\left(N_{min}^3\right)+O\left(N_{min}C^2dt\right)+T\cdot\left(O\left(N_{min}^2\right)+O\left(N_{min}Cd\right)+O\left(C^2d\right)\right)\right)+O\left(QN_t\right)$. It is worth mentioning that in the case of relatively high imbalance ratio, N_{min} is much smaller than the number of total samples, so the increased time cost is not significant and relatively close compared to the classical imbalanced ensemble methods.

6. Experiments

To verify the proposed algorithm (MNEFD_IE), groups of experiments and an analysis of the results were conducted. First, the experimental environment is introduced. Second, the effects of relevant parameters on the performance of the proposed algorithm are analyzed. In addition, ablation

experiments are conducted for verification of the proposed algorithm. Finally, the proposed algorithm is compared with several representative classical and state-of-the-art imbalanced ensemble algorithms.

6.1. Experimental conditions

Since most of the imbalanced ensemble algorithms choose decision trees as base classifiers, decision tree C 4.5 was chosen as the base classifier here. The 5-fold cross-validation (5-CV) method was chosen. To avoid randomness, each experiment was repeated 5 times and the mean ± standard deviation is reported.

6.1.1. Datasets

The 38 public datasets were chosen from the KEEL [39] and UCI [40] databases for study. The datasets are representative, and they have different domains, dimensions, numbers of samples, and imbalance ratios (1.82-100.14). Table 1 provides the basic information of these datasets.

Table 1. Basic information of imbalanced datasets

ID	Name	Features	Samples	Minority	Majority	Imbalance ratio
1	Glass1	9	214	76	138	1.82
2	Wisconsin	9	683	239	444	1.86
3	Pima	8	768	268	500	1.87
4	Iris0	4	150	50	100	2.00
5	Yeast1	8	1484	429	1055	2.46
6	Haberman	3	306	81	225	2.78
7	Vehicle2	18	846	218	628	2.88
8	Vehicle3	18	846	212	634	2.99
9	Glass-0-1-2-3_vs_4-5-6	9	214	51	163	3.20
10	Vehicle0	18	846	199	647	3.25
11	Ecoli1	7	336	77	259	3.36
12	Ecoli2	7	336	52	284	5.46
13	Glass6	9	214	29	185	6.38
14	Yeast3	8	1484	163	1321	8.10
15	Ecoli3	7	336	35	306	8.60
16	Yeast-2-vs-4	8	514	51	463	9.08
17	Yeast-0-5-6-7-9-vs-4	8	528	51	477	9.35
18	Glass-0-1-6_vs_2	9	192	17	175	10.29
19	Glass2	9	214	17	197	11.59
20	Yeast-1-vs-7	8	459	30	429	14.30
21	Glass4	9	214	13	201	15.47
22	Ecoli4	7	336	20	316	15.80
23	Abalone9-18	8	731	42	689	16.40
24	Shuttle-c2-vs-c4	9	129	6	123	20.50
25	Glass5	9	214	9	205	22.78
26	Yeast-2-vs-8	8	482	20	462	23.10
27	Yeast4	8	1484	51	1433	28.10
28	Winequality-red-4	11	1599	53	1506	29.17
29	Yeast-1-2-8-9-vs-7	8	947	30	917	30.57

30	Yeast5	8	1484	44	1440	32.73
31	Yeast6	8	1484	35	1449	41.40
32	Winequality-white-3_vs_7	11	900	20	880	44.00
33	Winequality-red-8_vs_6-7	11	855	18	837	46.50
34	Kr-vs-k-zero_vs_eight	6	1460	27	1433	53.07
35	Shuttle-2_vs_5	9	3316	49	3267	66.67
36	Kddcup-buffer_overflow_vs_back	41	2233	30	2203	73.43
37	Kr-vs-k-zero_vs_fifteen	6	2193	27	2166	80.22
38	Rootkit_imapvsback	41	2225	22	2203	100.14

6.1.2. Parameter setting

The important parameters of the proposed algorithm are as follows: (1) Number of sample subsets: Q . (2) Number of manifold nearest neighbors in MNESR: $MN-num$. (3) Number of clustering centers (proportion of clustering centers to the samples before clustering) in CFCMD: $C-num$. For most experiments, these three parameters are set as: $Q=10, MN-num=1, C-num=50\%$. In this paper, three hyperparameters η,γ,μ are involved, which determine the contribution of different loss items in the objective function.The range of the hyperparameters is set to $\eta,\gamma,\mu=\left[10^{-5},10^{-4},...,10^{2}\right]$.

The optimum value search is performed using the grid search method. As different value sets (η,γ,μ) are combined, the corresponding best results are selected for each dataset when executing MNEFD_IE. The parameter settings of the classical imbalanced ensemble algorithms are: Number of subsets=10, Other parameters are default.

6.1.3. Evaluation metrics and non-parametric statistical tests

In this paper, we evaluate the performance of each method based on Accuracy (ACC), AUC, F-measure (F-M), and G-mean (G-M) criteria. These evaluation metrics are calculated as follows.

$$Accuracy(ACC)=\frac{TP+TN}{TP+FP+TN+FN}$$

$$AUC=\frac{Sensitivity+Specificity}{2}$$

$$F-measure=\frac{2*Recall*Precision}{Recall+Precision}$$

$$G-mean=\sqrt{\frac{TP}{TP+FN}*\frac{TN}{TN+FP}}$$

Where TP denotes true positive, FP denotes false positive, TN denotes true negative and FN denotes false negative. In addition, sensitivity, specificity, recall, and precision are calculated as follows.

$$Sensitivity=\frac{TP}{TP+FN}$$

$$Specificity=\frac{TN}{TN+FP}$$

$$Recall=\frac{TP}{TP+FN}$$

$$Precision = \frac{TP}{TP + FP}$$

To determine whether there is a significant difference between the algorithms, we used two kinds of nonparametric statistical tests. (1). Multiple comparisons, based on the Friedman test [41] with its corresponding post hoc test to determine whether there are significant differences between all comparison algorithms. In this paper, the Holm post hoc test [42] was chosen, and the significance level was set at $\alpha = 0.05$. (2). Pairwise comparisons, wherein the Wilcoxon paired signed-rank test [43] was used to determine whether there was a significant difference in the classification ability between the two algorithms. This was complemented by the ranking of all compared algorithms with respect to different evaluation metrics based on the Friedman aligned rank test, where a lower rank number indicates better classification ability.

6.2. Parameter analysis

In this section, the influences of two important parameters: the number of manifold nearest neighbors $MN\text{-}num$ and the number of clustering centers $C\text{-}num$ on the performance of MNEFD_IE is studied. In addition, the optimization of three hyperparameters η, γ, μ is analyzed based on the grid search method.

6.2.1. Effect analysis of the number of manifold nearest neighbors

$MN\text{-}num$ is the number of manifold nearest neighbors selected based on each sample when performing the MNESR. In the proposed algorithm, $MN\text{-}num$ affects the number of samples in each sample envelope. This will affect the structural information in the neighboring envelope samples, which in turn affects the classification performance and diversity of the base classifier trained on the subsets indirectly. Therefore, to investigate the effect of $MN\text{-}num$ on the performance of MNEFD_IE, six datasets with different imbalance ratios (1.86-28.10) were selected for parametric analysis at $MN\text{-}num = 0,1,2,3,4,5$. Fig. 4 shows the four evaluation metrics based on different $MN\text{-}num$ for different datasets.

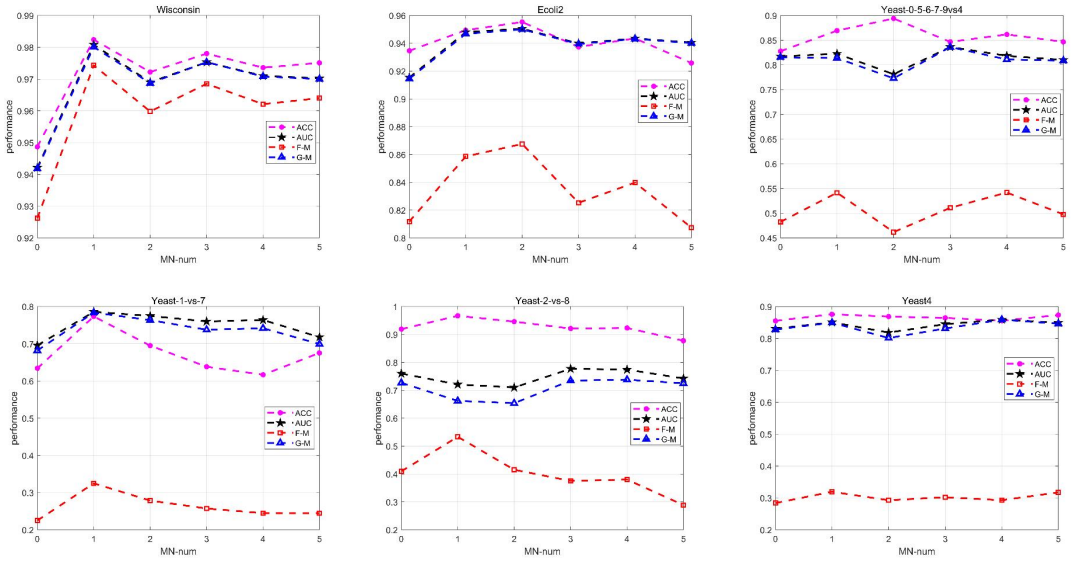


Fig. 4. MNEFD_IE performance with different $MN\text{-}num$

As shown in Fig. 4, when $MN\text{-}num$ changes from 0 to 1, each evaluation metric generally

improves to a certain extent, which shows that the introduction of a neighboring sample is effective. The possible reason for this is that the envelope sample reconstruction can effectively explore the interneighbor structure information between samples, thus improving the model's classification performance. However, as *MN-num* increases, the performance no changes or starts to worsen. Therefore, an excessive value of *MN-num* is not suitable, probably because too many selected nearest neighbors increase the redundant information. Therefore, a reasonable value of *MN-num* ranges from 1-3. To balance the accuracy and computational complexity, this paper sets *MN-num*=1 .

6.2.2. Effect analysis of the number of clusters

C-num is the ratio of the number of clusters to the number of samples before clustering when performing clustering. The smaller *C-num* is, the more compact the mined structural information will be. However, the risk of missing useful structural information will be larger. The reverse is also true.

To investigate the effect of *C-num* on the performance of MNEFD_IE, six datasets with different imbalance ratios (1.86-28.10) were selected at *C-num*=30%, 40%, 50%, 60%, 70%, 80% . Fig. 5 shows the four evaluation metrics in terms of different *C-num* and datasets.

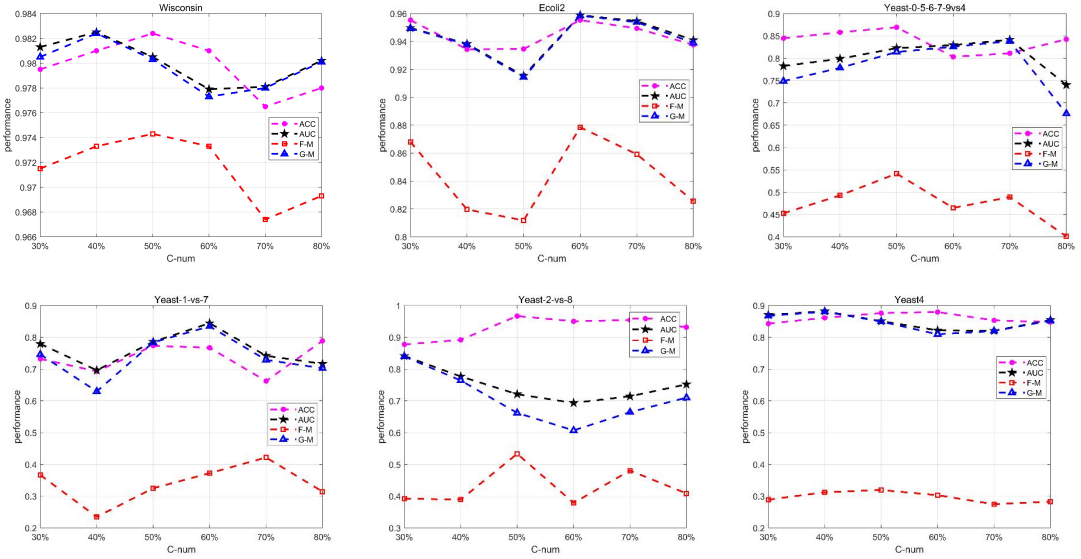


Fig. 5. MNEFD_IE performance with different *C-num*

As shown in Fig. 5, with the growth of *C-num* , the performance of the algorithm based on each evaluation metric tends to increase and then decrease, and the best performance is generally obtained when the *C-num* is approximately 50%. The *C-num* should not be too large or too small: if it is too large, some poor-quality neighboring envelope samples will be generated, and if it is too small, useful information may be lost. Therefore, a reasonable value of *C-num* should be chosen from 40% to 60%. To balance the accuracy and computational complexity, this paper sets *C-num*=50% .

