International Journal for Housing Science and Its Applications
Publish August 6, 2025. Volume 46, Issue 3 Pages 7289-7299

https://doi.org/10.70517/ijhsa463624

Research on performance enhancement of integrated learning algorithm based on sample weight allocation mechanism and Its Application in Image Classification

Yuting Zhang[1], Jiao Bao[1] and Jueyan Li[2,*]
[1] Chengdu Technological University, Chengdu, Sichuan, 610000, China
[2] 460media, Adelaide, SA5000, Australia
Corresponding authors: (e-mail: ljy25311@gmail.com).

Abstract Aiming at the challenges of insufficient model generalization ability and computational inefficiency in class imbalance multiclassification problems, this paper proposes an integrated learning algorithm optimization framework based on the sample weight distribution mechanism. A Gaussian mapping-enhanced G-SMOTE oversampling method is designed to dynamically adjust the boundary distribution weights of a few class samples. Combining the fast binary classification property of TWSVM and the weight adaptation mechanism of AdaBoost, the integrated model based on the OVO strategy is constructed.The average AUC value of the G-SMOTE method on the 2 datasets with low imbalance ratios is 0.891, which is higher than that of the original dataset, the single downsampling, and the SMOTE oversampling, respectively, by 0.181, 0.187, and 0.137.The mean AUC value on the 2 datasets with high imbalance ratios is 0.891, which is higher than that of the original dataset, the single downsampling, and the SMOTE oversampling, respectively. The same performance is optimal on the 2 datasets with high imbalance ratios.The convergence speed of AdaBoost-TWSVM has an advantage over Pa_Ada, and a large advantage over SWA_Adaboost and IPAB. The test error of AdaBoost-TWSVM is reduced by an average of 9.22, 15.41, 6.08, and 9.38 percentage points compared to the other six algorithms on the four datasets, respectively. Compared with TWSVM, the acceleration ratios of AdaBoost-TWSVM algorithms are all improved to a certain extent, and the acceleration effect is most significant in the high-dimensional dataset Kddcupbuffer, with the acceleration ratio of node 3 reaching 2.37 ± 0.03. This algorithm demonstrates strong parallel computing capabilities and scalability when handling large-scale datasets, making it suitable for the classification and detection of painted images. When applied to the classification of Zhang Daqian's early and later landscape paintings, the algorithm achieved more satisfactory results in image classification accuracy.

Index Terms integrated learning, Image Classification, G-SMOTE, OVO, AdaBoost-TWSVM

I. Introduction

With the rapid development of key technologies such as the Internet, cloud computing, and big data, artificial intelligence technology is profoundly changing people's lives and promoting changes in all aspects of society. Machine learning is an important part of AI, aiming to learn potential relationships between data from a large amount of data, with great business potential [1], [2]. And integration learning occupies an important position in machine learning, aiming at generating and combining multiple base learners to obtain a learner with stronger generalization performance [3], [4]. This technique is often used to improve the generalization performance of a single learner, and has achieved significant success in practical applications in biology, finance, and recommender systems, etc. It has been called the "evergreen tree" in the field of machine learning, and has attracted extensive attention in both academia and industry [5]-[7].

However, in these fields, the problem of unbalanced data classification has become one of the main directions of classification learning, and it is difficult to achieve a desirable effect on the detection accuracy of data of a few classes using the traditional integrated learning classification algorithms [8]-[10]. In order to effectively solve the pseudo-balancing problem of undersampling techniques in dealing with unbalanced data, weights can be adaptively assigned to the original data during the iteration process of training classifiers, so as to avoid that poorly performing classifiers affect the overall decision-making, and thus to improve the generalization ability of the integrated model [11]-[14].

In this paper, the G-SMOTE oversampling method is firstly designed to solve the unbalanced data problem, and Gaussian mapping is introduced in adaptively deciding the number of samples. Combining the fast solving ability of TWSVM and the sample weight optimization mechanism of AdaBoost, the AdaBoost-TWSVM multi-classification algorithm based on OVO is proposed. A multi-dimensional experimental validation framework is designed to cover four types of data features, LL-LH-HL-HH. Compare the convergence speeds of different algorithms using eight-fold cross-validation. Quantify the stability of the algorithms using Friedman's test, and verify the engineering applicability by parallel acceleration experiments.At the end of the article, the applicability of the algorithm in image classification was validated using Zhang Daqian's early and late Chinese painting works.

II. Unbalanced data multi-classification algorithm design based on improved SVM integrated learning

In today's information society, extracting valuable information from massive and complex data has become one of the key research topics. Especially in the processing of categorized datasets, the class imbalance phenomenon - i.e., the proportion of data in some classes is significantly higher than that of other classes - often occurs, especially in the multiclassification problem, which is more common. In this paper, based on cutting-edge machine learning techniques, we explore the problem of multiclassified unbalanced data in depth and propose targeted solution strategies.

