



Adaptive Fusion Based Method for Imbalanced Data Classification

Zefeng Liang^{1†}, Huan Wang^{2*†}, Kaixiang Yang^{3*} and Yifan Shi⁴

¹ School of Computer Science and Engineering, South China University of Technology, Guangzhou, China, ² Guangdong Institute of Scientific and Technical Information, Guangzhou, China, ³ State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou, China, ⁴ College of Engineering, Huaqiao University, Quanzhou, China

The imbalance problem is widespread in real-world applications. When training a classifier on the imbalance datasets, the classifier is hard to learn an appropriate decision boundary, which causes unsatisfying classification performance. To deal with the imbalance problem, various ensemble algorithms are proposed. However, conventional ensemble algorithms do not consider exploring an effective feature space to further improve the performance. In addition, they treat the base classifiers equally and ignore the different contributions of each base classifier to the ensemble result. In order to address these problems, we propose a novel ensemble algorithm that combines effective data transformation and an adaptive weighted voting scheme. First, we utilize modified metric learning to obtain an effective feature space based on imbalanced data. Next, the base classifiers are assigned different weights adaptively. The experiments on multiple imbalanced datasets, including images and biomedical datasets verify the superiority of our proposed ensemble algorithm.

OPEN ACCESS

Edited by:

Xin Jin,
Yunnan University, China

Reviewed by:

Liang Liao,
Zhongyuan University of Technology,
China
Peng Cao,
Northeastern University, China
Goksu Tuysuzoglu,
Dokuz Eylul University, Turkey

*Correspondence:

Huan Wang
wanghuan6@email.szu.edu.cn
Kaixiang Yang
yangkaixiang@zju.edu.cn

[†]These authors have contributed
equally to this work

Received: 02 December 2021

Accepted: 25 January 2022

Published: 28 February 2022

Citation:

Liang Z, Wang H, Yang K and Shi Y
(2022) Adaptive Fusion Based Method
for Imbalanced Data Classification.
Front. Neurobot. 16:827913.
doi: 10.3389/fnbot.2022.827913

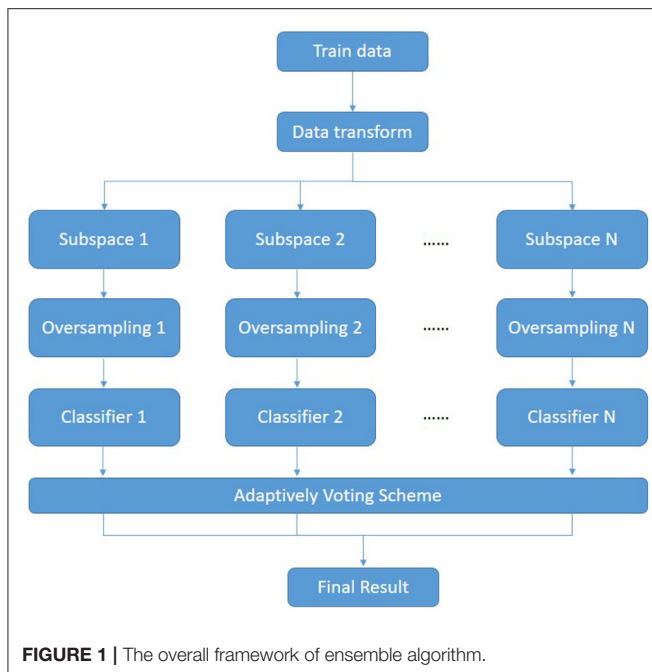
Keywords: imbalance learning, metric learning, information fusion, classification, ensemble learning

1. INTRODUCTION

Many applications face imbalance problems (Farrand et al., 2020; Khushi et al., 2021; Zhang et al., 2021). The imbalance problem is caused by the difference in the number of samples in each class. When the classifiers are trained on imbalanced datasets, the classifiers tend to favor the majority class and predict more samples to be the majority class. Therefore, the minority class samples can not be correctly classified, which is called the imbalance problem. The imbalance problem is widespread in the applications, so more and more researchers focus on dealing with the imbalance problem.

To solve the imbalance problem, researchers have proposed various methods from different perspectives. Cost-sensitive method (Elkan, 2001) is a typical one. The cost-sensitive method assigns different classification losses to each class. The minority class has a higher classification loss than the majority class, such that the classifiers pay more attention to the minority class and get a correct result. Resampling is another typical method. Resampling methods remove or synthesize samples from the original data to balance the number of samples in each class, including undersampling, oversampling, and hybrid sampling. Undersampling (He and Garcia, 2009) method removes the majority class samples by some informed rules. Undersampling can produce a more clear decision boundary while the information of the excluded samples is lost.

On the other hand, the oversampling method proposes to generate the synthesis of minority class samples until the data is balanced. The synthesis samples may lie in the overlapped area and make the distribution worsen. To overcome the disadvantages, the hybrid sampling is investigated.



As a two-stage strategy, hybrid sampling method combines undersampling and oversampling (L, 1976; Han et al., 2005). Other researchers combine the clustering method with oversampling (Barua et al., 2011).

Ensemble learning is widely used in solving the imbalance problem. Ensemble learning trains different classifiers and gets the result by integrated voting, which contains boosting and bagging (Ho, 1998; Skurichina and Duin, 2002; Chawla et al., 2003; Wang and Yao, 2009; Chen et al., 2010). Ensemble learning plays an essential role in the imbalance classification tasks (Bi and Zhang, 2018).