6.2.3. Effect analysis of hyperparameters

The objective function of the proposed algorithm involves three hyperparameters η, γ and μ, and these three hyperparameters determine the contribution of the different loss items.

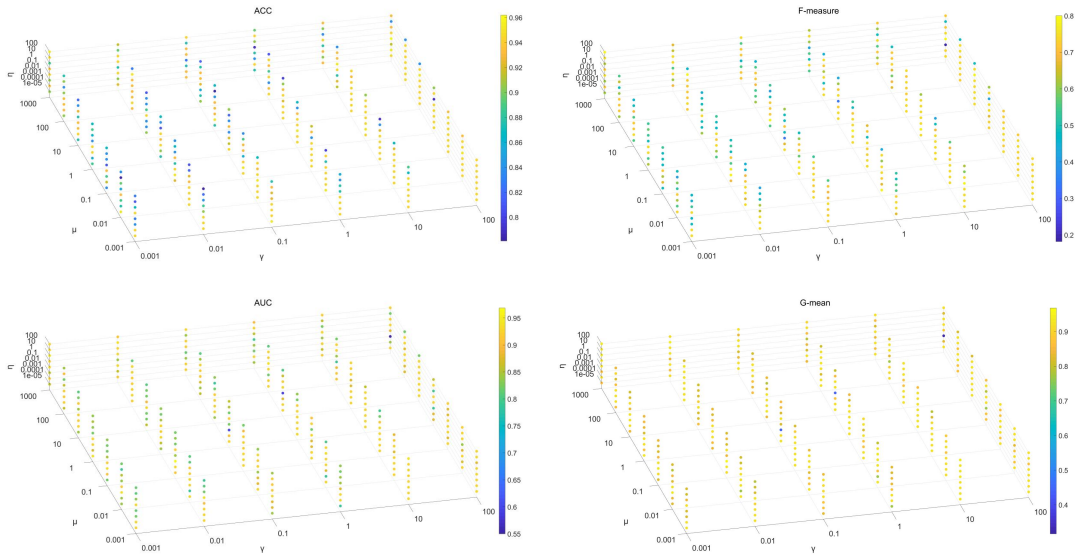


Fig. 6. MNEFD_IE performance with different (η, γ, μ)

To analyze the impacts of these hyperparameters, the performance of MNEFD_IE with different parameter value sets executed on Yeast-0-6-5-7-9-vs-4 is shown in Fig. 6. The color of each point in Fig. 6 denotes the ACC, F-M, AUC, and G-M values for the corresponding parameter values (η, γ, μ). It can be found in Fig. 6 that better performance can be produced with relatively large values of γ and μ and a moderate value of η. The values of γ and μ of approximately 10 and the value of η of approximately 1 could be potential suitable ones.

6.3. Ablation study

To verify the effectiveness of the proposed algorithm, an ablation experiment was conducted on six datasets with different imbalance ratios (1.86-28.10). The proposed algorithm is compared with the 'Random_based' and 'MNESR_based' algorithms. 'Random_based' means that the balanced subsets are divided based on random undersampling, as performed by most of the existing ensemble learning algorithms. 'MNESR_based' algorithm means that the subsets balanced by random resampling are handled by MNESR. The comparison of the above three algorithms is presented in Table 3.

Table 3. Ablation results for the proposed method

Dataset	Algorithms	ACC(%)	AUC(%)	F-M(%)	G-M(%)
Wisconsin	'Random_based'	0.9400±0.0400	0.9403±0.0377	0.9178±0.0524	0.9400±0.0377
	'MNESR_based'	0.9765±0.0168	0.9770±0.0212	0.9666±0.0249	0.9769±0.0214
	MNEFD_IE	**0.9898±0.0083**	**0.9921±0.0064**	**0.9857±0.0115**	**0.9921±0.0065**
Ecoli2	'Random_based'	0.9314±0.0482	0.9118±0.0620	0.8079±0.1261	0.9095±0.0655
	'MNESR_based'	0.9349±0.0367	0.9311±0.0512	0.8192±0.0927	0.9307±0.0514
	MNEFD_IE	**0.9761±0.0257**	**0.9538±0.0291**	**0.9242±0.0787**	**0.9532±0.0293**
Yeast-0-5-6-7-9_vs_4	'Random_based'	0.7899±0.0483	0.8497±0.0627	0.4642±0.0765	0.8453±0.0607
	'MNESR_based'	0.7916±0.0803	0.8596±0.0622	0.4811±0.1021	0.8537±0.0647
	MNEFD_IE	**0.9263±0.0039**	**0.9235±0.0779**	**0.6977±0.0370**	**0.9184±0.0873**
Yeast-1-vs-7	'Random_based'	0.7321±0.0963	0.8102±0.0401	0.3179±0.0587	0.7973±0.0409
	'MNESR_based'	0.7868±0.1449	0.7930±0.0604	0.3716±0.1134	0.7784±0.0749
	MNEFD_IE	**0.8110±0.0531**	**0.8214±0.0294**	**0.3732±0.0614**	**0.8210±0.0291**
Yeast-2-vs-8	'Random_based'	0.7907±0.0759	0.7952±0.0865	0.2543±0.0871	0.7857±0.0909
	'MNESR_based'	0.9143±0.0274	0.7460±0.0657	0.3612±0.0908	0.7201±0.0788
	MNEFD_IE	**0.9792±0.0121**	**0.8696±0.0978**	**0.7532±0.1358**	**0.8547±0.1152**
Yeast4	'Random_based'	0.8294±0.0494	0.8653±0.0551	0.2738±0.0529	0.8607±0.0547

	'MNESR_based'	0.8330±0.0611	**0.8661±0.0506**	0.2837±0.0652	**0.8645±0.0510**
	MNEFD_IE	**0.9097±0.0142**	0.8209±0.0693	**0.3553±0.0457**	0.8122±0.0787

As shown in Table 3, the performance of the MNEFD_IE is generally significantly better than that of the 'Random_based' algorithm in terms of all four evaluation metrics. This indicates that the two kinds of envelope samples are helpful for accuracy. In terms of the four evaluation metrics, 'MNESR_based' generally outperforms 'Random_based'. This means that the neighboring envelope samples can effectively mine the neighbor relationship among samples, thereby improving the classification performance. At the same time, MNEFD_IE generally achieves significant improvement compared with 'MNESR_based' in terms of four evaluation metrics. This means that the neighboring cluster envelope samples produced by CFCMD are more effective.

In addition, the diversity and performance of the base classifiers in the algorithm are also analyzed based on the Kappa-error diagram [44]. Fig. 7 shows the diversity performance of the base classifiers obtained using MNEFD_IE, SMOTE Bagging, and Under Bagging algorithms on three datasets with different imbalance ratios. Among them, 'Neighboring samples based' means that the base classifiers are trained on the neighboring envelope samples. 'Hierarchical samples based' means that the base classifiers are trained on the neighboring cluster envelope samples.

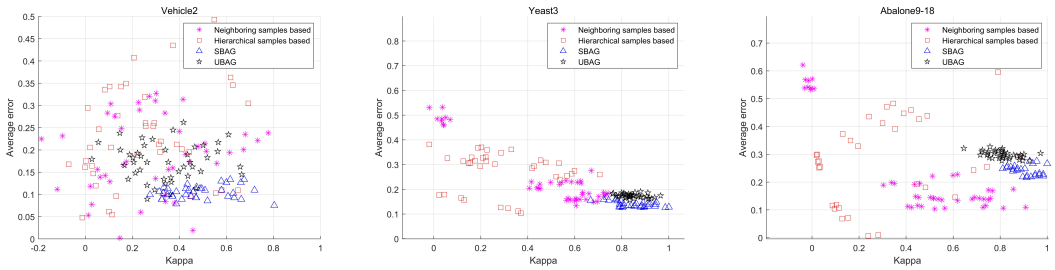


Fig. 7. Diversity and performance analysis of base classifiers

As shown in Fig. 7, MNEFD_IE can obtain data points with smaller Kappa values and average errors compared to the other two classical ensemble algorithms. This means that the base classifiers obtained by MNEFD_IE produce greater diversity and performance. Moreover, the Kappa values of data points obtained by 'Hierarchical samples based' are generally smaller than that of 'Neighboring samples based'. The possible reason for this is that the neighboring cluster envelope samples are obtained based on the neighboring envelope samples, so include the local neighboring and hierarchical structural information.

6.4. Algorithm comparison

6.4.1. Comparison with classical imbalanced ensemble algorithms

The proposed MNEFD_IE is compared with seven classical imbalanced ensemble algorithms for verification: SMOTE Bagging [32], Under Bagging [33], SMOTE Boost [34], RUSBoost [35], EUSBoost [36], Balance Cascade [37], and Easy Ensemble [37].

The details and parameter settings of the classical ensemble learning algorithms are shown in subsection 'Experimental conditions'. The results are shown in Table 4. The last row of Table 4 shows the percentage of each algorithm achieving the best performance among the compared algorithms based on 38 datasets and 4 evaluation metrics.

Table 4. Comparison with classical imbalanced ensemble algorithms

ID	Measure	SBAG	UBAG	SBO	RBO	EBO	BAC	Easy	MNEFD_IE
1	ACC	0.7526± 0.0872	0.7430± 0.0882	0.8354± 0.0596	0.7663± 0.0717	0.7860± 0.0812	0.7055± 0.0693	0.6903± 0.0926	**0.9721± 0.0104**
	AUC	0.7400± 0.0747	0.7549± 0.0817	0.7839± 0.0652	0.7724± 0.0864	0.8013± 0.0695	0.7274± 0.0738	0.7127± 0.0962	**0.9662± 0.0011**
	F-M	0.6727± 0.0763	0.6903± 0.0807	0.7232± 0.0830	0.7043± 0.0911	0.7440± 0.0683	0.6599± 0.0703	0.6468± 0.0963	**0.9599± 0.0125**
	G-M	0.7364± 0.0728	0.7508± 0.0826	0.7767± 0.0659	0.7702± 0.0838	0.7979± 0.0745	0.7218± 0.0741	0.7079± 0.0957	**0.9656± 0.0011**
2	ACC	0.9648± 0.0095	0.9582± 0.0119	0.9648± 0.0141	0.9736± 0.0083	0.9546± 0.0141	0.9657± 0.0095	0.9619± 0.0119	**0.9898± 0.0083**
	AUC	0.9611± 0.0138	0.9609± 0.0112	0.9652± 0.0145	0.9739± 0.0111	0.9526± 0.0154	0.9473± 0.0132	0.9610± 0.0117	**0.9921± 0.0064**
	F-M	0.9501± 0.0138	0.9445± 0.0137	0.9506± 0.0196	0.9627± 0.0120	0.9358± 0.0194	0.9658± 0.0095	0.9463± 0.0166	**0.9857± 0.0115**
	G-M	0.9609± 0.0139	0.9607± 0.0114	0.9651± 0.0145	0.9738± 0.0113	0.9522± 0.0157	0.9619± 0.0095	0.9610± 0.0117	**0.9921± 0.0065**
3	ACC	0.7590± 0.0158	0.7214± 0.0408	0.7408± 0.0356	0.7356± 0.0309	0.7792± 0.0480	0.6901± 0.0307	0.7143± 0.0306	**0.8243± 0.0302**
	AUC	0.7274± 0.0155	0.7383± 0.0352	0.7291± 0.0529	0.7293± 0.0337	0.7566± 0.0392	0.7020± 0.0256	0.7124± 0.0423	**0.7954± 0.0343**
	F-M	0.6434± 0.0212	0.6666± 0.0356	0.6469± 0.0697	0.6511± 0.0414	0.6852± 0.0437	0.6252± 0.0277	0.6344± 0.0512	**0.7327± 0.0494**
	G-M	0.7196± 0.0165	0.7355± 0.0363	0.7257± 0.0575	0.7278± 0.0342	0.7541± 0.0379	0.6982± 0.0258	0.7114± 0.0423	**0.7823± 0.0460**
4	ACC	0.9866± 0.0182	0.9866± 0.0182	0.9933± 0.0149	0.9933± 0.0149	0.9933± 0.0149	**1±0**	0.9933± 0.0149	**1±0**
	AUC	0.9800± 0.0273	0.9800± 0.0273	0.9900± 0.0223	0.9900± 0.0223	0.9900± 0.0223	**1±0**	0.9900± 0.0223	**1±0**
	F-M	0.9789± 0.0288	0.9789± 0.0288	0.9894± 0.0235	0.9894± 0.0235	0.9894± 0.0235	**1±0**	0.9894± 0.0235	**1±0**
	G-M	0.9794± 0.0281	0.9794± 0.0281	0.9897± 0.0229	0.9897± 0.0229	0.9897± 0.0229	**1±0**	0.9897± 0.0229	**1±0**
5	ACC	0.7030± 0.0246	0.7263± 0.0239	0.7183± 0.0336	0.7398± 0.0253	0.6913± 0.0551	0.6440± 0.0433	0.6543± 0.0207	**0.7399± 0.0516**
	AUC	0.6924± 0.0239	0.7210± 0.0347	0.6946± 0.0280	0.7104± 0.0400	0.7081± 0.0307	0.6794± 0.0080	0.6752± 0.0144	**0.7397± 0.0239**
	F-M	0.5621± 0.0347	0.5989± 0.0418	0.5648± 0.0401	0.5864± 0.0522	0.5857± 0.0394	0.5885± 0.0456	0.5481± 0.0146	**0.6246± 0.0357**
	G-M	0.6855± 0.0331	0.7204± 0.0351	0.6865± 0.0366	0.7066± 0.0434	0.7032± 0.0304	0.6696± 0.0192	0.6728± 0.0156	**0.7366± 0.0211**
6	ACC	0.6532± 0.0654	0.6598± 0.0565	0.6370± 0.0473	0.6575± 0.0641	0.7018± 0.0427	0.6172± 0.0830	0.6843± 0.0758	**0.9905± 0.0130**