II. A.G-SMOTE algorithm with weight matching

Class imbalance is an unavoidable problem in software defect prediction work, although the existing oversampling methods improve the performance of defect prediction to a certain extent, the sampling base point and sampling location still need to be considered in depth, so this paper proposes a new oversampling method, G-SMOTE.Firstly, the sampling base point, i.e., based on which samples to be sampled is analyzed, and the second consideration is the sampling location, and finally, the above mentioned requirement is given The technical realization of the

(1) Sampling base point: both Borderline-SMOTE and SVMSMOTE only consider the samples of the DANGER class near the boundary, while ignoring the other two minority class samples.The class imbalance problem in SDP arises precisely because there are more defect-free samples than defective samples. Therefore, the number of samples is a non-negligible factor. Analyzing the existing data set through the process inquiry experiment, it was found that the number of samples of the remaining two minority classes was about 3-10 times that of the DANGER class. Therefore, the remaining two minority classes cannot be ignored. Therefore, it is more reasonable to use all the minority classes to synthesize new samples.

(2) Sampling location: it is more difficult for samples at the boundary to be classified correctly.Borderline-SMOTE oversampling method emphasizes that samples at the edges are more likely to be classified incorrectly and are more important for classification. If the classification hyperplane can separate the sample points near the boundary, then the samples away from the boundary can be separated as well. Therefore, more samples should be sampled near the boundary to obtain a clearer classification boundary.

(3) Requirement realization: the adaptive nature of the ADASYN method makes the number of samples of the few classes with larger weights sampled more, which will make the sampling concentrated near the noise points. If the sampling base point is a noise point, the generated samples are also extremely probable to be noise points, which leads to the degradation of model performance. Therefore, there is a need to classify the minority class samples. The new classification criterion should be able to use the whole data without discarding the possible non-noise points, and should also be able to take more new samples at the boundary. Therefore, this paper proposes a soft classification scheme for Gaussian mapping.

Based on the bell curve property inherent in the Gaussian distribution, it is observed that in most categories, the samples of a few categories tend to be surrounded by a large number of non-similar samples, whereas the presence of dissimilar samples is extremely sparse within those samples belonging to the few categories. Specifically, when the number of dissimilar samples is used as an input variable to the Gaussian function, the samples in these two different contexts will be distributed on the two opposite flanks of the bell curve, in stark contrast. In this context, the results of sampling the samples in this way usually show a tendency to collect relatively few samples far from the center of the curve and near the tails of the two flanks in the overall sample space, especially in those regions that represent a few categories. Conversely, near the peak of the curve, i.e., in the middle part of the bell curve, the number of samples taken is more targeted compared to the edge regions due to the large number of samples from each category it covers. Therefore, with the help of this sampling strategy optimized according to the characteristics of the Gaussian distribution, the density of samples in the classification boundary region can be increased in a targeted manner, thus strengthening the ability and accuracy of the model for category differentiation, and ultimately realizing the effective improvement of the overall classification

performance. A comparison of the sampling positions with and without Gaussian mapping is shown in Figure 1. It can be seen that when oversampling a few classes in proportion to the Gaussian value, more samples are taken at the boundary, and the points at the non-boundary are not discarded, but the number of samples is less.

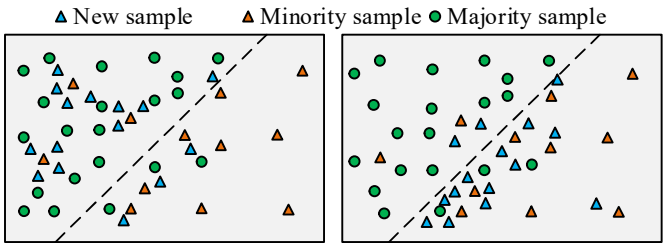


Figure 1: Comparison of sampling quantities

It is clear from practical experience that in a variety of industry sectors, whether industrial, manufacturing or aerospace, the damage caused by misclassification of defective data is often greater than that caused by misclassification of non-defective data. At the same time, responses from the industry indicate that it is even more important to correctly categorize modules that are prone to defects. Therefore, the ultimate goal of this paper is to correctly predict more defects while maintaining the overall prediction results. The over-sampled samples only reach the class equilibrium state, and the classification effect of the model does not realize the purpose of this paper. Inspired by the cost-sensitive method, the classification ability of the model for a few classes can be improved by setting weights for the samples and increasing the weights of the samples for a few classes. The environment factor after Gaussian mapping can satisfy the above requirements. Therefore, the Gaussian mapped environmental factors can be set as the weights of the minority class samples, and the environmental factors without Gaussian mapping can be set as the weights of the majority class samples.

Although setting the environmental factor can make the weight of the minority class greater than that of the majority class, the ratio of the weights of the minority class and the majority class cannot be set in advance, so a balancing coefficient μ is needed to measure the importance of the minority class and the majority class, and a reasonable value of μ should be derived through experiments. If necessary, the stability of the model should also be explored by setting threat conditions to further verify the reasonableness of the value of μ.

According to the above analysis, the G-SMOTE algorithm with weight matching can be summarized as follows: G-SMOTE considers all minority class samples when oversampling, and samples more near the boundary without ignoring possible non-noise points. After the completion of oversampling to get the minority class environmental factors and the majority class environmental factors and balanced data sets, at this time has basically solved the class imbalance state. Then increase the influence of the minority class on the model prediction results, the minority class and the majority class μ set different weights.