Nevertheless, the traditional imbalance algorithms have the following problems. Most algorithms do not consider mapping the imbalance data to another feature space for better classification performance. In addition, the importance of base classifiers is different, so it is inappropriate to treat the voting weight of base classifiers equally.

Metric learning is a hot topic in machine learning, which has been utilized in practical applications (Cao et al., 2019; Bai et al., 2021). Metric learning learns a feature space that is more effective than the original space. Euclidean distance is a common measure. However, Euclidean distance can not reflect the relationship correctly in the overlapped area. Consequently, some researchers capture the distance between samples by finding a transformation that can increase the distance between dissimilar samples and reduce the distance of similar samples (Köstinger et al., 2012). When training on the imbalance datasets, metric learning also suffers from imbalance problems (Gautheron et al., 2019). It needs to be modified before training on the imbalance datasets.

In this article, we propose an ensemble learning framework that combines metric learning and resampling. The metric learning is employed by building a feature space from the imbalanced dataset. The classifier is trained on the balanced datasets after oversampling on the feature space to reduce the

impact of imbalance. Finally, the classifier is integrated by adaptive weighted voting.

The contributions of the article are as follows:

1) An imbalanced version of the large margin nearest neighbor (LMNN) algorithm is proposed to alleviate the influence of imbalanced data distribution and learn a robust feature space.

2) An GA-based weighting scheme is designed to adaptively optimize the importance of different classifiers.

3) Extensive experiments are conducted on various imbalanced datasets to verify the effectiveness of the proposed approach.

The main framework is as follows: Section 2 introduces the related work about resampling, ensemble learning, and metric learning; Section 3 discusses the proposed ensemble framework in detail; Section 4 shows the experiments about our proposed methods and discusses the result of the experiments. Section 5 draws the conclusion and future study.

2. RELATED WORK

The resampling method contains undersampling and oversampling. To get a balanced dataset, undersampling methods remove the majority of samples randomly or by informed rules. Tomek link (Batista et al., 2004) removes samples that are of a different class from the neighbor. Undersampling can reduce the imbalance problem, while it may suffer from information loss. When the number of samples in each class is quite different, most of the majority of class samples are removed. Hence, the information of the majority of class is lost severely. On the other hand, oversampling proposes to generate synthesis minority class samples to balance the dataset. SMOTE (Chawla et al., 2002) propose to generate samples by interpolating between a given sample and its neighbors. The synthesis sample x_i^* is generated as follows:

$$x_i^* = x_i + (x_n - x_i) * r \quad (1)$$

In which x_n is the neighbor of sample x_i and r is a random value between $[0, 1]$. Adaptive Synthetic sampling approach (ADASYN) (He et al., 2008) makes different samples generate different numbers of synthetic samples. Some methods combine oversampling with clustering to overcome the problem that synthesized samples are located in the overlapped area. Majority Weighted Minority Oversampling Technique (MWMOTE) (Barua et al., 2014) generates minority samples within the cluster. Additionally, geometric-SMOTE (Douzas and Bacao, 2019) proposes a universal method that can be used in most oversampling methods. Mahalanobis Distance-based Oversampling technique (MDO) (Abdi and Hashemi, 2016) and its variant (Yang et al., 2018) propose to generate samples in the principal component space.

Euclidean distance is a traditional measure to reflect the similarity between samples. However, dissimilar samples may be closer to the similar samples in the overlapped area, which is inefficient to apply Euclidean distance. Metric learning learns a feature space that can reflect the relationship between samples more correctly. In the feature space, similar samples are closer while dissimilar samples are separated apart. To achieve this