	AUC	0.6304± 0.0521	0.6422± 0.0476	0.6145± 0.0540	0.6639± 0.0376	0.6548± 0.0647	0.6210± 0.0920	0.6008± 0.1185	**0.9947± 0.0072**
	F-M	0.4698± 0.0637	0.4861± 0.0640	0.4517± 0.0698	0.5076± 0.0488	0.4901± 0.0849	0.6581± 0.1180	0.3728± 0.2543	**0.9556± 0.0609**
	G-M	0.6208± 0.0463	0.6188± 0.0802	0.6096± 0.0572	0.6575± 0.0382	0.6375± 0.0719	0.6189± 0.0919	0.5165± 0.2490	**0.9947± 0.0073**
7	ACC	0.9621± 0.0130	0.9566± 0.0179	0.9657± 0.0188	0.9621± 0.0184	**0.9704± 0.0200**	0.9515± 0.0153	0.9455± 0.0211	0.9503± 0.0456
	AUC	0.9525± 0.0245	0.9536± 0.0082	0.9661± 0.0140	0.9671± 0.0206	**0.9681± 0.0210**	0.9539± 0.0050	0.9559± 0.0176	0.9606± 0.0313
	F-M	0.9270± 0.0261	0.9235± 0.0320	0.9328± 0.0375	0.9304± 0.0340	**0.9442± 0.0373**	0.9114± 0.0241	0.9031± 0.0352	0.9148± 0.0694
	G-M	0.9532± 0.0244	0.9599± 0.0175	0.9660± 0.0140	0.9669± 0.0206	**0.9680± 0.0212**	0.9536± 0.0051	0.9555± 0.0178	0.9599± 0.0325
8	ACC	0.7677± 0.0092	0.7411± 0.0234	0.7718± 0.0287	0.7564± 0.0295	0.7494± 0.0343	0.7055± 0.0446	0.7257± 0.0351	**0.8120± 0.0383**
	AUC	0.7618± 0.0170	0.7768± 0.0235	0.7441± 0.0348	0.7493± 0.0545	**0.7902± 0.0315**	0.7360± 0.0400	0.7386± 0.0360	0.7870± 0.0509
	F-M	0.6190± 0.0195	0.6213± 0.0265	0.6015± 0.0423	0.5993± 0.0643	0.6364± 0.0377	0.5765± 0.0459	0.5830± 0.0420	**0.6601± 0.0586**
	G-M	0.7613± 0.0177	0.7720± 0.0219	0.7402± 0.0386	0.7464± 0.0584	0.7854± 0.0312	0.7314± 0.0391	0.7377± 0.0362	**0.7785± 0.0612**
9	ACC	0.9203± 0.0487	0.8831± 0.0520	0.9392± 0.0390	0.9203± 0.0428	0.9161± 0.0533	0.9157± 0.0359	0.9219± 0.0268	**0.9953± 0.0104**
	AUC	0.9198± 0.0466	0.8885± 0.0503	0.9185± 0.0514	0.9269± 0.0324	0.9101± 0.0514	0.8892± 0.0517	0.9171± 0.0313	**0.9970± 0.0068**
	F-M	0.8509± 0.0872	0.7903± 0.0863	0.8730± 0.0807	0.8529± 0.0740	0.8414± 0.0880	0.8255± 0.0715	0.8536± 0.0521	**0.9905± 0.0213**
	G-M	0.9183± 0.0465	0.8862± 0.0491	0.9171± 0.0524	0.9258± 0.0322	0.9078± 0.0522	0.9054± 0.0435	0.9159± 0.0319	**0.9969± 0.0068**
10	ACC	0.9337± 0.0215	0.9349± 0.0131	0.9396± 0.0252	0.9550± 0.0106	0.9397± 0.0234	0.9255± 0.0239	0.9302± 0.0224	**0.9657± 0.0190**
	AUC	0.9359± 0.0308	0.9523± 0.0189	0.9311± 0.0385	0.9618± 0.0138	0.9466± 0.0279	0.9356± 0.0223	0.9370± 0.0312	**0.9655± 0.0118**
	F-M	0.8703± 0.0416	0.8769± 0.0251	0.8768± 0.0519	0.9108± 0.0206	0.8827± 0.0440	0.8588± 0.0393	0.8652± 0.0418	**0.9311± 0.0349**
	G-M	0.9355± 0.0311	0.9516± 0.0187	0.9305± 0.0389	0.9617± 0.0137	0.9465± 0.0278	0.9353± 0.0225	0.9365± 0.0315	**0.9652± 0.0119**
11	ACC	0.8814± 0.0376	0.8721± 0.0540	0.8840± 0.0436	0.8839± 0.0266	0.8750± 0.0734	0.8600± 0.0584	0.8481± 0.0611	**0.9910± 0.0082**
	AUC	0.8998± 0.0196	0.8990± 0.0501	0.8474± 0.0587	0.9068± 0.0290	0.8918± 0.0578	0.8637± 0.0439	0.8698± 0.0564	**0.9800± 0.0183**
	F-M	0.7865± 0.0489	0.7777± 0.0761	0.7562± 0.0916	0.7908± 0.0348	0.7820± 0.0953	0.7475± 0.0773	0.7383± 0.0855	**0.9793± 0.0189**

	G-M	0.8975± 0.0187	0.8961± 0.0510	0.8436± 0.0621	0.9042± 0.0289	0.8878± 0.0616	0.8616± 0.0434	0.8666± 0.0570	**0.9797± 0.0186**
12	ACC	0.9168± 0.0379	0.8929± 0.0408	0.9346± 0.0301	0.9046± 0.0252	0.8990± 0.0570	0.8539± 0.0376	0.8212± 0.0427	**0.9761± 0.0257**
	AUC	0.8926± 0.0721	0.8899± 0.0353	0.9090± 0.0786	0.8905± 0.0332	0.8862± 0.0639	0.8670± 0.0341	0.8485± 0.0298	**0.9538± 0.0291**
	F-M	0.7606± 0.1042	0.7247± 0.0834	0.8023± 0.0962	0.7408± 0.0364	0.7357± 0.1204	0.6552± 0.0653	0.6095± 0.0525	**0.9242± 0.0787**
	G-M	0.8870± 0.0744	0.8897± 0.0353	0.9054± 0.0842	0.8878± 0.0356	0.8857± 0.0645	0.8653± 0.0332	0.8459± 0.0284	**0.9532± 0.0293**
13	ACC	0.9346± 0.0191	0.8972± 0.0421	0.9345± 0.0106	0.9108± 0.0518	0.8877± 0.0894	0.8875± 0.0395	0.8550± 0.0387	**0.9766± 0.0233**
	AUC	0.8923± 0.0965	0.9159± 0.0576	0.8504± 0.0815	0.9227± 0.0342	0.8932± 0.0700	0.8932± 0.0359	0.8463± 0.0935	**0.9586± 0.0709**
	F-M	0.7657± 0.1005	0.7159± 0.0953	0.7447± 0.0615	0.7507± 0.1082	0.7121± 0.1598	0.6893± 0.0791	0.6050± 0.1065	**0.9132± 0.0881**
	G-M	0.8836± 0.1123	0.9115± 0.0553	0.8353± 0.0970	0.9182± 0.0340	0.8912± 0.0701	0.8913± 0.0352	0.8390± 0.1036	**0.9551± 0.0783**
14	ACC	0.9413± 0.0218	0.9279± 0.0258	0.9386± 0.0086	0.9225± 0.0263	0.9198± 0.0252	0.9076± 0.0257	0.9130± 0.0222	**0.9905± 0.0130**
	AUC	0.9401± 0.0197	0.9353± 0.0155	0.8795± 0.0148	0.9188± 0.0240	0.9281± 0.0166	0.9079± 0.0316	0.9109± 0.0274	**0.9947± 0.0072**
	F-M	0.7822± 0.0661	0.7472± 0.0673	0.7427± 0.0290	0.7260± 0.0698	0.7244± 0.0597	0.6871± 0.0697	0.6989± 0.0580	**0.9556± 0.0609**
	G-M	0.9401± 0.0197	0.9350± 0.0156	0.8761± 0.0157	0.9188± 0.0240	0.9277± 0.0168	0.9079± 0.0316	0.9108± 0.0274	**0.9947± 0.0073**
15	ACC	0.8898± 0.0227	0.8424± 0.0520	0.9107± 0.0179	0.8720± 0.0522	0.8621± 0.0174	0.8090± 0.1216	0.8451± 0.0822	**0.9628± 0.0584**
	AUC	0.8754± 0.0467	0.8994± 0.0386	0.8744± 0.0735	0.8402± 0.0332	0.8605± 0.0746	0.8112± 0.0749	0.8504± 0.0463	**0.9442± 0.0516**
	F-M	0.6197± 0.0525	0.5710± 0.0843	0.6563± 0.0814	0.5778± 0.0686	0.5621± 0.0549	0.4820± 0.1089	0.5544± 0.0845	**0.8806± 0.1533**
	G-M	0.8736± 0.0481	0.8955± 0.0396	0.8693± 0.0823	0.8346± 0.0337	0.8552± 0.0810	0.7996± 0.0831	0.8439± 0.0493	**0.9413± 0.0544**
16	ACC	0.9163± 0.0108	0.9124± 0.0305	0.9435± 0.0187	0.9299± 0.0371	0.9144± 0.0510	0.9066± 0.0520	0.9046± 0.0221	**0.9845± 0.0128**
	AUC	0.8910± 0.0753	0.9335± 0.0229	0.8893± 0.0455	0.9263± 0.0296	0.9267± 0.0196	0.8866± 0.0358	0.9212± 0.0433	**0.9833± 0.0251**
	F-M	0.6664± 0.0600	0.6914± 0.0662	0.7428± 0.0829	0.7342± 0.0819	0.7024± 0.1029	0.6632± 0.0875	0.6642± 0.0618	**0.9286± 0.0532**
	G-M	0.8861± 0.0830	0.9324± 0.0229	0.9058± 0.0581	0.9247± 0.0309	0.9253± 0.0209	0.8827± 0.0396	0.9199± 0.0435	**0.9832± 0.0253**
17	ACC	0.8541± 0.0340	0.7916± 0.0179	0.8900± 0.0321	0.8351± 0.0295	0.8180± 0.0324	0.7557± 0.0313	0.8047± 0.0516	**0.9263± 0.0039**