The formula for Gaussian mapping is shown in equation (1):

$$Gaussian\left(n_i^{\min}\right)=\exp\left(-\frac{n_i^{\min}-\frac{1}{N}\sum_{j}^{N}n_j^{\min}}{2\sigma^2}\right) \tag{1}$$

where σ is the variance of the Gaussian mapping with a value of 1 and N denotes the number of samples. The environmental factors $wMin$ and $wMaj$ can be obtained by substituting the Gaussian mapping of $n_i^{\min}$ and $n_j^{\min}$ into the function of $soft\max(\cdot)$ and the formula of $soft\max(\cdot)$ is shown in Eq. (2) :

$$soft\max(n_i)=n_i/\sum_{j}^{N}n_j \tag{2}$$

The formula for calculating the number of samples to be synthesized for a few classes is shown in equation (3):

$$\delta_i^{\min}=wMin_i\times G \tag{3}$$

Set weights μ for the minority environmental factors and $1-\mu$ for the majority environmental factors. By adjusting the balance coefficients, the predictions of the model can be controlled so that the probability value of its prediction for the minority class can be further increased. The weights of the samples are generated using the weight mean complementary new, and the $mean(\cdot)$ function is calculated as shown in equation (4).

$$mean(wMin)=\frac{1}{N_{\min}}\sum_{i=0}^{N_{\min}-1}wMin[i] \tag{4}$$

At this point, the data is in equilibrium and all weighting information is captured.

II. B.Support vector machine model

SVM is a model for solving binary classification problems that aims to find a hyperplane of dimension $m-1$ that divides N m-dimensional data into two classes. Its learning strategy is to maximize the interval; the larger the interval, the greater the difference between the two types of samples, and the easier they can be distinguished. Thus, the problem of finding an optimal decision hyperplane can be transformed into solving the problem of maximizing the interval between the two types of samples. The resulting problem can be turned into constructing and solving a constrained optimization problem:

$$\min_{w}\frac{1}{2}\|w\|^2 \tag{5}$$

$$s.t.\ y_i(w\cdot x_i+b)\geq 1,\ i=1,2,\cdots,N \tag{6}$$

The separating hyperplane obtained by interval maximization for a given linearly divisible training dataset is:

$$w^*\cdot x+b^*=0 \tag{7}$$

The corresponding classification decision function is:

$$f(x)=\mathrm{sgn}\left(w^*\cdot x+b^*\right) \tag{8}$$

However, when it comes to nonlinearly divisible datasets, there are two ways to solve this problem, the first is to dimension the data by kernel function, and use SVM model in high dimension to find the corresponding decision hyperplane; the second is to introduce relaxation variables, i.e., constraints can be violated to some extent to form a soft spacing, which at this point can be represented by the following optimization problem:

$$\min_{w,\xi}\frac{1}{2}\|w\|^2+C\sum_i\xi_i \tag{9}$$

$$s.t.\ y_i(w\cdot x_i+b)\geq 1-\xi_i \tag{10}$$

$$\xi_i\geq 0\ i=1,2,\cdots,N \tag{11}$$

The solution to the above problem is usually to transform it into a dyadic problem, then the original problem is transformed into a convex quadratic programming problem with a globally optimal solution:

$$\min_{\lambda}\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j y_i y_j K(x_i,x_j)-\sum_{i=1}^{N}\lambda_i \tag{12}$$

$$s.t.\sum_{i=1}^{N}\lambda_i y_i=0 \tag{13}$$

$$0\leq\lambda_i\leq C_i\ i=1,2\cdots,N \tag{14}$$

The dyadic problem can be solved by solving the following very, very small problem for the Lagrangian function:

$$L=\frac{1}{2}\|w\|^2+C\sum_{i=1}^{N}\xi_i-\sum_{i=1}^{N}\lambda_i\left[y_i(w\cdot x_i+b)-1+\xi_i\right]-\sum_{i=1}^{N}\mu_i\xi_i \tag{15}$$

where $\lambda_i\geq 0$, $\mu_i\geq 0$. There have been many optimization algorithms to solve convex optimization problems, in this paper, we use Sequential Minimum Optimization (SMO) algorithm, the basic idea of SMO algorithm is that if

the solutions of all variables satisfy the KKT condition of the optimization problem, then the solution of the optimization problem can be obtained, because the KKT condition is a sufficiently necessary condition of the optimization problem. Otherwise, the SMO algorithm solves the original problem by continuously decomposing it into subproblems and solving them. The subproblems have two variables, the one that violates the KKT condition the most and the other that is automatically determined by the constraints, fixing the other variables and constructing a quadratic programming problem for these two variables. Finally, the separating hyperplane can be derived as:

$$\sum_{i=1}^{N} \lambda_i y_i K(x_i, x_j) + b^* = 0 \tag{16}$$

A decision function is constructed:

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{N} \lambda_i y_i K(x_i, x_j) + b^* \right) \tag{17}$$

II. C. OVO-based AdaBoost-TWSVM multiple classification algorithm

II. C. 1) OVO-based TWSVM multiclassification

The OVA-based TWSVM multiclassification algorithm is a multiclassification algorithm obtained by combining the TWSVM with the OVA architecture, which requires the construction of a hyperplane for each class. In constructing the hyperplane corresponding to the samples of the C th class, OVA-TWSVM treats the C th class therein as one class and treats the other classes as one class to construct a TWSVM, which exacerbates the complexity of the algorithm to a certain extent since all the samples are required for each construction of the binary classifier.