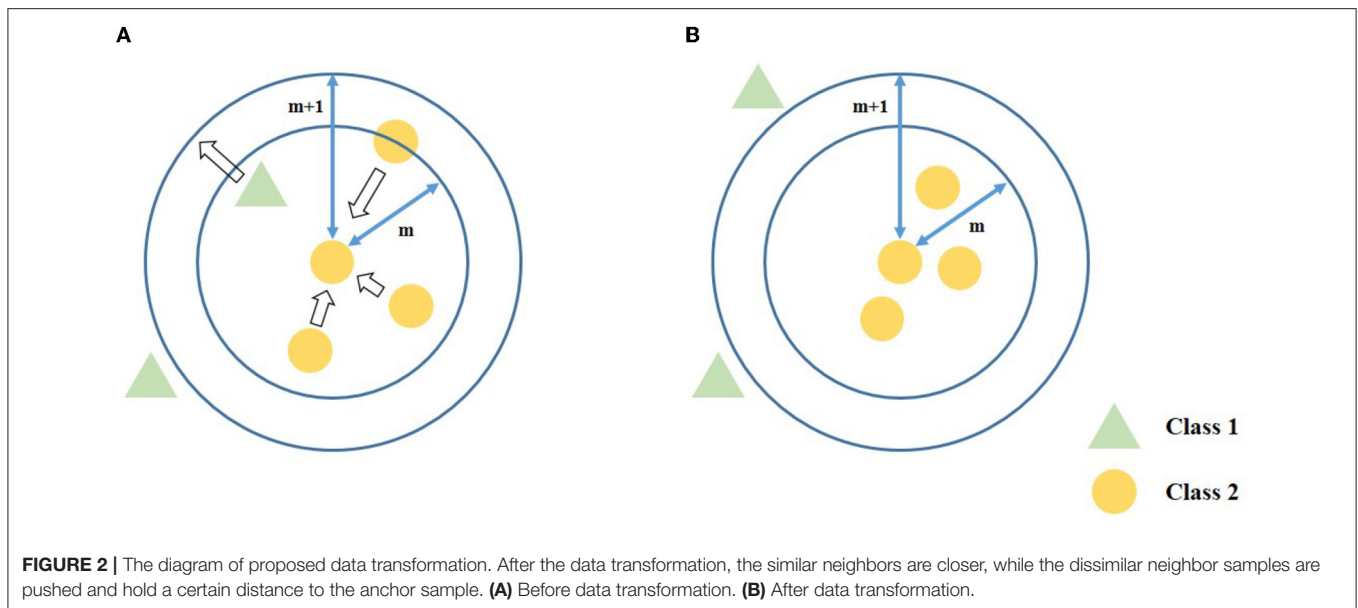


FIGURE 2 | The diagram of proposed data transformation. After the data transformation, the similar neighbors are closer, while the dissimilar neighbor samples are pushed and hold a certain distance to the anchor sample. **(A)** Before data transformation. **(B)** After data transformation.

goal, many metric learning algorithms have been proposed. LMNN (Weinberger and Saul, 2009) minimizes the distance between the anchor sample and its neighbors of the same class. At the same time, the anchor sample maintains a margin with neighbors of a different class. Information-theoretic metric learning (ITML) (Davis et al., 2007) makes distribution on the feature space similar to the Gaussian distribution. Some methods utilize metric learning on imbalanced datasets. Imbalance metric learning (IML) (Gautheron et al., 2019) modified LMNN by assigning different weights to sample pairs. Distance Metric by Balancing KL-divergence (DMBK) (Feng et al., 2019) balances the divergence of each class on the feature space. Iterative metric learning (Wang et al., 2018) learns a feature space for the area near each testing data.

Ensemble learning integrates classifiers to improve the robustness and performance of classification results. EasyEnsemble (Liu et al., 2009) trains several classifiers on the subset, which contains part of majority class samples and whole minority class samples. BalanceCascade (Liu et al., 2009) splits the majority class samples as several subsets and trains AdaBoost classifiers based on the subsets. Yang et al. (2021) proposes an ensemble framework based on subspace feature space ensemble and metric learning.

3. PROPOSED METHODOLOGY

In this section, we propose an ensemble framework combining metric learning with oversampling. **Figure 1** shows the overall framework of our proposed algorithm. First, the metric learning methods based on the imbalance problem (denoted as ImLMNN) are applied for getting a better feature space L . The data X is transformed by mapping matrix L and gets the mapped data X^* . Then, the feature space S_i is constructed. Next, the oversampling method is employed for getting a balance training dataset S_i^* . Finally, different classifiers are applied in balance

TABLE 1 | The attributes of datasets.

	IR	Samples	Features
climate	10.74	540	18
libras_move	14.00	360	90
ecoli2	5.46	90	7
glass_0_1_2_3_vs_4_5_6	3.20	214	9
yeast3	8.10	1484	8
cleveland_0_vs_4	12.31	173	13
winequality_red_4	29.17	1599	11
ecoli1	3.36	336	7

datasets and voting for the result. The pseudo-code is shown in the **Algorithm 1**.

We aim at finding a feature space that can better describe the sample relationship to improve the performance of classifiers. LMNN transforms data to a latent feature space, in which similar samples are closer while dissimilar samples are separated apart. The loss function of LMNN is as follows:

$$f(\mathbf{L}) = f_{push}(\mathbf{L}) + f_{pull}(\mathbf{L}) \tag{2}$$

where

$$f_{pull}(\mathbf{L}) = \sum_{i,j} \|\mathbf{L}(x_i - x_j)\|^2 \tag{3}$$

$$f_{push}(\mathbf{L}) = \sum_{i,j} \sum_l (1 - y_{il}) \left[1 + \|\mathbf{L}(x_i - x_j)\|^2 - \|\mathbf{L}(x_i - x_l)\|^2 \right]_+ \tag{4}$$

The loss function of LMNN contains $f_{push}(\mathbf{L})$ and $f_{pull}(\mathbf{L})$. $f_{pull}(\mathbf{L})$ reduces the distance between the anchor and its similar neighbors, while $f_{push}(\mathbf{L})$ penalizes the distance between

TABLE 2 | Comparisons between imbalance learning algorithm and our proposed method in terms of AUC.