	AUC	0.8217± 0.0822	0.7969± 0.0385	0.7727± 0.0907	0.7862± 0.0899	0.8206± 0.0471	0.7511± 0.0658	0.7801± 0.0640	**0.9235± 0.0779**
	F-M	0.5089± 0.1030	0.4271± 0.0403	0.5235± 0.1359	0.4581± 0.0987	0.4692± 0.0679	0.3686± 0.0432	0.4298± 0.0675	**0.6977± 0.0370**
	G-M	0.8165± 0.0885	0.7963± 0.0386	0.7523± 0.1114	0.7788± 0.0982	0.8197± 0.0473	0.7456± 0.0636	0.7757± 0.0693	**0.9184± 0.0873**
18	ACC	0.8331± 0.0475	0.6667± 0.0261	0.8491± 0.0423	0.7809± 0.0607	0.7182± 0.0896	0.6300± 0.1074	0.6345± 0.1013	**0.9795± 0.0115**
	AUC	0.6676± 0.1698	0.7035± 0.1160	0.6014± 0.1304	0.6999± 0.1416	0.7930± 0.1055	0.6254± 0.1762	0.7390± 0.1278	**0.9000± 0.0559**
	F-M	0.4020± 0.1666	0.2771± 0.0862	0.3234± 0.1658	0.3196± 0.1500	0.3669± 0.1032	0.2199± 0.0984	0.3066± 0.1166	**0.8857± 0.0639**
	G-M	0.5605± 0.3434	0.6863± 0.1213	0.4569± 0.2871	0.6692± 0.1733	0.7851± 0.1015	0.5965± 0.1821	0.7259± 0.1226	**0.8928± 0.0599**
19	ACC	0.8174± 0.0518	0.5888± 0.0579	0.8594± 0.0585	0.7662± 0.1562	0.7053± 0.0626	0.5750± 0.1166	0.5933± 0.1077	**0.9814± 0.0104**
	AUC	0.7019± 0.1399	0.7766± 0.0316	0.7330± 0.0945	0.6375± 0.1506	0.7646± 0.0711	0.6174± 0.1515	0.7795± 0.0565	**0.8607± 0.0745**
	F-M	0.3847± 0.1180	0.2796± 0.0375	0.4141± 0.1420	0.3190± 0.2350	0.3129± 0.0714	0.2064± 0.0854	0.2903± 0.0845	**0.8400± 0.0894**
	G-M	0.6077± 0.3421	0.7428± 0.0422	0.7114± 0.1150	0.6110± 0.1610	0.7569± 0.0638	0.6134± 0.1478	0.7445± 0.0766	**0.8532± 0.0821**
20	ACC	0.8237± 0.0556	0.7363± 0.0262	0.8065± 0.0742	**0.8257± 0.0558**	0.7996± 0.0336	0.6404± 0.0785	0.6817± 0.0821	0.8110± 0.0531
	AUC	0.7208± 0.0367	0.6883± 0.0798	0.7259± 0.0319	0.7827± 0.0943	0.7222± 0.1005	0.6526± 0.1016	0.7212± 0.0346	**0.8214± 0.0294**
	F-M	0.3174± 0.0480	0.2361± 0.0525	0.3142± 0.0662	0.3601± 0.0813	0.2946± 0.0899	0.1974± 0.0634	0.2425± 0.0201	**0.3732± 0.0614**
	G-M	0.7074± 0.0455	0.6778± 0.0981	0.7170± 0.0356	0.7749± 0.1005	0.7105± 0.1192	0.6425± 0.1109	0.7113± 0.0416	**0.8210± 0.0291**
21	ACC	0.9060± 0.0610	0.8773± 0.0577	0.9343± 0.0355	0.9018± 0.0665	0.8975± 0.1390	0.8036± 0.0881	0.8693± 0.0706	**0.9902± 0.0134**
	AUC	0.9025± 0.0885	0.8593± 0.1031	0.8866± 0.1011	0.9004± 0.1392	0.8666± 0.1520	0.8480± 0.1249	0.8208± 0.1130	**0.9949± 0.0070**
	F-M	0.5733± 0.1382	0.4666± 0.1027	0.6189± 0.1166	0.5563± 0.2347	0.6187± 0.2352	0.3915± 0.2176	0.4389± 0.1094	**0.9200± 0.1095**
	G-M	0.8921± 0.1066	0.8487± 0.1153	0.8749± 0.1178	0.8962± 0.1469	0.8578± 0.1610	0.8393± 0.1301	0.8108± 0.1214	**0.9948± 0.0071**
22	ACC	0.9524± 0.0193	0.8363± 0.0802	0.9582± 0.0164	0.8927± 0.0867	0.8600± 0.1016	0.8539± 0.0492	0.8869± 0.0248	**0.9911± 0.0081**
	AUC	0.8810± 0.0497	0.8427± 0.0773	0.8373± 0.0964	0.8961± 0.0714	0.9021± 0.1029	0.8989± 0.0593	0.8930± 0.0531	**0.9953± 0.0043**
	F-M	0.7463± 0.1452	0.4118± 0.1298	0.6666± 0.1020	0.5680± 0.2540	0.5121± 0.2370	0.4491± 0.1020	0.4888± 0.0248	**0.9333± 0.0609**

	G-M	0.8753± 0.0523	0.8394± 0.0756	0.8177± 0.1151	0.8920± 0.0737	0.9005± 0.1026	0.8956± 0.0596	0.8897± 0.0541	**0.9952± 0.0043**
23	ACC	0.8617± 0.0601	0.7369± 0.0566	0.9137± 0.0458	0.8002± 0.0353	0.7523± 0.0597	0.7358± 0.0713	0.6880± 0.0665	**0.9623± 0.0048**
	AUC	0.7257± 0.1537	0.7819± 0.0526	0.7533± 0.1430	0.7906± 0.0992	0.7326± 0.0941	0.7502± 0.1094	0.7220± 0.0595	**0.9212± 0.0807**
	F-M	0.3392± 0.1904	0.2755± 0.0453	0.4560± 0.2876	0.3146± 0.0807	0.2504± 0.0572	0.2628± 0.0932	0.2257± 0.0444	**0.7166± 0.0151**
	G-M	0.6932± 0.1850	0.7770± 0.0536	0.7149± 0.1897	0.7834± 0.1052	0.7196± 0.1073	0.7463± 0.1094	0.7165± 0.0586	**0.9173± 0.0859**
24	ACC	0.9221± 0.0109	0.9384± 0.0842	0.9340± 0.0241	0.9692± 0.0688	**1±0**	**1±0**	**1±0**	**1±0**
	AUC	0.9558± 0.0077	0.9473± 0.0447	0.9114± 0.0904	0.9840± 0.0357	**1±0**	**1±0**	**1±0**	**1±0**
	F-M	0.6334± 0.0383	0.7666± 0.3248	0.6269± 0.0362	0.8666± 0.2981	**1±0**	**1±0**	**1±0**	**1±0**
	G-M	0.9551± 0.0076	0.9658± 0.0467	0.9027± 0.1095	0.9833± 0.0373	**1±0**	**1±0**	**1±0**	**1±0**
25	ACC	0.9488± 0.0382	0.9495± 0.0310	0.9673± 0.0126	0.8976± 0.0879	0.9297± 0.0572	0.9018± 0.0476	0.9390± 0.0538	**0.9884± 0.0164**
	AUC	0.9256± 0.0998	0.5266± 0.2712	0.9353± 0.1037	0.9463± 0.0461	0.9634± 0.0298	0.9487± 0.0249	0.9682± 0.0280	**0.9939± 0.0086**
	F-M	0.6619± 0.1980	0.9512± 0.0298	0.6933± 0.0596	0.5815± 0.3015	0.6066± 0.2832	0.4866± 0.1849	0.6266± 0.2385	**0.9000± 0.1414**
	G-M	0.9164± 0.1185	0.9065± 0.0569	0.9266± 0.1228	0.9438± 0.0487	0.9623± 0.0310	0.9471± 0.0261	0.9674± 0.0292	**0.9939± 0.0087**
26	ACC	0.9584± 0.0181	0.7590± 0.1437	0.9646± 0.0174	0.9335± 0.0059	0.7820± 0.0375	0.6823± 0.0972	0.7637± 0.0961	**0.9792± 0.0121**
	AUC	0.8109± 0.1566	0.7307± 0.1349	0.7902± 0.1070	0.8457± 0.0030	0.7666± 0.0844	0.7625± 0.1149	0.7571± 0.1331	**0.8696± 0.0978**
	F-M	0.5473± 0.1410	0.2327± 0.1257	0.5833± 0.1666	0.4846± 0.0210	0.2238± 0.0514	0.1910± 0.0758	0.2214± 0.0736	**0.7532± 0.1358**
	G-M	0.7697± 0.2034	0.7215± 0.1378	0.7528± 0.1564	0.8403± 0.0027	0.7613± 0.0878	0.7485± 0.1142	0.7471± 0.1340	**0.8547± 0.1152**
27	ACC	0.7978± 0.0482	0.7742± 0.0275	0.8215± 0.0885	0.8200± 0.0347	0.8126± 0.0403	0.7850± 0.0532	0.7998± 0.0588	**0.9097± 0.0142**
	AUC	0.8339± 0.0374	**0.8452± 0.0970**	0.7979± 0.0779	0.8199± 0.0472	0.8372± 0.0611	0.7747± 0.0966	0.8191± 0.0865	0.8209± 0.0693
	F-M	0.2345± 0.0278	0.2097± 0.0219	0.2466± 0.0488	0.2398± 0.0170	0.2437± 0.0397	0.2031± 0.0592	0.2337± 0.0750	**0.3553± 0.0457**
	G-M	0.8307± 0.0327	0.8179± 0.0717	0.7835± 0.0916	0.8165± 0.0462	**0.8340± 0.0594**	0.7710± 0.1002	0.8159± 0.0880	0.8122± 0.0781
28	ACC	0.8930± 0.0156	0.6897± 0.0454	0.9392± 0.0047	0.8116± 0.0324	0.7535± 0.0876	0.6113± 0.0831	0.6611± 0.1143	**0.9677± 0.0018**