Decision function of OVA-TWSVM:

$$Label(x) = arg \min_{C=1,2...c} \left(\frac{\left| x^T \omega_c + b_c \right|}{\sqrt{\omega_c^T \omega_c}} \right) \tag{18}$$

The decision function for the nonlinear case is:

$$Label(x) = \arg \min_{C=1,2...c} \left(\frac{\left| K(x^T, c^T) u_c + b_c \right|}{\sqrt{u_c^T K(c, c^T) u_c}} \right) \tag{19}$$

The working principle of the OVA-TWSVM based multicategorization algorithm is illustrated in Fig. 2 by solving a classification problem containing three categories in a two-dimensional space. There are three data points representing Label1, Label2 and Label3 in the figure. Applying the OVA-TWSVM method, we obtain three straight lines representing different categories, i.e., the hyperplanes of our proposed TWSVM. The common feature of these hyperplanes is that each hyperplane is as close as possible to the category it represents, while being far away from the other two categories. Thus, if a point to be categorized is close to the hyperplane of Label1 and far from the hyperplanes of Label2 and Label3, we categorize this point as Label1.

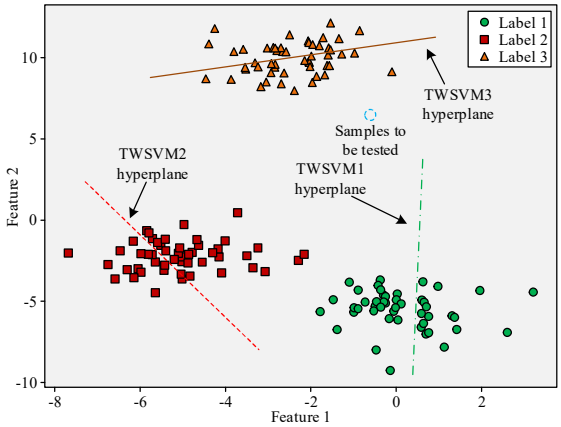


Figure 2: Working Principle of OVA-TwSVM multi-classification

II. C. 2) Improvement based on AdaBoost

(1) AdaBoost-TWSVM algorithm

Combining the advantages of AdaBoost algorithm with TWSVM as a weak classifier for training.

Combining TWSVM as a base classifier into the AdaBoost framework for classification combines the advantages of both techniques and can provide some significant benefits, especially when dealing with complex classification problems.TWSVM approximates the two classes separately by solving two optimization problems, which can capture the boundary between the classes more accurately, and combining it with AdaBoost can further improve the Classification accuracy is further improved by combining AdaBoost, because AdaBoost focuses on those samples that are difficult to classify correctly by iteratively adjusting the sample weights, so that the overall model is optimized after many iterations. At the same time, AdaBoost helps to reduce the risk of overfitting the model by combining multiple base classifiers to build a powerful classifier, using TWSVM as the base classifier, and the integrated model can be better generalized to unknown data because TWSVM itself has good generalization ability. On class-imbalanced datasets, AdaBoost makes the model focus more on misclassified samples by adjusting the sample weights, while the two optimization problems of TWSVM can be optimized for different classes separately, and this combination helps to improve the classification performance under class-imbalanced conditions.

This combination can also reduce the computational cost, although TWSVM needs to solve two optimization problems, it has lower computational complexity compared with traditional support vector machines, especially when dealing with large-scale datasets. Combined with AdaBoost, the overall training time can be effectively reduced by increasing the focus on only the current hardest classification samples in each round.

(2) AdaBoost-TWSVM algorithm based on OVO

Although the importance of algorithm complexity is decreasing with the improvement of computer storage technology and computing speed, the time complexity of the algorithm affects its utility in large-scale data classification problems as the size of data volume increases. Therefore, the algorithm in this paper explores new multiclassification algorithms with the main objective of reducing the algorithm complexity.

OVO-based AdaBoost-TWSVM algorithm, firstly constructs binary classifiers by AdaBoost-TWSVM, and constructs $C(C-1)/2$ binary classified AdaBoost-TWSVMs for the C class problem.After constructing all the binary subclassifiers, the voting method is used for the judgment of the classes.

III. Research on the performance level of integrated learning algorithm based on sample weight allocation mechanism

III. A. Experimental data and related settings

The experimental data in this paper are taken from UCI and KEEL databases, and the data features are collected on the basis of data features, which are categorized into low-dimensional low imbalance data (LL), low-dimensional high imbalance data (LH), and high-dimensional low imbalance data (HL) according to the height of dimension (the methodology makes use of the central limit theorem, and the high and low dimensionality cut-offs in this case are generally 30 dimensions) and the degree of imbalance, high-dimensional high imbalance data (HH) four categories, and the specific description of the data set is shown in Table 1. This experiment uses R software as the development environment. In order to facilitate clustering, the categorical variables present in the dataset were firstly treated as dummy variables, and the variables with non-uniform units were standardized.