	SMOTE	RandomForest	RUSboost	Balance bagging	LMNN ensemble	ImLMNN ensemble
climate	0.8785 ± 0.0092	0.5353 ± 0.0188	0.7462 ± 0.0245	0.8496 ± 0.0206	0.8787 ± 0.0162	0.8796 ± 0.021
libras_move	0.8741 ± 0.0227	0.8133 ± 0.025	0.8015 ± 0.06	0.8544 ± 0.0153	0.8146 ± 0.0286	0.8975 ± 0.0144
ecoli2	0.8671 ± 0.0121	0.813 ± 0.0113	0.8252 ± 0.0504	0.8678 ± 0.0138	0.8537 ± 0.0102	0.873 ± 0.0086
glass_0_1_2_3_vs_4_5_6	0.8945 ± 0.0172	0.876 ± 0.0129	0.8441 ± 0.039	0.8804 ± 0.0251	0.827 ± 0.0295	0.9064 ± 0.0173
yeast3	0.8278 ± 0.8278	0.7348 ± 0.0219	0.8163 ± 0.0258	0.8256 ± 0.0273	0.8381 ± 0.0163	0.8395 ± 0.0125
cleveland_0_vs_4	0.8919 ± 0.0103	0.6367 ± 0.088	0.7547 ± 0.05	0.848 ± 0.0405	0.8871 ± 0.0236	0.8955 ± 0.0419
winequality_red_4	0.6667 ± 0.0175	0.5291 ± 0.0093	0.5919 ± 0.0323	0.6515 ± 0.6515	0.6615 ± 0.0111	0.6776 ± 0.0116
ecoli1	0.8715 ± 0.0208	0.8461 ± 0.0297	0.7868 ± 0.0464	0.8756 ± 0.0228	0.8786 ± 0.0062	0.8804 ± 0.0117
AVERAGE_AUC	0.8465	0.723	0.7708	0.8316	0.8299	0.8562

The bold value means the best result among the compared algorithms.

Algorithm 1 : Imbalance Ensemble Framework.

Input: Training set $\mathbf{X} = (x_1, x_2, \dots, x_n)$

Parameter: Number of subspace N

Procedure:

- 1: Obtain the feature space \mathbf{L} by ImLMNN;
- 2: Map the data by learned feature space \mathbf{L} and get the mapped dataset \mathbf{X}^* ;
- 3: **for** i in $1, \dots, N$ **do**
- 4: Extract part of features by a threshold M and get the subspace \mathbf{S}_i ;
- 5: Apply SMOTE oversampling method on subspace \mathbf{S}_i and get the balanced subset \mathbf{S}_i^* ;
- 6: Classify on subset \mathbf{S}_i^* and get predict result y_i^* ;
- 7: **end for**
- 8: Vote for result by adaptive weight $\mathbf{W} = [w_1, w_2, \dots, w_N]$;

Output: The final result $\mathbf{Y} = (y_{final}^1, y_{final}^2, \dots, y_{final}^n)$.

dissimilar samples. $[z]_+ = \max(z, 0)$ is the hinge loss. $y_{il} = 1$ when x_i and x_l belong to the same class, otherwise $y_{il} = 0$.

However, the LMNN algorithm is inappropriate directly to apply in imbalanced datasets. To solve this problem, we assign different weights to samples. The weight w_i of sample x_i is as follows:

$$w_i = \frac{\delta^i}{|N_c| * d(x_i, \bar{X}_c)} \quad (5)$$

The loss of samples is divided by the number of samples N_c in the corresponding class, such that the impact caused by the imbalance problem is alleviated. To emphasize the samples near decision boundaries, we compute the sum of the density of majority class δ_n^i and minority class δ_p^i as density δ^i . The density δ^i is defined as follows:

$$\delta^i = \delta_n^i + \delta_p^i \quad (6)$$

$$\delta_n^i = \frac{1}{\frac{1}{k} \sum_{j=1}^k d_{ij}}, \quad \delta_p^i = \frac{1}{\frac{1}{h} \sum_{j=1}^h d_{ij}} \quad (7)$$

δ_n^i and δ_p^i describe the aggregation of samples in neighboring areas about majority class and minority class. k and h are the number of neighbor samples in calculating δ_n^i and δ_p^i , respectively. When the density δ_c^i is large, the samples in class c are close to x_i . Therefore, a large sum of density δ^i reflects that sample x_i is close to samples of both majority and minority classes or in the inner of class with high density.

Outliers and noises are also in the border area. To alleviate the influence of outliers and noises, we divide sample weight by $d(x_i, \bar{X}_c)$ which is the distance between sample x_i and the center of class c . The center of class c is defined as:

$$\bar{X}_c = \sum_{\{i|y_i=c\}} x_i. \quad (8)$$

Therefore, the overall objective function of the data transformation algorithm is:

$$f(\mathbf{L}) = \sum_{i,j} w_i \|\mathbf{L}(x_i - x_j)\|^2 + \sum_{i,j} \sum_i w_i (1 - y_{il}) \left[1 + \|\mathbf{L}(x_i - x_j)\|^2 - \|\mathbf{L}(x_i - x_l)\|^2 \right] \quad (9)$$

The diagram of data transformation is shown in **Figure 2**. Similar to LMNN, Equation (9) contains f_{pull} which pulls similar samples closer and f_{push} which push dissimilar samples separate apart. In addition, each sample has a different weight to deal with the imbalance problem.

After the data transformation, we extract M features to build the subspace \mathbf{S}_i :

$$\mathbf{S}_i = [f_i^1, f_i^2, \dots, f_i^M] \quad (10)$$

The feature is extracted N times to generate N subspace. In the subspace, the dataset is still imbalanced, which affects the classifier's performance. To solve this problem, oversampling is utilized in each feature subspace. Specifically, the subset is $\mathbf{S}_i^* = \mathbf{S}_i \cup \mathbf{Syn}_i$. The \mathbf{Syn}_i is the synthesis minority data that is generated by SMOTE on feature subspace \mathbf{S}_i and helps to form a new balanced subset \mathbf{S}_i^* with the original subspace.