	AUC	0.6523± 0.0732	**0.6666± 0.0750**	0.5867± 0.0582	0.6228± 0.0453	0.6110± 0.0788	0.6500± 0.0791	0.6403± 0.0669	0.6034± 0.1003
	F-M	0.1970± 0.0716	0.1206± 0.0285	0.2180± 0.0403	0.1330± 0.0335	0.1051± 0.0124	0.1021± 0.0186	0.1083± 0.0217	**0.2699± 0.1787**
	G-M	0.5921± 0.1039	**0.6600± 0.0797**	0.3999± 0.2266	0.5879± 0.0552	0.5630± 0.1616	0.6335± 0.0727	0.6206± 0.0824	0.4239± 0.2120
29	ACC	0.8253± 0.0177	0.7285± 0.0661	0.8171± 0.0096	0.8456± 0.0239	0.7696± 0.0990	0.6472± 0.0484	0.5957± 0.1497	0.8468± 0.0976
	AUC	0.6711± 0.1485	0.7147± 0.0517	0.7299± 0.1313	0.7389± 0.0896	0.6714± 0.0611	0.6727± 0.0260	0.6783± 0.0562	**0.8000± 0.0066**
	F-M	0.1498± 0.0788	0.1423± 0.0212	0.1851± 0.0714	0.1992± 0.0365	0.1440± 0.0471	0.1122± 0.0093	0.1130± 0.0231	0.2628± 0.0997
	G-M	0.6200± 0.2011	0.7089± 0.0561	0.7065± 0.1625	0.7191± 0.1208	0.6537± 0.0801	0.6702± 0.0229	0.6591± 0.0712	**0.7954± 0.0137**
30	ACC	0.9703± 0.0122	0.9393± 0.0189	**0.9757± 0.0087**	0.9440± 0.1475	0.9602± 0.0096	0.9333± 0.0197	0.9198± 0.0191	0.9636± 0.0140
	AUC	**0.9618± 0.0320**	0.9458± 0.0293	0.9444± 0.0555	0.9604± 0.0282	0.9458± 0.0308	0.9534± 0.0220	0.9479± 0.0223	0.9597± 0.0291
	F-M	0.6669± 0.0925	0.4917± 0.0811	**0.7018± 0.0541**	0.5158± 0.0678	0.5849± 0.0624	0.4736± 0.0642	0.4253± 0.0464	0.6231± 0.0788
	G-M	**0.9612± 0.0325**	0.9451± 0.0294	0.9418± 0.0589	0.9600± 0.0282	0.9452± 0.0312	0.9526± 0.0221	0.9470± 0.0222	0.9592± 0.0295
31	ACC	0.9521± 0.0114	0.8679± 0.0335	**0.9676± 0.0084**	0.8746± 0.0377	0.8712± 0.0517	0.7776± 0.0334	0.7755± 0.0493	0.9137± 0.0173
	AUC	0.8360± 0.1076	0.8766± 0.0594	0.7743± 0.1489	0.8382± 0.0795	0.8225± 0.1086	0.8162± 0.0426	0.8293± 0.0513	**0.8868± 0.0170**
	F-M	0.4103± 0.0799	0.2466± 0.0502	**0.4407± 0.1667**	0.2353± 0.0390	0.2206± 0.0575	0.1548± 0.0158	0.1603± 0.0266	0.3234± 0.0472
	G-M	0.8179± 0.1288	0.8746± 0.0593	0.7252± 0.1973	0.8307± 0.0848	0.8135± 0.1236	0.8135± 0.0405	0.8246± 0.0488	**0.8861± 0.0175**
32	ACC	0.9333± 0.0143	0.7411± 0.0529	0.9518± 0.0032	0.7866± 0.0873	0.7688± 0.0550	0.7069± 0.0517	0.5966± 0.1318	**0.9889± 0.0045**
	AUC	0.6605± 0.0675	0.7698± 0.0618	0.7310± 0.1207	0.7198± 0.1365	0.7107± 0.1565	0.7279± 0.1075	0.6715± 0.0420	**0.8111± 0.0722**
	F-M	0.1980± 0.0557	0.1237± 0.0255	0.3025± 0.1000	0.1347± 0.0713	0.1176± 0.0589	0.1043± 0.0323	0.0779± 0.0115	**0.7113± 0.1216**
	G-M	0.5865± 0.1133	0.7674± 0.0581	0.6772± 0.1771	0.7101± 0.1409	0.6940± 0.1782	0.7223± 0.1054	0.6521± 0.0527	**0.7855± 0.0916**
33	ACC	0.9005± 0.0354	0.6596± 0.0886	**0.9502± 0.0139**	0.8070± 0.0423	0.6830± 0.1012	0.6783± 0.1094	0.6549± 0.0726	0.8302± 0.1735
	AUC	0.7349± 0.0847	0.7446± 0.0555	0.7298± 0.1264	0.7139± 0.1216	0.6750± 0.0523	0.7625± 0.1077	0.6933± 0.1589	**0.9136± 0.0082**
	F-M	0.1937± 0.0712	0.0956± 0.230	**0.2955± 0.1582**	0.1151± 0.0452	0.0813± 0.0108	0.1043± 0.0379	0.0819± 0.0382	0.2679± 0.2273

	G-M	0.7054± 0.1165	0.7323± 0.0490	0.6764± 0.1776	0.6886± 0.1536	0.6610± 0.0447	0.7467± 0.1106	0.6748± 0.1727	**0.9069± 0.0974**
34	ACC	0.9294± 0.0140	0.8376± 0.0610	0.9374± 0.0122	0.8945± 0.0467	0.9095± 0.0337	0.8362± 0.0679	0.8602± 0.0333	**0.9932± 0.0080**
	AUC	0.9564± 0.0120	0.8779± 0.0683	0.9177± 0.0611	0.9462± 0.0238	0.9539± 0.0172	0.9165± 0.0347	0.8732± 0.0909	**0.9965± 0.0041**
	F-M	0.6436± 0.0659	0.1802± 0.0277	0.6434± 0.0661	0.2836± 0.0897	0.3081± 0.0860	0.1970± 0.0545	0.1979± 0.0683	**0.8684± 0.0239**
	G-M	0.9562± 0.0118	0.8704± 0.0744	0.9151± 0.0642	0.9444± 0.0252	0.9526± 0.0180	0.9128± 0.0391	0.8693± 0.0940	**0.9965± 0.0041**
35	ACC	**1±0**	**1±0**	**1±0**	0.9834± 0.0371	**1±0**	**1±0**	**1±0**	**1±0**
	AUC	**1±0**	**1±0**	**1±0**	0.9915± 0.0188	**1±0**	**1±0**	**1±0**	**1±0**
	F-M	**1±0**	**1±0**	**1±0**	0.8533± 0.3279	**1±0**	**1±0**	**1±0**	**1±0**
	G-M	**1±0**	**1±0**	**1±0**	0.9913± 0.0192	**1±0**	**1±0**	**1±0**	**1±0**
36	ACC	**0.9996± 0.0010**	0.9861± 0.0168	0.9986± 0.0020	0.9605± 0.0232	0.9852± 0.0201	0.9771± 0.0223	0.9520± 0.0078	**1±0**
	AUC	0.9833± 0.0373	**0.9929± 0.0085**	0.9828± 0.0370	0.9800± 0.0117	0.9760± 0.0345	0.9884± 0.0113	0.9757± 0.0039	**1±0**
	F-M	**0.9818± 0.0407**	0.7471± 0.2704	0.9532± 0.0666	0.4848± 0.2898	0.7387± 0.2556	0.6533± 0.3207	0.3622± 0.0357	**1±0**
	G-M	0.9826± 0.0390	**0.9929± 0.0086**	0.9821± 0.0387	0.9797± 0.0119	0.9752± 0.0361	0.9882± 0.0114	0.9754± 0.0041	**1±0**
37	ACC	0.9694± 0.0142	0.9334± 0.0196	0.9660± 0.0153	0.9402± 0.0347	0.9544± 0.0340	0.9366± 0.0292	0.9357± 0.0235	**0.9995± 0.0010**
	AUC	0.9570± 0.0492	0.9662± 0.0099	0.9462± 0.0542	0.9697± 0.0176	0.9769± 0.0172	0.9679± 0.0148	0.9674± 0.0119	**0.9833± 0.0622**
	F-M	0.7258± 0.0399	0.2830± 0.0757	0.7178± 0.0448	0.3391± 0.1441	0.4431± 0.2470	0.3257± 0.1712	0.2950± 0.0914	**0.9818± 0.0407**
	G-M	0.9553± 0.0523	0.9656± 0.0102	0.9439± 0.0577	0.9691± 0.0181	0.9765± 0.0176	0.9672± 0.0151	0.9668± 0.0122	**0.9826± 0.0390**
38	ACC	0.9854± 0.0644	0.9825± 0.0279	0.9858± 0.0290	0.9818± 0.0283	0.9829± 0.0282	0.9863± 0.0292	0.9784± 0.0266	**0.9991± 0.0012**
	AUC	0.9543± 0.0621	0.9115± 0.0506	0.9545± 0.0623	0.9327± 0.0620	0.9315± 0.0631	0.9547± 0.0625	0.9094± 0.0514	**0.9550± 0.0622**
	F-M	0.8769± 0.1315	0.8420± 0.1243	0.8987± 0.1244	0.8057± 0.2371	0.8642± 0.1432	0.9169± 0.1327	0.7531± 0.2110	**0.9492± 0.0705**
	G-M	0.9525± 0.0644	0.9073± 0.0528	0.9528± 0.0646	0.9300± 0.0643	0.9285± 0.0658	**0.9530± 0.0648**	0.9055± 0.0535	0.9521± 0.0664
Performance	ACC	2/38	1/38	4/38	0/38	3/38	3/38	2/38	**31/38**
	AUC	2/38	4/38	1/38	0/38	4/38	3/38	2/38	**32/38**

F-M	2/38	1/38	4/38	0/38	3/38	3/38	2/38	**32/38**
G-M	2/38	3/38	1/38	0/38	4/38	3/38	2/38	**32/38**

As shown in the above table, the proposed MNEFD_IE achieves the best performance on more than 30 datasets. It generally shows significant improvement in each evaluation metric compared to the classical imbalanced ensemble algorithms. It means that the structural information among samples and high-quality envelope samples produced by MNEFD_IE is effective in improving the classification performance.

The average rankings of all compared algorithms on different evaluation metrics based on the Friedman aligned rank test are given to estimate the performance of MNEFD_IE. Table 5 and Fig. 8 show the average rank numbers of the proposed algorithm and the other seven classical imbalanced ensemble algorithms based on 38 datasets, where lower rank number indicates better classification ability.

Table 5. Average ranks of all compared ensemble methods

Algorithm	ACC	AUC	F-M	G-M
MNEFD_IE	**1.5**	**1.7763**	**1.5658**	**1.9605**
SBAG	3.6053	4.7105	3.7895	5.0395
UBAG	5.9868	4.4211	5.4079	4.4211
SBO	2.7632	5.4605	3.2237	5.6711
RBO	4.1974	3.8816	4.3026	3.75
EBO	4.6579	4.2368	4.6053	4.1184
BAC	6.7105	5.6974	6.4474	5.4079
Easy	6.5789	5.8158	6.6579	5.6316

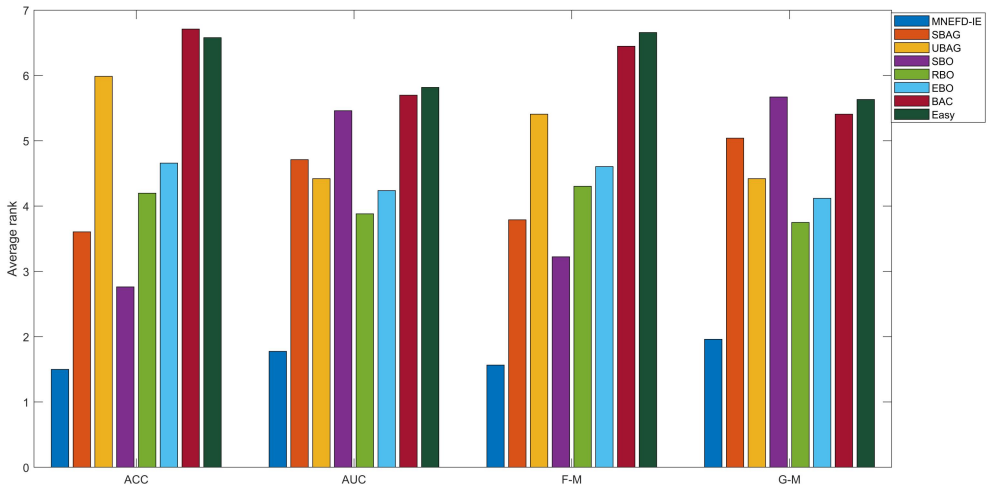


Fig. 8. Average ranks of all compared ensemble methods

As shown in Table 5, the rank numbers of MNEFD_IE in terms of the four evaluation metrics are 1.5, 1.7763, 1.5658, and 1.9605 in order, which are the lowest. It can be found from Fig. 8 that the average rank numbers of the other compared algorithms for the four metrics are significantly worse than those of MNEFD_IE. Therefore, the proposed algorithm MNEFD_IE outperforms the classical

imbalanced ensemble algorithms apparently.

The performance of MNEFD_IE was further evaluated using Holm's test. The results are shown in Table 6 by setting the MNEFD_IE as a control algorithm, thus further analyzing whether there are significant differences between the proposed algorithm and the compared algorithms.

Table 6. Results of Holm's test

Algorithm	ACC	AUC	F-M	G-M	Hypothesis (0.05)
SBAG	0.000179	0	0.000076	0	**Rejected**
UBAG	0	0.000003	0	0.000012	**Rejected**
SBO	0.024589	0	0.003175	0	**Rejected**
RBO	0.000002	0.000179	0.000001	0.001451	**Rejected**
EBO	0	0.000012	0	0.000123	**Rejected**
BAC	0	0	0	0	**Rejected**
Easy	0	0	0	0	**Rejected**

As shown in Table 6, the equivalence hypotheses between MNEFD_IE and the compared algorithms are all rejected, which indicates significant differences between MNEFD_IE and the compared algorithms at the significance level of 0.05.