Table 1: Description of the dataset

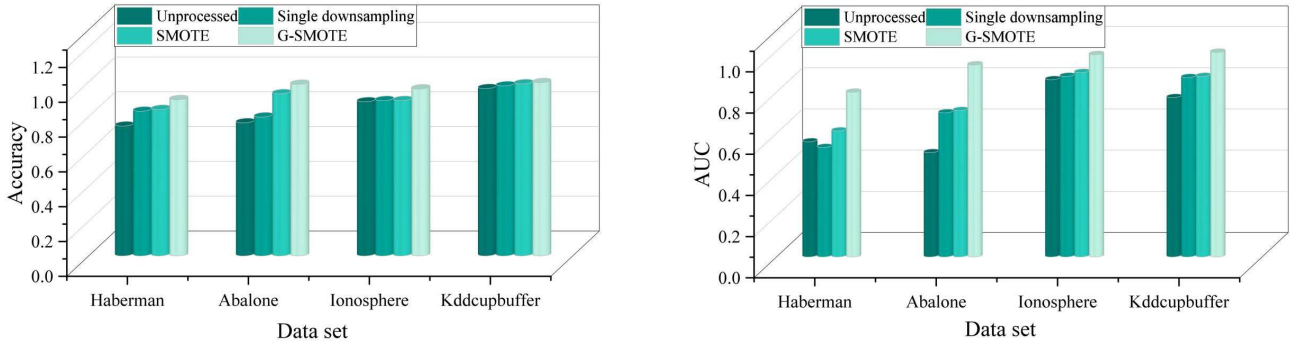
Data characteristics	Data set	Feature dimension	Negative sample size	Positive sample size	Imbalance ratio
LL	Haberman	4	217	78	2.78
LH	Abalone	6	2084	45	46.31
HL	Ionosphere	42	309	173	1.79
HH	Kddcupbuffer	51	2496	65	38.40

In order to get an objective demonstration of the classification performance, the experiment uses the eight-fold cross-validation method, which divides the dataset equally into eight parts, so that the first part is the test set and the remaining seven parts are the training set, and after conducting one experiment to get the performance results, so that the second part is the test set and the remaining part is the training set, and then the experiment is conducted again to get the results, and so forth to arrive at the results of the eight experiments, and finally, the results of the eight times will be averaged to arrive at the classification performance.

III. B. Validation of the effectiveness of the G-SMOTE algorithm

Due to the large difference in the imbalance ratio between the imbalanced datasets encountered in reality, in order to clarify the relationship between the classification accuracy of the classifier and the imbalance ratio of the classified samples after different sampling methods are processed, the four datasets are processed with single downsampling treatment, SMOTE over-sampling treatment, and G-SMOTE over-sampling treatment, respectively, and the Support Vector Machines are selected as the classification method. Accuracy and AUC value are the evaluation indexes, and the classification effect is compared with the unprocessed dataset, and the results of the classification accuracy comparison of the four datasets are shown in Fig. 3 (a~b).

It can be seen that the accuracies of different methods in the high unbalanced dataset mostly reach more than 80%, but the AUC values are relatively low. Compared with the classification effect of single downsampling and SMOTE oversampling, the G-SMOTE method proposed in this paper has an average AUC value of 0.891 on the 2 datasets with low imbalance ratios, which is higher than that of the original dataset, single downsampling, and SMOTE oversampling, respectively, by 0.181, 0.187, and 0.137, whereas, on the 2 datasets with high imbalance ratios, the average AUC value is 0.963, which is 0.323, 0.177, and 0.169 higher than the original dataset, single downsampling, and SMOTE oversampling, respectively.This indicates that when faced with high imbalance data, the classifiers mostly converge to the majority class, and that the single downsampling and SMOTE oversampling mostly sacrifices the AUC value in exchange for the increase in the accuracy rate, while the G-SMOTE oversampling maintains high accuracy while the AUC value also performs better.