The classifier c_i is trained on the balance subset \mathbf{S}_i^* and votes for obtaining the result. However, the performance of each

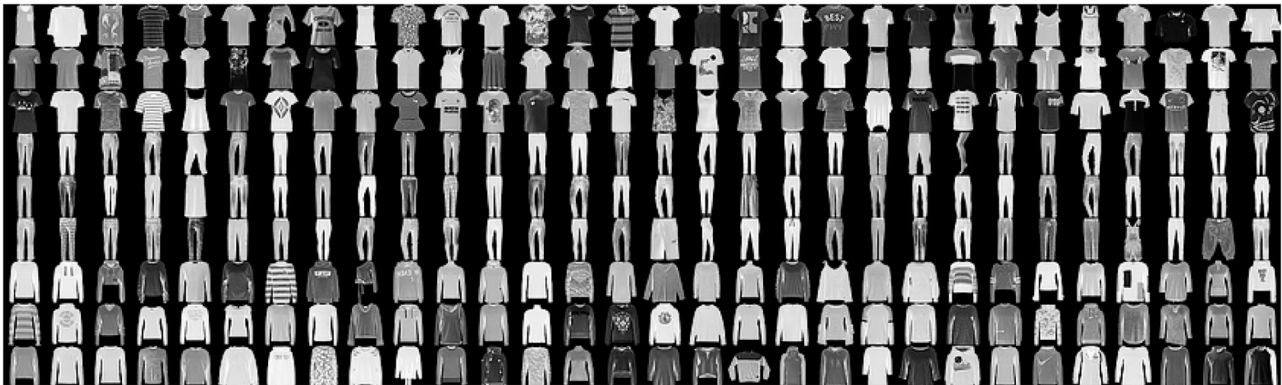


FIGURE 3 | The Fashion-mnist dataset.

TABLE 3 | Comparisons between imbalance learning algorithm and our proposed method in terms of AUC.

	SMOTE	RandomForest	RUSboost	Balance bagging	LMNN ensemble	ImLMNN ensemble
Fashion-mnist	0.9384 ± 0.0018	0.9523 ± 0.0025	0.9414 ± 0.006	0.9561 ± 0.0043	0.9543 ± 0.0044	0.9610 ± 0.0012

The bold value means the best result among the compared algorithms.

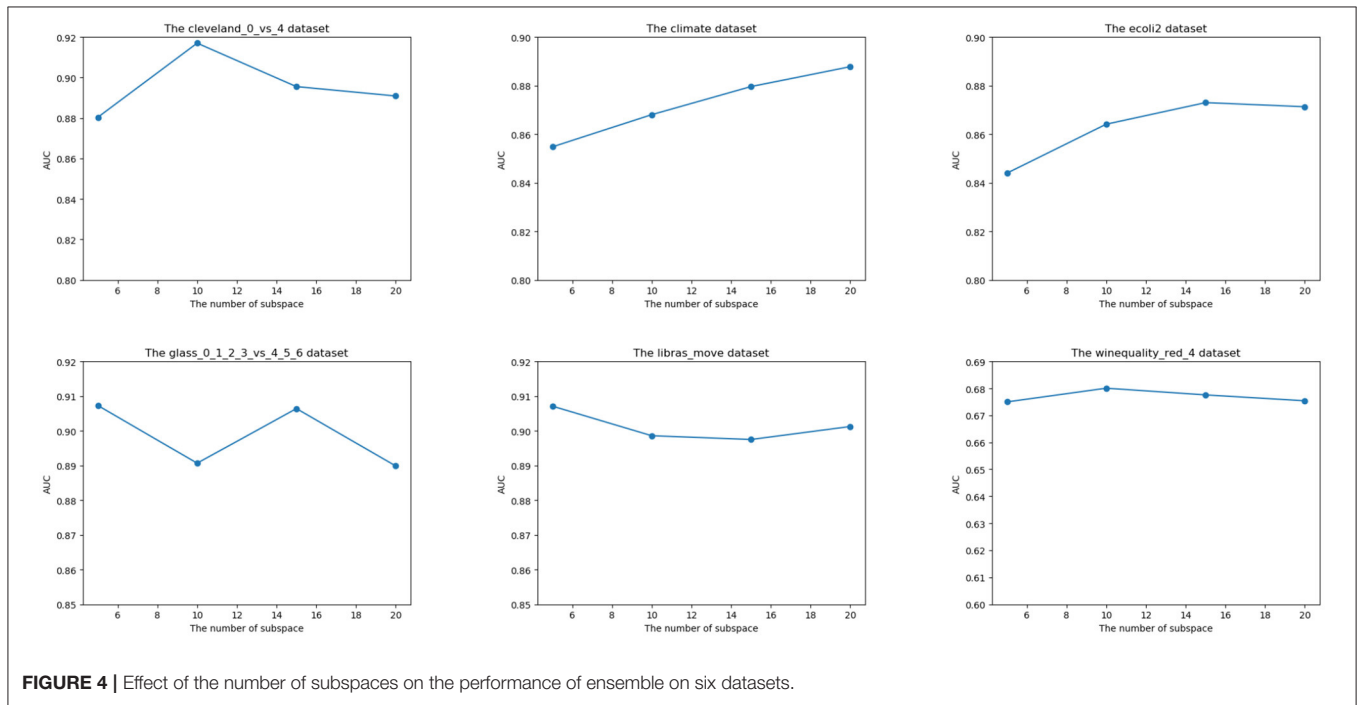


FIGURE 4 | Effect of the number of subspaces on the performance of ensemble on six datasets.

classifier is different. The contribution of each classifier in voting should be determined by the classification result, rather than being treated equally. Therefore, we utilize the weight assign process to progressively vote. In detail, the GA algorithm is applied to obtain the weight of each classifier adaptively. The detailed description is shown as **Algorithm 2**.