6.4.2. Comparison with state-of-the-art algorithms

To further verify the performance of the proposed MNEFD_IE, four state-of-the-art imbalanced ensemble algorithms were chosen: EASE [45], SPE [46], HUE [47], and ECUBoost [48]. The specific results are shown in Table 7.

Table 7. The comparison results between EASE, SPE, HUE, ECUBoost and MNEFD_IE

ID	Measure	EASE	SPE	HUE	ECUBoost	MNEFD_IE
1	ACC	0.7756±0.0801	0.7754±0.0646	0.7942±0.0351	0.7620±0.0776	**0.9721±0.0104**
	AUC	0.7687±0.0901	0.7546±0.0679	0.7872±0.0430	0.7543±0.0821	**0.9662±0.0011**
	F-M	0.7008±0.1109	0.6845±0.0901	0.7238±0.0462	0.6849±0.0934	**0.9599±0.0125**
	G-M	0.7668±0.0911	0.7509±0.0694	0.7848±0.0426	0.7511±0.0816	**0.9656±0.0011**
2	ACC	0.9326±0.0224	0.9238±0.0149	0.9253±0.0202	0.9385±0.0228	**0.9898±0.0083**
	AUC	0.9337±0.0234	0.9154±0.0055	0.9262±0.0182	0.9401±0.0218	**0.9921±0.0064**
	F-M	0.9072±0.0303	0.8915±0.0150	0.8975±0.0255	0.9154±0.0308	**0.9857±0.0115**
	G-M	0.9335±0.0234	0.9140±0.0057	0.9259±0.0183	0.9399±0.0218	**0.9921±0.0065**
3	ACC	0.6523±0.0306	0.6470±0.0277	0.6588±0.0338	0.6745±0.0314	**0.8243±0.0302**
	AUC	0.6474±0.0297	0.6335±0.0344	0.6419±0.0339	0.6437±0.0312	**0.7954±0.0343**
	F-M	0.5590±0.0319	0.5372±0.0446	0.5455±0.0387	0.5357±0.0431	**0.7327±0.0494**
	G-M	0.6466±0.0297	0.6311±0.0359	0.6392±0.0339	0.6326±0.0367	**0.7823±0.0460**
4	ACC	**1±0**	**1±0**	**1±0**	**1±0**	**1±0**
	AUC	**1±0**	**1±0**	**1±0**	**1±0**	**1±0**
	F-M	**1±0**	**1±0**	**1±0**	**1±0**	**1±0**
	G-M	**1±0**	**1±0**	**1±0**	**1±0**	**1±0**
5	ACC	0.7358±0.0238	0.7237±0.0408	0.7243±0.0278	0.7385±0.0191	**0.7399±0.0516**
	AUC	0.7256±0.0244	0.7040±0.0357	0.7058±0.0278	0.7193±0.0161	**0.7397±0.0239**
	F-M	0.6054±0.0311	0.5804±0.0441	0.5818±0.0353	0.5986±0.0205	**0.6246±0.0357**
	G-M	0.7241±0.0248	0.7015±0.0355	0.7042±0.0280	0.7175±0.0162	**0.7366±0.0211**
6	ACC	0.6312±0.0802	0.5846±0.0690	0.5819±0.0796	0.4932±0.0634	**0.9905±0.0130**

	AUC	0.5877±0.0779	0.5823±0.0730	0.5339±0.0715	0.5519±0.0743	**0.9947±0.0072**
	F-M	0.4199±0.0827	0.4259±0.0793	0.3559±0.0764	0.4132±0.0719	**0.9556±0.0609**
	G-M	0.5792±0.0792	0.5807±0.0720	0.5188±0.0707	0.5327±0.0719	**0.9947±0.0073**
7	ACC	0.9645±0.0134	0.9645±0.0158	0.9538±0.0219	**0.9704±0.0106**	0.9503±0.0456
	AUC	0.9559±0.0189	0.9595±0.0166	0.9525±0.0178	**0.9636±0.0126**	0.9606±0.0313
	F-M	0.9257±0.0280	0.9328±0.0294	0.9149±0.0375	**0.9434±0.0183**	0.9148±0.0694
	G-M	0.9556±0.0191	0.9593±0.0167	0.9523±0.0178	**0.9632±0.0128**	0.9599±0.0325
8	ACC	0.7623±0.0278	0.7553±0.0079	0.7270±0.0374	0.7647±0.0285	**0.8120±0.0383**
	AUC	0.7281±0.0367	0.7206±0.0073	0.7333±0.0376	0.6860±0.0193	**0.7870±0.0509**
	F-M	0.5815±0.0494	0.5713±0.0100	0.5783±0.0438	0.5308±0.0293	**0.6601±0.0586**
	G-M	0.7242±0.0387	0.7168±0.0088	0.7311±0.0363	0.6672±0.0172	**0.7785±0.0612**
9	ACC	0.9346±0.0307	0.8967±0.0448	0.7270±0.0374	0.9345±0.0175	**0.9953±0.0104**
	AUC	0.9173±0.0380	0.8558±0.0552	0.7333±0.0376	0.9172±0.0419	**0.9970±0.0068**
	F-M	0.8668±0.0569	0.7840±0.0720	0.5783±0.0438	0.8640±0.0376	**0.9905±0.0213**
	G-M	0.9157±0.0391	0.8472±0.0607	0.7311±0.0363	0.9133±0.0461	**0.9969±0.0068**
10	ACC	0.9597±0.0196	0.9598±0.0137	0.9491±0.0087	0.9539±0.0195	**0.9657±0.0190**
	AUC	0.9545±0.0184	0.9459±0.0160	0.9581±0.0087	0.9542±0.0203	**0.9655±0.0118**
	F-M	0.9181±0.0376	0.9154±0.0267	0.9004±0.0162	0.9084±0.0357	**0.9311±0.0349**
	G-M	0.9544±0.0184	0.9451±0.0168	0.9579±0.0086	0.9535±0.0206	**0.9652±0.0119**
11	ACC	0.8989±0.0314	0.8841±0.0349	0.8571±0.0545	0.9136±0.0404	**0.9910±0.0082**
	AUC	0.8607±0.0500	0.8484±0.0653	0.8704±0.0412	0.8569±0.0625	**0.9800±0.0183**
	F-M	0.7802±0.0693	0.7524±0.0802	0.7476±0.0791	0.7981±0.0987	**0.9793±0.0189**
	G-M	0.8560±0.0536	0.8406±0.0732	0.8693±0.0418	0.8488±0.0684	**0.9797±0.0186**
12	ACC	0.9256±0.0523	0.9435±0.0303	0.8838±0.0176	0.8746±0.1476	**0.9761±0.0257**
	AUC	0.8871±0.0689	0.8942±0.0525	0.8846±0.0208	0.8575±0.0961	**0.9538±0.0291**
	F-M	0.7886±0.1235	0.8214±0.0922	0.7035±0.0223	0.7371±0.1897	**0.9242±0.0787**
	G-M	0.8840±0.0704	0.8895±0.0569	0.8831±0.0203	0.8454±0.1095	**0.9532±0.0293**
13	ACC	0.9486±0.0092	0.9439±0.0347	0.9302±0.0465	0.9485±0.0519	**0.9766±0.0233**
	AUC	0.9250±0.0623	0.8944±0.0963	0.9175±0.0435	0.9250±0.0669	**0.9586±0.0709**
	F-M	0.8197±0.0512	0.7921±0.1383	0.7967±0.1158	0.8387±0.1382	**0.9132±0.0881**
	G-M	0.9210±0.0676	0.8861±0.1090	0.9157±0.0441	0.9203±0.0721	**0.9551±0.0783**
14	ACC	0.9353±0.0062	0.9366±0.0157	0.9110±0.0208	0.9366±0.0152	**0.9905±0.0130**
	AUC	0.8826±0.0331	0.8810±0.0263	0.8964±0.0203	0.8969±0.0334	**0.9947±0.0072**
	F-M	0.7335±0.0327	0.7392±0.0576	0.6876±0.0478	0.7477±0.0422	**0.9556±0.0609**
	G-M	0.8791±0.0354	0.8779±0.0281	0.8955±0.0213	0.8942±0.0365	**0.9947±0.0073**
15	ACC	0.9078±0.0472	0.8986±0.0259	0.8572±0.0444	0.9195±0.0336	**0.9628±0.0584**
	AUC	0.8097±0.1166	0.7919±0.0953	0.8698±0.0447	0.8414±0.0921	**0.9442±0.0516**
	F-M	0.6100±0.1820	0.5695±0.1207	0.5736±0.0911	0.6571±0.1281	**0.8806±0.1533**
	G-M	0.7882±0.1378	0.7721±0.1094	0.8693±0.0443	0.8274±0.1096	**0.9413±0.0544**
16	ACC	0.9416±0.0261	0.9572±0.0131	0.8910±0.0187	0.9163±0.0265	**0.9845±0.0128**
	AUC	0.8623±0.0856	0.9316±0.0273	0.9047±0.0378	0.8661±0.0434	**0.9833±0.0251**
	F-M	0.7188±0.1360	0.8096±0.0486	0.6277±0.0546	0.6630±0.0430	**0.9286±0.0532**
	G-M	0.8517±0.0979	0.9302±0.0285	0.9037±0.0374	0.8594±0.0508	**0.9832±0.0253**
17	ACC	0.8806±0.0233	0.8599±0.0494	0.8086±0.0549	0.8977±0.0299	**0.9263±0.0039**