(a)Accuracy (b)AUC

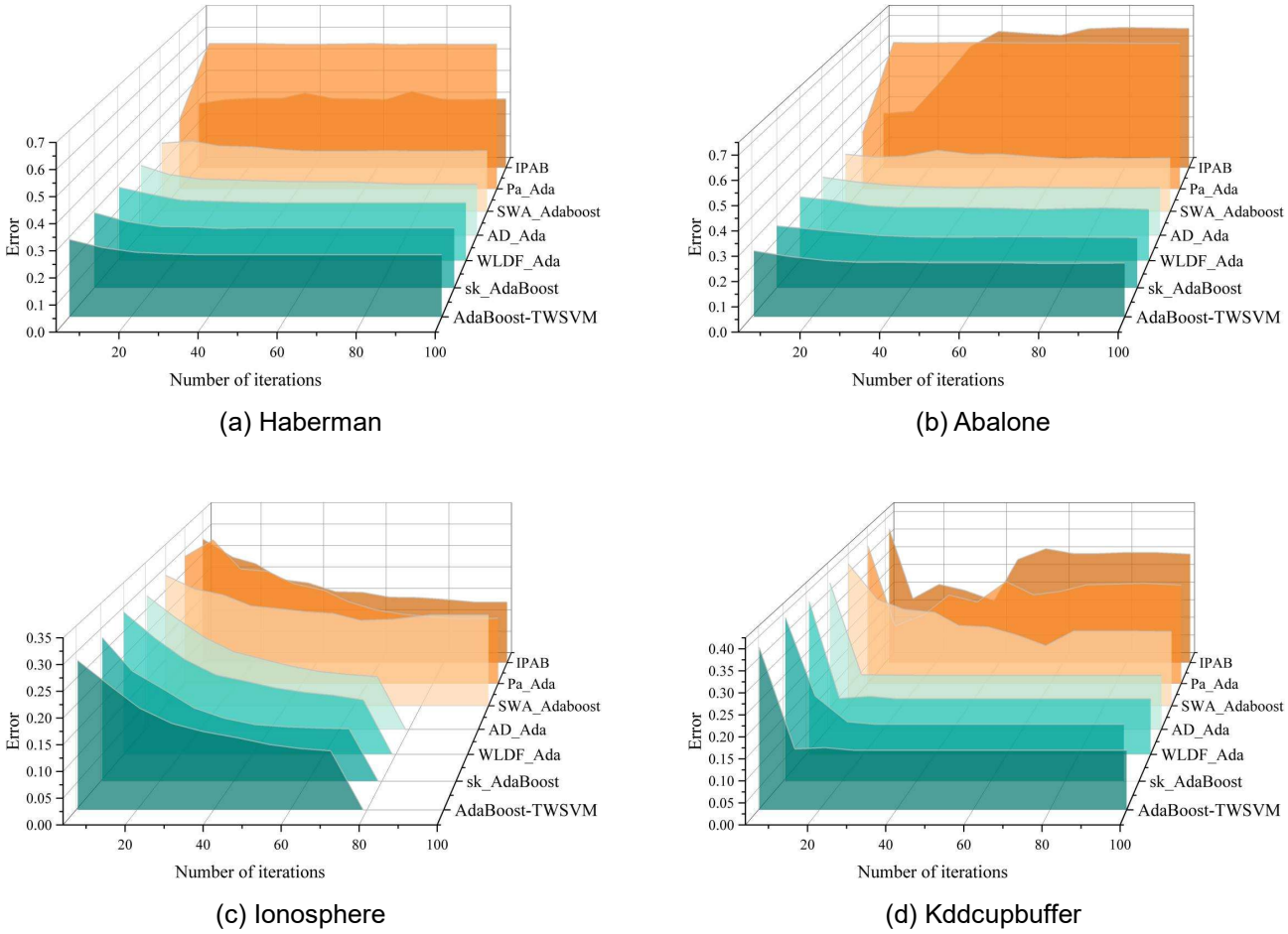
Figure 3: Comparison results of classification accuracy of four data sets

III. C. Effectiveness analysis of AdaBoost-TWSVM algorithm

III. C. 1) Convergence speed comparison

In order to verify the stability of the AdaBoost-TWSVM algorithm, the experiment uses eight-fold cross-validation to compare the average of eight training errors of six algorithms, namely, AdaBoost-TWSVM and sk_AdaBoost, WLDF_Ada, AD_Ada, SWA_Adaboost, Pa_Ada, and IPAB, with different number of iterations. The experimental results are shown in Fig. 4. On Haberman, AdaBoost-TWSVM, sk_AdaBoost, WLDF_Ada, AD_Ada, SWA_Adaboost converged after 46 iterations, Pa_Ada converged after 13 iterations, and IPAB converged after 80 iterations. On Abalone, AdaBoost-TWSVM, sk_AdaBoost, WLDF_Ada, AD_Ada converged after 32 iterations, Pa_Ada converged after 14 iterations, SWA_Adaboost converged after 70 iterations, and IPAB converged after 79 iterations. On Ionosphere, AdaBoost-TWSVM, sk_AdaBoost, WLDF_Ada, and AD_Ada simultaneously converged to 0 after 79 iterations, SWA_Adaboost, and Pa_Ada converged after 83 iterations, and only IPAB completed convergence after 90 iterations. On Kddcupbuffer, AdaBoost-TWSVM, sk_AdaBoost, WLDF_Ada, AD_Ada finished converging after 29 iterations, and IPAB, SWA_Adaboost, Pa_Ada converged after 71 iterations.

In summary, it can be concluded that the convergence speed of AdaBoost-TWSVM is almost indistinguishable from sk_AdaBoost, WLDF_Ada, AD_Ada, and has a certain advantage over Pa_Ada, and has a greater advantage over SWA_Adaboost and IPAB, which indicates that the AdaBoost-TWSVM algorithm in terms of training time achieves better results, and AdaBoost-TWSVM is not a bad choice in application scenarios that emphasize the speed of training.



(a) Haberman (b) Abalone

(c) Ionosphere (d) Kddcupbuffer

Figure 4: Comparison of convergence speed of the seven algorithms

III. C. 2) Comparison of test errors

In this experiment, the test error of AdaBoost-TWSVM is compared with other 6 different algorithms using eight fold cross validation, and the average of 8 test errors for each dataset is shown in Table 2, with the number of base classifiers being 100.