First, the initial genes $G = (g_1, g_2, \dots, g_n)$ is generated as the weight of the classifier, in which n is the number of subspace and g_i is the weight of classifier i . Next, the GA algorithm finds the n individual and does crossover and mutation. Given two parent

genes $g_i = [p_i^1, p_i^2, \dots, p_i^S]$ and $g_j = [p_j^1, p_j^2, \dots, p_j^S]$ with length S , the crossover method exchanges part of features in genes. Suppose the exchange occurs at position α ($\alpha \in [1, S]$), then the genes after the exchange are:

$$\begin{aligned} g_i^* &= [p_i^1, p_i^2, \dots, p_j^\alpha, \dots, p_i^S] \\ g_j^* &= [p_j^1, p_j^2, \dots, p_i^\alpha, \dots, p_j^S] \end{aligned} \tag{11}$$

Algorithm 2 : Adaptive weight Procedure.

Input: Classifier set $\mathbf{C} = (c_1, c_2, \dots, c_n)$

Parameter: Number of genes n , Population size of genes n_p , Max iteration N

Initialize: genes $\mathbf{G} = (g_1, g_2, \dots, g_n)$

Procedure:

```

1: while not converge do
2:   Select genes as parent randomly;
3:   Do crossover and mutation on parent genes to generate
    $n_{child}$  child genes by Eq. (11) and Eq. (12);
4:   for  $i$  in  $1, \dots, n$  do
5:     Calculates the fitness of gene  $F_{parent_i}$ ;
6:   end for
7:   for  $j$  in  $1, \dots, n_{child}$  do
8:     Calculates the fitness of gene  $F_{child_j}$ ;
9:   end for
10:  if  $F_{child_h} > F_{parent_l}$  then
11:    replace parent gene  $g_l$  by child gene  $g_h$ ;
12:  end if
13:  Until  $N$  Converge
14: end while

```

Output: The optimal genes $\mathbf{G}^* = (g_1^*, g_2^*, \dots, g_n^*)$.

The mutation may occur in each position of genes. Suppose the mutation happens in position γ ($\gamma \in [1, S]$) of gene g_k , then we have:

$$g_k^* = [p_k^1, p_k^2, \dots, p_k^\gamma, \dots, p_k^S] \quad (12)$$

in which p_k^γ is a random value. After crossover and mutation, the child's genes are generated. We calculate the fitness of parent genes F_{parent_l} ($l \in [1, n]$) and child genes F_{child_h} ($h \in [1, n_{child}]$). The fitness is set as AUC value. Finally, the parent genes are replaced by child genes with higher fitness values. When the iteration is over, the optimal classifier weight $\mathbf{G}^* = (g_1^*, g_2^*, \dots, g_n^*)$ is obtained.

We can get the final result by weighted voting integration. The result of classifier c_i is denoted as y_i . Then, the final result is

$$y_{final} = \sum_N g_i^* y_i \quad (13)$$

4. EXPERIMENT

In this section, we show the experiments about the proposed ensemble framework and compare the algorithm on various datasets from UCI (Dua and Graff, 2017) and KEEL (Alcala-Fdez et al., 2010). Our algorithm is also applied in the Fashion-mnist image dataset. Finally, we analyze the effect of parameters on our proposed algorithm.

4.1. Datasets

To evaluate the performance of our algorithm, we choose eight datasets from UCI and KEEL with different attributes, such as imbalance ratio (IR), number of samples, and features. The attributes are shown in **Table 1** in detail.

4.2. Evaluation Criteria

Accuracy is the typical criterion to evaluate the performance of the algorithm. However, due to the imbalance problem, accuracy is inappropriate for imbalance learning. AUC (Fawcett, 2004) is the area under the receiver operating characteristic curve, which is not sensitive to the imbalance data, and it is widely used in imbalance learning. In the experiments, we use AUC as evaluation criteria.

4.3. Comparison With Other Algorithms

To show the superiority, several algorithms and imbalance ensemble frameworks are compared with our algorithm. Specifically, we choose RandomForest, RUSboost, and BalanceBagging as the baseline. In addition, SMOTE algorithm is also chosen. The baseline algorithms are shown as follows:

1. SMOTE: A typical oversampling method. It generates samples by interpolating between samples and their neighbors.
2. RandomForest: An ensemble framework that uses bagging to build subsets for tree classifiers. The number of trees we set is 15.
3. RUSboost: A hybrid method that combines sampling with boosting. The number of iterations is 15.
4. BalanceBagging: A variant of Bagging that is applied sampling in each bootstrap. The number of subspaces we set is 15.