	AUC	0.8095±0.0620	0.8016±0.0910	0.7813±0.0486	0.7840±0.0367	**0.9235±0.0779**
	F-M	0.5387±0.0817	0.5109±0.1306	0.4440±0.1000	0.5538±0.0394	**0.6977±0.0370**
	G-M	0.8021±0.0676	0.7956±0.0947	0.7794±0.0493	0.7672±0.0488	**0.9184±0.0873**
18	ACC	0.7916±0.0600	0.7759±0.0543	0.6399±0.1113	0.6197±0.1118	**0.9795±0.0115**
	AUC	0.5700±0.0783	0.7054±0.1384	0.7059±0.0675	0.5061±0.1231	**0.9000±0.0559**
	F-M	0.2004±0.1249	0.2966±0.1573	0.2799±0.0489	0.1207±0.0815	**0.8857±0.0639**
	G-M	0.4436±0.2274	0.6121±0.3086	0.6843±0.0737	0.3878±0.2236	**0.8928±0.0599**
19	ACC	0.7944±0.0165	0.7849±0.0237	0.6966±0.0859	0.7988±0.0549	**0.9814±0.0104**
	AUC	0.7202±0.1439	0.6842±0.0887	0.7596±0.0851	0.7003±0.0758	**0.8607±0.0745**
	F-M	0.3147±0.1203	0.2911±0.0997	0.3145±0.0845	0.3233±0.0975	**0.8400±0.0894**
	G-M	0.6918±0.1640	0.6617±0.1144	0.7518±0.0819	0.6831±0.0928	**0.8532±0.0821**
20	ACC	0.8323±0.0389	0.7581±0.0205	0.6950±0.0887	**0.8583±0.0308**	0.8110±0.0531
	AUC	0.7088±0.0945	0.7001±0.0920	0.6663±0.1180	0.7537±0.0489	**0.8214±0.0294**
	F-M	0.2996±0.0735	0.2519±0.0564	0.2188±0.0905	0.3722±0.0445	**0.3732±0.0614**
	G-M	0.6767±0.1230	0.6888±0.0945	0.6430±0.1286	0.7390±0.0603	**0.8210±0.0291**
21	ACC	0.9486±0.0270	0.9486±0.0227	0.8321±0.0919	0.9627±0.0237	**0.9902±0.0134**
	AUC	0.8160±0.1426	0.8943±0.1058	0.8626±0.0873	0.9492±0.0710	**0.9949±0.0070**
	F-M	0.5895±0.2248	0.6599±0.1688	0.4204±0.1110	0.7690±0.1489	**0.9200±0.1095**
	G-M	0.7847±0.1711	0.8842±0.1187	0.8502±0.0989	0.9466±0.0759	**0.9948±0.0071**
22	ACC	0.9582±0.0274	0.9642±0.0223	0.8747±0.0515	0.9612±0.0152	**0.9911±0.0081**
	AUC	0.8373±0.1251	0.8873±0.0871	0.8865±0.1137	0.8623±0.1014	**0.9953±0.0043**
	F-M	0.6716±0.2234	0.7365±0.1263	0.4833±0.1598	0.6978±0.0939	**0.9333±0.0609**
	G-M	0.8096±0.1669	0.8756±0.1012	0.8808±0.1211	0.8436±0.1202	**0.9952±0.0043**
23	ACC	0.9083±0.0205	0.8768±0.0096	0.8371±0.0160	0.8617±0.0470	**0.9623±0.0048**
	AUC	0.7935±0.0862	0.7574±0.0591	0.7571±0.0417	0.7951±0.0890	**0.9212±0.0807**
	F-M	0.4554±0.1128	0.3636±0.0334	0.3202±0.0416	0.3780±0.0694	**0.7166±0.0151**
	G-M	0.7758±0.1065	0.7407±0.0688	0.7501±0.0469	0.7766±0.1180	**0.9173±0.0859**
24	ACC	**1±0**	0.9923±0.0153	**1±0**	0.9923±0.0153	**1±0**
	AUC	**1±0**	0.9000±0.2000	**1±0**	0.9500±0.0999	**1±0**
	F-M	**1±0**	0.8000±0.4000	**1±0**	0.9333±0.1333	**1±0**
	G-M	**1±0**	0.8000±0.4000	**1±0**	0.9414±0.1171	**1±0**
25	ACC	0.9812±0.0271	0.9812±0.0271	0.8788±0.0677	0.9112±0.0629	**0.9884±0.0164**
	AUC	0.8926±0.1968	0.9902±0.0142	0.9365±0.0356	0.9536±0.0330	**0.9939±0.0086**
	F-M	0.7142±0.3938	0.8476±0.1890	0.4504±0.1291	0.5454±0.2082	**0.9000±0.1414**
	G-M	0.7925±0.3965	0.9900±0.0144	0.9336±0.0382	0.9518±0.0348	**0.9939±0.0087**
26	ACC	0.9169±0.0287	0.8609±0.0267	0.7923±0.0579	0.8318±0.0444	**0.9792±0.0121**
	AUC	0.7653±0.0626	0.7600±0.1574	0.7959±0.0973	0.7448±0.0453	**0.8696±0.0978**
	F-M	0.3915±0.1006	0.2637±0.1019	0.2517±0.0684	0.2510±0.0470	**0.7532±0.1358**
	G-M	0.7433±0.0756	0.7180±0.2037	0.7895±0.0990	0.7334±0.0569	**0.8547±0.1152**
27	ACC	0.9083±0.0164	0.8712±0.0095	0.8167±0.0300	**0.9204±0.0229**	0.9097±0.0142
	AUC	0.7893±0.0782	0.8000±0.0392	0.8112±0.0534	0.8176±0.0429	**0.8209±0.0693**
	F-M	0.3316±0.0671	0.2791±0.0350	0.2351±0.0373	**0.3891±0.0734**	0.3553±0.0457
	G-M	0.7725±0.0917	0.7951±0.0437	0.8094±0.0520	0.8087±0.0485	**0.8122±0.0781**
28	ACC	0.7785±0.0302	0.7173±0.0542	0.6960±0.0204	0.6292±0.1777	**0.9677±0.0018**

	AUC	0.6124±0.0568	0.6168±0.0693	**0.6892±0.0451**	0.5906±0.0374	0.6034±0.1003
	F-M	0.1142±0.0266	0.1091±0.0345	0.1294±0.0156	0.0932±0.0192	**0.2699±0.1787**
	G-M	0.5779±0.0831	0.6010±0.0795	**0.6877±0.0459**	0.5529±0.0582	0.4239±0.2120
29	ACC	0.7877±0.0520	0.6980±0.0373	0.6737±0.0439	0.6737±0.1902	**0.8468±0.0976**
	AUC	0.6646±0.0267	0.6989±0.0447	0.6703±0.0719	0.7025±0.1043	**0.8000±0.0066**
	F-M	0.1424±0.0266	0.1274±0.0117	0.1129±0.0223	0.1559±0.0735	**0.2628±0.0997**
	G-M	0.6491±0.0288	0.6940±0.0469	0.6592±0.0845	0.6843±0.1096	**0.7954±0.0137**
30	ACC	**0.9770±0.0068**	0.9757±0.0107	0.9501±0.0100	0.9770±0.0077	0.9636±0.0140
	AUC	0.9006±0.0427	0.8892±0.0673	0.9513±0.0299	0.8885±0.0738	**0.9597±0.0291**
	F-M	**0.6854±0.0728**	0.6666±0.1285	0.5349±0.0514	0.6720±0.0938	0.6231±0.0788
	G-M	0.8956±0.0481	0.8815±0.0777	0.9509±0.0300	0.8794±0.0842	**0.9592±0.0295**
31	ACC	0.9420±0.0124	0.9103±0.0311	0.8786±0.0248	**0.9615±0.0083**	0.9137±0.0173
	AUC	0.8030±0.0698	0.8425±0.1048	0.8542±0.0245	0.8130±0.0597	**0.8868±0.0170**
	F-M	0.3509±0.0635	0.3038±0.0909	0.2482±0.0312	**0.4507±0.0919**	0.3234±0.0472
	G-M	0.7842±0.0874	0.8302±0.1192	0.8529±0.0265	0.7946±0.0768	**0.8861±0.0175**
32	ACC	0.9433±0.0214	0.8888±0.0312	0.8022±0.0191	0.6188±0.1883	**0.9889±0.0045**
	AUC	0.7511±0.1385	0.7232±0.0841	0.7034±0.0887	0.6829±0.0831	**0.8111±0.0722**
	F-M	0.2923±0.1600	0.1834±0.0531	0.1153±0.0293	0.1007±0.0560	**0.7113±0.1216**
	G-M	0.6427±0.3257	0.6890±0.1201	0.6816±0.1219	0.6595±0.0964	**0.7855±0.0916**
33	ACC	0.7964±0.0550	0.6257±0.0731	0.5918±0.0551	0.2350±0.1469	**0.8302±0.1735**
	AUC	0.5859±0.1207	0.6213±0.0944	0.6367±0.0701	0.5604±0.0913	**0.9136±0.0082**
	F-M	0.0732±0.0488	0.0623±0.0188	0.0638±0.0046	0.0480±0.0092	**0.2679±0.2273**
	G-M	0.4735±0.2590	0.5999±0.1044	0.6250±0.0588	0.4130±0.1613	**0.9069±0.0974**
34	ACC	0.9863±0.0075	0.9678±0.0073	0.9171±0.0135	0.8712±0.0521	**0.9932±0.0080**
	AUC	0.9244±0.0983	0.8627±0.1268	0.9577±0.0069	0.8951±0.0940	**0.9965±0.0041**
	F-M	0.7045±0.1528	0.4549±0.1158	0.3116±0.0331	0.2418±0.1103	**0.8684±0.0239**
	G-M	0.9151±0.1137	0.8407±0.1573	0.9568±0.0072	0.8914±0.0970	**0.9965±0.0041**
35	ACC	**1±0**	**1±0**	**1±0**	**1±0**	**1±0**
	AUC	**1±0**	**1±0**	**1±0**	**1±0**	**1±0**
	F-M	**1±0**	**1±0**	**1±0**	**1±0**	**1±0**
	G-M	**1±0**	**1±0**	**1±0**	**1±0**	**1±0**
36	ACC	**1±0**	**1±0**	0.9995±0.0008	0.9991±0.0010	**1±0**
	AUC	**1±0**	**1±0**	0.9833±0.0333	0.9666±0.0408	**1±0**
	F-M	**1±0**	**1±0**	0.9818±0.0363	0.9636±0.0445	**1±0**
	G-M	**1±0**	**1±0**	0.9825±0.0348	0.9651±0.0426	**1±0**
37	ACC	0.9990±0.0011	0.9995±0.0009	0.9872±0.0099	**1±0**	0.9995±0.0010
	AUC	0.9797±0.0398	0.9997±0.0004	0.9771±0.0317	**1±0**	0.9833±0.0622
	F-M	0.9595±0.0498	0.9846±0.0307	0.6964±0.1634	**1±0**	0.9818±0.0407
	G-M	0.9786±0.0421	0.9997±0.0004	0.9763±0.0331	**1±0**	0.9826±0.0390
38	ACC	**0.9995±0.0008**	0.9991±0.0011	0.9950±0.0078	0.9991±0.0017	0.9991±0.0012
	AUC	**0.9800±0.0399**	0.9550±0.0556	0.9479±0.0637	0.9600±0.0799	0.9550±0.0622
	F-M	**0.9777±0.0444**	0.9492±0.0630	0.8300±0.2357	0.9500±0.1000	0.9492±0.0705
	G-M	**0.9788±0.0422**	0.9520±0.0593	0.9446±0.0678	0.9549±0.0901	0.9521±0.0664
Perfor	ACC	6/38	3/38	3/38	7/38	**31/38**

manc	AUC	5/38	3/38	4/38	4/38	**34/38**
e	F-M	6/38	3/38	3/38	6/38	**32/38**
	G-M	5/38	3/38	4/38	4/38	**34/38**

As shown in Table 7, the proposed algorithm obtained the best performance on 34 datasets for AUC and G-M metrics and on more than 30 datasets for ACC and F-M metrics. This indicates that the proposed algorithm outperforms the four compared algorithms.

The average rank number and test results of the above five algorithms are given based on the Friedman aligned rank test and Holm's test in terms of four evaluation metrics. The specific results are shown in Tables 8-9.

Table 8. Average rank numbers of ensemble learning algorithms

Algorithm	ACC	AUC	F-M	G-M
MNEFD_IE	**1.5658**	**1.3816**	**1.5395**	**1.3947**
EASE	2.6053	3.25	2.6842	3.3684
SPE	3.3158	3.6184	3.4605	3.6316
HUE	4.3816	3.2895	4.1842	3.1053
ECUBoost	3.1316	3.4605	3.1316	3.5

Table 9. Results of Holm's test

Algorithm	ACC	AUC	F-M	G-M	Hypothesis (0.05)
EASE	0.004162	0	0.0016	0	**Rejected**
SPE	0.000001	0	0	0	**Rejected**
HUE	0	0	0	0.000002	**Rejected**
ECUBoost	0.000016	0	0.000011	0	**Rejected**

As shown in Table 8, the rank numbers of MNEFD_IE in the four evaluation metrics are 1.5658, 1.3816, 1.5395, and 1.3947 in order, which are the lowest. Table 9 shows that the equivalence hypotheses between MNEFD_IE and the compared algorithms are all rejected. Therefore, the proposed algorithm MNEFD_IE outperforms the state-of-the-art imbalanced ensemble algorithms.

To further evaluate the performance of the MNEFD_IE algorithm, the Wilcoxon paired signed-rank test was used to compare the proposed algorithm and the four compared algorithms one by one. The specific results are shown in Table 10.

Table 10. Results of the Wilcoxon pairwise test

Algorithm	Measure	R+	R-	P value	Hypothesis (0.05)
MNEFD_IE vs. EASE	ACC	676	65	1.0708e-05	**Rejected**
	AUC	681	22	8.1248e-07	**Rejected**
	F-M	698	43	2.4761e-06	**Rejected**
	G-M	697	44	5.2082e-06	**Rejected**
MNEFD_IE vs. SPE	ACC	679	24	1.4692e-06	**Rejected**
	AUC	683	20	6.8159e-07	**Rejected**
	F-M	715	26	8.1248e-07	**Rejected**
	G-M	664	39	4.0111e-06	**Rejected**
MNEFD_IE vs. HUE	ACC	697.5	5.5	2.9491e-07	**Rejected**
	AUC	713	28	1.2457e-06	**Rejected**

	F-M	698.5	4.5	2.7019e-07	**Rejected**
	G-M	667.5	35.5	3.5642e-06	**Rejected**
MNEFD_IE vs. ECUBoost	ACC	638.5	64.5	1.5890e-05	**Rejected**
	AUC	685.5	17.5	3.8848e-07	**Rejected**
	F-M	686.5	54.5	4.4861e-06	**Rejected**
	G-M	697.5	43.5	2.6341e-06	**Rejected**

As shown in Table 10, all equivalence hypotheses are rejected, which means that the proposed algorithm outperforms the four compared algorithms. Meanwhile, R+ in the table indicates that the rank sum of MNEFD_IE outperforms the compared algorithm based on 38 datasets, and R- indicates that the rank sum of the compared algorithm outperforms MNEFD_IE. Table 10 shows that R+ is much larger than R-, so MNEFD_IE is significantly better than the state-of-the-art algorithms.