On the Haberman, Abalone, Ionosphere, and Kddcupbuffer datasets, the algorithms in this paper outperform the other six algorithms, with an average reduction of 9.22 percentage points on the Haberman dataset compared to the other six algorithms, a maximum reduction of 38.75 percentage points on the Abalone dataset, and an average reduction of 15.41 percentage points on the Ionosphere dataset. The average reduction is 6.08 percentage points on the Ionosphere dataset and 9.38 percentage points on the Kddcupbuffer dataset. The above experimental results show that the AdaBoost-TWSVM algorithm has better classification ability and generalization performance, which verifies the correctness of the design of this paper.

Table 2: Comparison of test errors of different algorithms

Data sequence number	Test error/%						
	AdaBoost-TWSVM	sk_AdaBoost	WLDF_Ada	AD_Ada	SWA_Adaboost	Pa_Ada	IPAB
1	22.49	24.38	23.52	26.77	28.12	59.47	27.99
2	20.18	23.64	22.78	24.16	27.77	56.28	58.93
3	14.66	16.75	17.15	18.26	21.59	20.96	29.71
4	4.17	5.22	5.18	5.63	14.85	20.11	30.29

To better illustrate the classification performance of the algorithms, the Friedman test for testing errors is shown in Figure 5. When comparing the performance of different algorithms on multiple datasets, the Friedman test based

on algorithm ordering can be used, and the experiment uses the Friedman test with a significant level of 0.05, and the AdaBoost-TWSVM with an average ordinal value of 1 significantly outperforms the remaining six algorithms in terms of classification error.

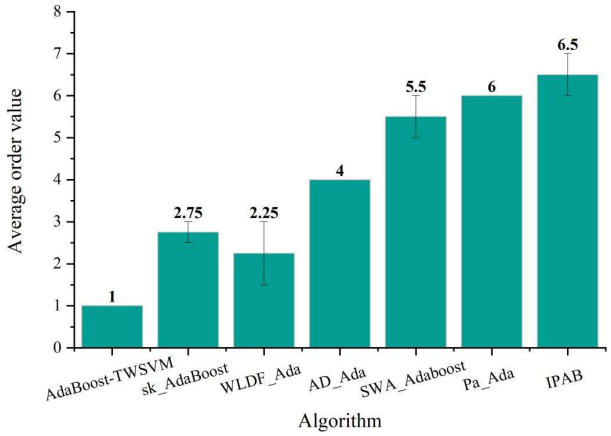


Figure 5: Friedman test of test errors

III. C. 3) Analysis of acceleration effects

In order to test the acceleration performance of the AdaBoost-TWSVM algorithm, this paper conducts experiments in a cluster environment using the dataset in Table 1, and the algorithm runtimes are counted, and the runtime results obtained are shown in Table 3. In order to minimize the impact of randomness on the experimental results, the data in the table are the mean ± standard deviation of the results of 10 experiments. From the time dimension, it is observed that with the increase of the number of computing nodes, the running time of each dataset shows a nonlinear decreasing trend. In the case of the Kddcupbuffer dataset, for example, when the number of nodes increases from 1 to 3, the running time decreases from 3189.46 to 1896.48 seconds, with a cumulative decrease of 40.5%, indicating that the algorithm is able to effectively cope with the computational demands of large-scale datasets.

Table 3: Running Time

Data set	1 node	2 node	3 node
Haberman	186.48±2.87	153.57±2.11	135.57±1.88
Abalone	198.84±2.44	166.86±2.03	144.26±1.96
Ionosphere	2978.36±18.38	2219.65±14.44	1642.86±12.48
Kddcupbuffer	3189.46±20.47	2503.67±15.93	1896.48±14.89

By running the program several times, the running time of AdaBoost-TWSVM and TWSVM algorithms under different datasets and different nodes is obtained, and the acceleration ratio of AdaBoost-TWSVM algorithm is calculated according to the acceleration ratio formula, and the results of the calculation are shown in Table 4. Compared with TWSVM, the acceleration ratios of AdaBoost-TWSVM algorithm are all improved to a certain extent, indicating that the execution efficiency of the algorithm is much better, and AdaBoost algorithm does shorten the running time, thus verifying the effectiveness of the improvement scheme in this paper. When the data size size is certain, the acceleration ratio is getting larger and larger with the increase of the number of nodes. The acceleration effect of Kddcupbuffer is most significant in the high-dimensional dataset Kddcupbuffer, and the acceleration ratio of node 3 reaches 2.37±0.03.

Table 4: Speedup Ratio

Data set	1 node	2 node	3 node
Haberman	1.50±0.50	1.31±0.08	1.54±0.03
Abalone	1.50±0.50	1.42±0.04	1.59±0.02
Ionosphere	1.50±0.50	1.93±0.07	2.26±0.04
Kddcupbuffer	1.50±0.50	2.01±0.05	2.37±0.03

III. D. Application in the classification of Zhang Daqian's early and late landscape works

Zhang Daqian's painting is a model of the integration of eastern and western painting techniques. His early and late works showed distinctive characteristics in the use of color and ink, which attracted wide attention at home and abroad. In this section, we use the G-SMOTE model to perform classification tests on Zhang Daqian's landscape paintings. We selected 110 early works represented by "Swamp Wilderness" (1946), "Autumn Clouds of Shudao" (1938), "Jiulongchi of Huangshan" (1930), "Zhongran Summer Mountain" (1944), "Huangshan" (1938) and "Surging River" (1938) (see Figure 6). And 119 late works represented by "Marks of Love" (1968), "Green Peak Cloud Waterfall" (1975), "Rain and Fog Return to Sail" (1973) (see Figure 7).