For our proposed algorithm, the number of subspaces is 15, and the ratio of extracted features is 0.7. To show the ablation experiment, we compare the LMNN ensemble, which replaces our proposed data transformation algorithm ImLMNN with the original LMNN algorithm. We choose linear SVM as the base classifier. The algorithms run five times and calculate average AUC as evaluation criteria. The 5-fold cross-validation is also applied. The result of the experiment is shown in **Table 2**.

From **Table 2**, we can see that our algorithm has the highest average AUC on given datasets, which is superior to other compared algorithms. Compared with other algorithms, the proposed algorithm has at least a 1% improvement in average AUC. Also, compared with the ensemble algorithm that applied the original LMNN algorithm as data transformation, our proposed method has a near 3% improvement in average AUC. Our method takes data transformation in the imbalanced datasets into account, which is superior to other compared algorithms.

4.4. Comparison With Different Algorithms on Image Dataset

Our algorithm is applied in the image dataset and compared with other algorithms. **Figure 3** shows part of samples in the Fashion-mnist dataset. Fashion-mnist is a famous image dataset that has 784 features and 60,000 samples. The dataset has 10 classes. To build the imbalanced dataset, we choose the T-shirt class as the majority class and the pullover class as the minority class. The majority and minority class samples are 3,000 and 600, respectively, which means the imbalance ratio is 5:1. Considering that the feature size is similar to the number of samples, we set the feature extraction ratio to 0.1. **Table 3** shows the result of the experiment.

We can see that our method also has the best performance in the Fashion-mnist dataset. The reason is that our proposed method can deal with the imbalance problem in the image dataset.

4.5. The Effect of Parameter

In this section, we show the impact of the parameter in our algorithm. The number of subspaces influences the performance of our ensemble framework. We set the number of subspace to [5,10,15,20]. The parameter experiment result is shown in **Figure 4**.

For Cleveland_0_vs_4, ecoli2, and winequality_red_4 datasets, the AUC is improved when the value of subspace is increased. However, when the number of subspaces exceeds a specific number, the AUC decreases as the subspace increases. The reason is that the increasing value of subspaces improves the diversity of subspace, while the excessive subspaces introduce redundant information and are harmful to the algorithm's performance. For other datasets, the trend of AUC is diverse due to the uncertainty of the GA algorithm. Considering the algorithm result on the overall dataset, the proposed number of the subspace is 15.

5. CONCLUSION AND FUTURE WORK

In this article, we propose an ensemble framework to deal with imbalanced datasets. We explore an effective feature space to improve the performance of the subsequent procedure. In addition, we propose an adaptive integrated voting process to assign weights for classifiers. The experiments on various

real-world imbalanced datasets, including the imbalanced image dataset, show the superiority of the proposed ensemble framework. Finally, we show the experiment to explore the effect of the parameter.

Future study contains several points: (1) Various methods can transform data into other feature spaces, so choosing appropriate methods should be considered. (2) A more effective adaptive weight process should be explored to assign weight based on the performance of the base classifiers.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found at: <http://archive.ics.uci.edu/ml/index.php>.

AUTHOR CONTRIBUTIONS

ZL, HW, and KY contributed to conception and design of the study. All authors contributed to manuscript revision, read, and approved the submitted version.

FUNDING

The work described in this study was partially funded by the National Key Research and Development Program of China under Grant 2019YFB1703600, in part by the National Natural Science Foundation of China No. 62106224 and Key-Area Research and Development Program of Guangdong Province (No. 2018B010107002).

REFERENCES

- Abdi, L., and Hashemi, S. (2016). To combat multi-class imbalanced problems by means of over-sampling techniques. *IEEE Trans. Knowl. Data Eng.* 28, 238–251. doi: 10.1109/TKDE.2015.2458858
- Alcala-Fdez, J., Fernández, A., Luengo, J., Garcia, S., Sanchez, L., et al. (2010). Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *J. Multiple Valued Logic Soft Comput.* 17, 255–287.
- Bai, S., Luo, Y., Yan, M., and Wan, Q. (2021). Distance metric learning for radio fingerprinting localization. *Exp. Syst. Appl.* 163:113747. doi: 10.3390/s21134605
- Barua, S., Islam, M. M., and Murase, K. (2011). *A Novel Synthetic Minority Oversampling Technique for Imbalanced Data Set Learning* (Shanghai: Springer), 735–744.
- Barua, S., Islam, M. M., Yao, X., and Murase, K. (2014). MWMOTE-majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Trans. Knowl. Data Eng.* 26, 405–425. doi: 10.1109/TKDE.2012.232
- Batista, G. E. A. P. A., Prati, R. C., and Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor. Newsl.* 6, 20–29. doi: 10.1145/1007730.1007735
- Bi, J., and Zhang, C. (2018). An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme. *Knowl. Based Syst.* 158, 81–93. doi: 10.1016/j.knsys.2018.05.037
- Cao, X., Ge, Y., Li, R., Zhao, J., and Jiao, L. (2019). Hyperspectral imagery classification with deep metric learning. *Neurocomputing* 356, 217–227. doi: 10.1016/j.neucom.2019.05.019
- Chawla, N., Bowyer, K., Hall, L., and Kegelmeyer, W. (2002). SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.* 16, 321–357. doi: 10.1007/978-3-540-39804-2_12
- Chawla, N. V., Lazarevic, A., Hall, L. O., and Bowyer, K. W. (2003). "Smoteboost: improving prediction of the minority class in boosting," in *Knowledge Discovery in Databases: PKDD 2003, 7th European Conference on Principles and Practice of Knowledge Discovery in Databases* (Cavtat), 107–119.
- Chen, S., He, H., and Garcia, E. A. (2010). Ramoost: ranked minority oversampling in boosting. *IEEE Trans. Neural Netw.* 21, 1624–1642. doi: 10.1109/TNN.2010.2066988
- Davis, J. V., Kulis, B., Jain, P., Sra, S., and Dhillon, I. S. (2007). *Information-Theoretic Metric Learning* (Corvallis, OR: Association for Computing Machinery), 209–216.
- Douzas, G., and Bacao, F. (2019). Geometric smote a geometrically enhanced drop-in replacement for smote. *Inf. Sci.* 501, 118–135. doi: 10.1016/j.ins.2019.06.007
- Dua, D., and Graff, C. (2017). UCI machine learning repository. Irvine, CA: University of California, School of Information and Computer Science. Available online at: <http://archive.ics.uci.edu/ml>
- Elkan, C. (2001). "The foundations of cost-sensitive learning," in *Proceedings of the 17th International Joint Conference on Artificial Intelligence* (San Francisco, CA) 973–978.
- Farrand, T., Miresghallah, F., Singh, S., and Trask, A. (2020). "Neither private nor fair: impact of data imbalance on utility and fairness in differential privacy," in *Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice PPMLP'20* (New York, NY: Association for Computing Machinery), 15–19.
- Fawcett, T. (2004). Roc graphs: notes and practical considerations for data mining researchers. *ReCALL* 31, 1–38.