7. Conclusion

Most current imbalanced ensemble algorithms use resampling to preprocess data to obtain balanced subsets for subsequent modeling. However, these algorithms still use a single sample as a unit for modeling, so they fail to consider the structural information among samples. This limitation leads to low separability, high sensitivity to noise, and high overlap among subsets. Studies show that the structural information between similar samples can solve the above limitation. Therefore, it is necessary to explore how to effectively mine the intersample structural information.

To solve the above problems, we propose a manifold neighboring envelope projection imbalanced ensemble algorithm with consistent FCM (MNEFD_IE). The algorithm constructs two kinds of structured samples (envelope samples) by mining structural information among samples and models based on the new samples rather than the original samples. The procedures of the algorithm are described in the Introduction and Method sections.

The Experiments section conducts groups of experiments to verify the proposed algorithm, including parameter analysis, ablation study, and algorithm comparison. Seven classical imbalanced ensemble algorithms, four advanced ensemble learning algorithms and 38 datasets are selected for comparison and verification of the proposed MNEFD_IE. As shown in Table 4, the proposed algorithm MNEFD_IE can effectively mine the structural information between samples and generate envelope samples through manifold neighboring envelope sample reconstruction (MNESR) and consistent fuzzy clustering (CFCMD) to improve the quality of samples and the model's classification performance. From Tables 5-6, we can find that the proposed algorithm MNEFD_IE outperforms the classical imbalanced ensemble algorithms in evaluation metrics. From Tables 7-10, it can be determined that the proposed algorithm MNEFD_IE also offers significant advantages over the state-of-the-art algorithms.

In summary, the proposed algorithm achieves satisfactory results, and the following conclusions can be drawn: 1) MNESR can effectively mine the nearest neighbor relationship between samples and reconstruct it into structured samples-neighboring envelope samples. 2) Consistent fuzzy clustering (CFCMD) can effectively mine the hierarchical structure information between samples and map them to the clustering centers, which are high-quality clustering envelope samples-neighboring cluster envelope samples. 3) The proposed algorithm can not only effectively improve the classification accuracy and the diversity of base classifiers but is also generally applicable to high-IR datasets. In addition, the proposed algorithm significantly outperforms the classical and state-of-the-art imbalanced ensemble algorithms. 4) The algorithm realize envelope-sample based modeling rather than

original-sample based modeling, different from the existing imbalanced ensemble algorithms.

Although the proposed algorithm is effective and the effectiveness of the envelope samples is shown, further research and improvement are still needed. Multiple-layer clustering to obtain multiple layers of envelope samples can be considered for further verification and improvement. In addition, it is necessary to explore other kinds of clustering algorithms except FCM for further verification and improvement.

References

[1] P. Yang, P. D. Yoo, J. Fernando, et al. Sample subset optimization techniques for imbalanced and ensemble learning problems in bioinformatics applications [J]. IEEE Transactions on Cybernetics, 2014, 44(3): 445-455.

[2] H. He, E. A. Garcia. Learning from imbalanced data [J]. IEEE Transactions on Knowledge and Data Engineering, 2009, 21(9): 1263-1284.

[3] Yang K, Yu Z , Chen C , et al. Incremental weighted ensemble broad learning system for imbalanced data [J]. IEEE Transactions on Knowledge and Data Engineering, 2021, 34(12): 5809-5824.

[4] S. F. Abdoh, M. A. Rizka, F. A. Maghraby. Cervical cancer diagnosis using random forest classifier with smote and feature reduction techniques [J]. IEEE Access, 2018, 6: 59475-59485.

[5] J. A. Sanz, D. Bernardo, F. Herrera, et al. A compact evolutionary interval-valued fuzzy rule-based classification system for the modeling and prediction of real-world financial applications with imbalanced data [J]. IEEE Transactions on Fuzzy System, 2015, 23(4): 973-990.

[6] X. Zhang, C. Zhu, H. Wu, et al. An imbalance compensation framework for background subtraction [J]. IEEE Transactions on Multimedia, 2017, 19(11): 2425-2438.

[7] Y. Liu, H. Lu, K. Yan, et al. Applying cost-sensitive extreme learning machine and dissimilarity integration to gene expression data classification [J]. Computational Intelligence and Neuroscience, 2016, 19: 1-10.

[8] Kotsiantis S B, Kanellopoulos D, Pintelas P E. Handling imbalanced datasets: A review [J]. GESTS International Transactions on Computer Science & Engineering, 2005, 30(1): 25-36.

[9] X. Gao, B. Ren, H. Zhang, et al. An ensemble imbalanced classification method based on model dynamic selection driven by data partition hybrid sampling [J]. Expert Systems with Applications, 2020, 160.

[10] P. Tapkan, L. Özbakır, S. Kulluk, A. Baykasog ˘ lu. A cost-sensitive classification algorithm: Bee-miner [J]. Knowledge-Based Systems, 2016, 95: 99-113.

[11] D. L. Donoho, J. Tanner. Precise undersampling theorems [J]. Proceedings of the IEEE, 2010, 98(6): 913-924.

[12] G. Douzas, F. Bacao. Self-organizing map oversampling (SOMO) for imbalanced data set learning [J]. Expert Systems with Applications, 2017, 82.

[13] E. Ramentol, Y . Caballero, R. Bello, et al. SMOTE-RSB: A hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory [J]. Knowledge and Information Systems, 2012, 33(2): 245-265.

[14] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches [J]. IEEE Transactions on Systems Man and Cybernetics Part C-Applications and Reviews, 2012, 42: 46-484.

[15] Yang K, Yu Z, Chen C. L. Philip, et al. Progressive Hybrid Classifier Ensemble for Imbalanced Data [J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2022, 52(4): 2464-2478.

[16] Wang S, Xin Y. Diversity analysis on imbalanced data sets by using ensemble models [C]//2009 IEEE Symposium on Computational Intelligence & Data Mining. Nashville, TN, USA, 324-331.

[17] Shuyin Xia, Yunsheng Liu, Guoyin Wang, et al. Granular ball computing classifiifiers for effiffifficient, scalable and robust

learning [J]. Information Sciences, 2019, 483:136-152.

[18] G. E. Batista, R. C. Prati, M. Monard. A study of the behavior of several methods for balancing machine learning training data [J]. ACM SIGKDD Explorations Newsletter, 2004, 6(1): 20-29.

[19] H. He, E. A. Garcia. Learning from imbalanced data [J]. IEEE Transactions on Knowledge and Data Engineering, 2009, 21(9): 1263-1284.

[20] H. He, Y . Bai, E. A. Garcia, and S. Li. ADASYN: Adaptive synthetic sampling approach for imbalanced learning [C]// IEEE International Joint Conference on Neural Networks, 2008: 1322-1328.

[21] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap. Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem [C]//13th Pacific–Asia Conference on Knowledge Discovery Data Mining, 2009, 5476: 475-482.

[22] WC. Lin, CF. Tsai, and YH. Hu. Clustering-based undersampling in class-imbalanced data [J]. Information Sciences, 2017, 409: 17-26.

[23] B. Krawczyk, M. Wo´zniak, and G. Schaefer. Cost-sensitive decision tree ensembles for effective imbalanced classification [J]. Applied Soft Computing, 2014, 14: 554-562.

[24] M. Z. Kukar and I. Kononenko. Cost-sensitive learning with neural networks [C]//13TH European Conference on Artificial Intelligence, 1998, 445-449.

[25] Y . Sun, M. S. Kamel, A. K. C. Wong, and Y . Wang. Cost-sensitive boosting for classification of imbalanced data [J]. Pattern Recognition, 2007, 40(12): 3358-3378.

[26] F.D. Frumosu, A.R. Khan, H. Schiøler, et al. Cost-sensitive learning classification strategy for predicting product failures [J]. Expert on Systems with Applications, 2020, 161: 113653.1-113653.20.

[27] Masnadi-Shirazi H, Vasconcelos N. Cost-sensitive boosting [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(2): 294-309.

[28] Barandela R, Sanchez JS, Valdovinos RM. New applications of ensembles of classifiers [J]. Pattern Analysis and Applications, 2003, 6(3): 245-256.

[29] D´ıez-Pastor JF, Rodr´ıguez JJ, Garc´ıa-Osorio CI, Kuncheva LI. Diversity techniques improve the performance of the best imbalance learning ensembles [J]. Information Sciences, 2015, 325: 98-117.

[30] Breiman L. Bagging predictors [J]. Machine Learning, 1996, 24(2): 123-140.

[31] Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting [J]. Journal of Computer and System Sciences, 1997, 55(1):119-139.

[32] Wang S, Xin Y. Diversity analysis on imbalanced data sets by using ensemble models [C]//2009 IEEE Symposium on Computational Intelligence & Data Mining. Nashville, TN, USA .324-331.

[33] Raghuwanshi B S, Shukla S. UnderBagging based reduced Kernelized weighted extreme learning machine for class imbalance learning[J]. Engineering Applications of Artificial Intelligence, 2018, 74:252-270.

[34] Chawla N V, Lazarevic A, Hall L O, et al. SMOTEBoost: Improving Prediction of the Minority Class in Boosting [C]//2003 European Conference on Knowledge Discovery in Databases: PKDD. Berlin, Heidelberg.107-119.

[35] Seiffert C, Khoshgoftaar T M, Van Hulse J, et al. RUSBoost: A Hybrid Approach to Alleviating Class Imbalance [J]. IEEE Transactions on Systems Man & Cybernetics Part A Systems & Humans, 2010, 40(1):185-197.

[36] Galar M, Fernández A, Barrenechea E, Herrera F. EUSBoost: enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling [J]. Pattern Recognition, 2013, 46(12): 3460-3471.

[37] X. Y. Liu, J. Wu, and Z. H. Zhou. Exploratory undersampling for class-imbalance learning [J]. IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics, 2009, 39(2): 539-550.

[38] Joshua B. Tenenbaum. A Global Geometric Framework for Nonlinear Dimensionality Reduction [J]. SCIENCE, 2000, 290(22): 2319-2323.

[39] Alcala-Fdez J, Fernandez A,Luengo J, et al. KEEL Data-Mining Software Tool: Data Set Repository, Integration of

Algorithms and Experimental Analysis Framework[J]. Journal of Multiple-Valued Logic & Soft Computing, 2011, 17(2-3):255-287.

[40] M. Lichman. (2013). UCI Machine Learning Repository. [Online].Available: http://archive.ics.uci.edu/ml

[41] S García, A Fernández, Luengo J, et al. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power[J]. Information Sciences, 2010, 180(10):2044-2064.

[42] Holm S. A simple sequentially rejective multiple test procedure [J]. Scandinavian Journal of Statistics, 1979, 6(2):65-70.

[43] S.M. Taheri, G. Hesamian. A generalization of the Wilcoxon signed-rank test and its applications [J]. Statistical Papers, 2013, 54(2): 457-470.

[44] Kuncheva LI. A Bound on Kappa-Error Diagrams for Analysis of Classifier Ensembles [J]. IEEE Transactions on Knowledge & Data Engineering, 2013, 25(3):494-501.

[45] Ren, JJ, Wang, YP. Equalization ensemble for large scale highly imbalanced data Classification [J]. Knowledge-Based Systems, 2022, 242: 108295.1-108295.11.

[46] Z. Liu, W. Cao, Z. Gao, J. Bian, H. Chen, Y. Chang, T. Liu. Self-paced ensemble for highly imbalanced massive data classification [C]// 36th IEEE International Conference On Data Engineering, ICDE 2020, Dallas, TX, USA, April 20-24, 2020, 841-852.

[47] W.W.Y. Ng, S. Xu, J. Zhang, X. Tian, T. Rong, S. Kwong. Hashing-based undersampling ensemble for imbalanced pattern classification problems [J]. IEEE Transactions on Cybernetics, 2022, 52(2):1269-1279.

[48] Z. Wang, C. Cao, Y. Zhu. Entropy and confidence-based undersampling boosting random forests for imbalanced problems [J]. IEEE Transactions Neural Networks and Learning Systems, 2021, 31(12): 5178-5191.