- Feng, L., Wang, H., Jin, B., Li, H., Xue, M., and Wang, L. (2019). Learning a distance metric by balancing kl-divergence for imbalanced datasets. *IEEE Trans. Syst. Man Cybern. Syst.* 49, 2384–2395. doi: 10.1109/TSMC.2018.2790914
- Gautheron, L., Habrard, A., Morvant, E., and Sebban, M. (2019). *Metric Learning From Imbalanced Data* (Portland, OR: Institute of Electrical and Electronics Engineers), 923–930.
- Han, H., Wang, W.-Y., and Mao, B.-H. (2005). *Borderline-Smote: a New Over-Sampling Method in Imbalanced Data Sets Learning* (Hefei: Springer), 878–887.
- He, H., Bai, Y., Garcia, E. A., and Li, S. (2008). “ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning,” in *Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)* (Hong Kong: IEEE), 1322–1328. doi: 10.1109/IJCNN.2008.4633969
- He, H., and Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* 21, 1263–1284. doi: 10.1109/TKDE.2008.239
- Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.* 20, 832–844.
- I., T. (1976). Two modifications of cnn. *IEEE Trans. Syst. Man Cybern. SMC-6*, 769–772.
- Khushi, M., Shaukat, K., Alam, T. M., Hameed, I. A., Uddin, S., Luo, S., et al. (2021). A comparative performance analysis of data resampling methods on imbalance medical data. *IEEE Access* 9, 109960–109975. doi: 10.1109/ACCESS.2021.3102399
- Köstinger, M., Hirzer, M., Wohlhart, P., Roth, P. M., and Bischof, H. (2012). “Large scale metric learning from equivalence constraints,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition* (Providence, RI), 2288–2295.
- Liu, X., Wu, J., and Zhou, Z. (2009). Exploratory undersampling for class-imbalance learning. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* 39, 539–550. doi: 10.1109/TSMCB.2008.2007853
- Skurichina, M. and Duin, R. P. W. (2002). Bagging, boosting and the random subspace method for linear classifiers. *Pattern Anal. Appl.* 5, 121–135. doi: 10.1007/s100440200011
- Wang, N., Zhao, X., Jiang, Y., Gao, Y., and BNRist, K. (2018). “Iterative metric learning for imbalance data classification,” *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence* (Stockholm), 2805–2811.
- Wang, S. and Yao, X. (2009). “Diversity analysis on imbalanced data sets by using ensemble models,” in *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2009, part of the IEEE Symposium Series on Computational Intelligence* (Nashville, TN), 324–331.
- Weinberger, K. Q., and Saul, L. K. (2009). Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.* 10, 207–244.
- Yang, K., Yu, Z., Chen, C. L. P., Cao, W., Wong, H.-S., You, J., and Han, G. (2021). Progressive hybrid classifier ensemble for imbalanced data. *IEEE Trans. Syst. Man Cybern. Syst.* 1–15. doi: 10.1109/TSMC.2021.3051138
- Yang, X., Kuang, Q., Zhang, W., and Zhang, G. (2018). Amdo: An over-sampling technique for multi-class imbalanced problems. *IEEE Trans. Knowl. Data Eng.* 30, 1672–1685. doi: 10.1109/TKDE.2017.2761347
- Zhang, J., Zhang, Q., He, X., Sun, G., and Zhou, D. (2021). Compound-fault diagnosis of rotating machinery: A fused imbalance learning method. *IEEE Trans. Control Syst. Technol.* 29, 1462–1474. doi: 10.1109/TCST.2020.3015514

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher’s Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Liang, Wang, Yang and Shi. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.