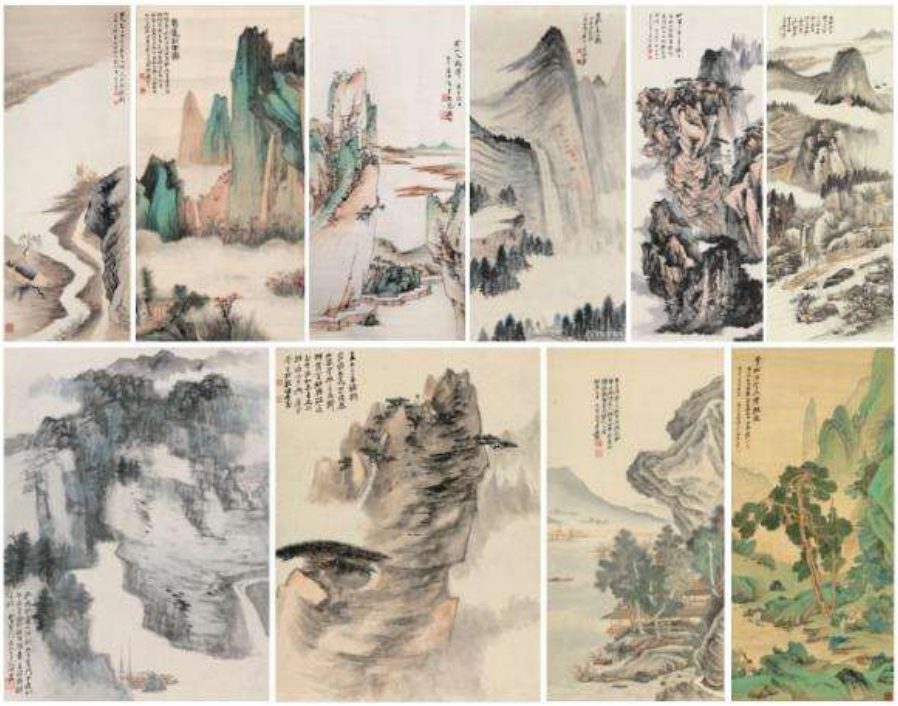


Figure 6: Zhang Daqian's early landscape painting representative works.

Figure 7: Zhang Daqian' slater landscape painting representative works.

We extracted 9 features including the mean, standard deviation, and skewness of RGB values;6 features from the mean and standard deviation of hue, saturation, and brightness; and 3 intercorrelation values from the color

correlation matrix (RG, RB, GB), totaling 18 features, Class 1 is Zhang Daqian's early landscape works, and 2 is his late landscape works. These feature data were applied to the G-SMOTE algorithm, which demonstrated the effectiveness and superiority of G-SMOTE in handling imbalanced data classification problems.

IV. Conclusion

Aiming at the common imbalance problem in multiclassification problems, this paper proposes the G-SMOTE algorithm for weight matching and designs the AdaBoost-TWSVM multiclassification algorithm based on OVO to improve the multiclassification performance.

Most of the different methods achieve more than 80% accuracy on the high imbalance dataset, but the AUC values are relatively low.The average AUC value of the G-SMOTE method on the 2 datasets with low imbalance ratios is 0.891, which is higher than that of the original dataset, the single downsampling, and the SMOTE over-sampling, respectively, by 0.181, 0.187, and 0.137.On the 2 datasets with high imbalance ratios, the average AUC value is 0.963, which is 0.323, 0.177, 0.169 higher than the original dataset, single downsampling, and SMOTE oversampling, respectively.The experiments demonstrate the effectiveness and superiority of G-SMOTE in dealing with unbalanced data classification problems.

AdaBoost-TWSVM convergence speed is almost indistinguishable from sk_AdaBoost, WLDF_Ada, AD_Ada, and has some advantage over Pa_Ada, and a large advantage over SWA_Adaboost and IPAB. The test error of AdaBoost-TWSVM is reduced by an average of 9.22 percentage points when comparing the other six algorithms on the Haberman dataset. The highest reduction on the Abalone dataset was 38.75 percentage points and the average reduction was 15.41 percentage points. The average reduction is 6.08 percentage points on the Ionosphere dataset and 9.38 percentage points on the Kddcupbuffer dataset. Compared with TWSVM, the acceleration ratios of AdaBoost-TWSVM algorithms are all improved to a certain extent, and the acceleration effect is most significant in the high-dimensional dataset Kddcupbuffer, with the acceleration ratio of node 3 reaching 2.37 ± 0.03. This classification algorithm can better improve the performance of image classification for artworks.

References

[1] Helm, J. M., Swiergosz, A. M., Haeberle, H. S., Karnuta, J. M., Schaffer, J. L., Krebs, V. E., ... & Ramkumar, P. N. (2020). Machine learning and artificial intelligence: definitions, applications, and future directions. Current reviews in musculoskeletal medicine, 13, 69-76.

[2] Kühl, N., Schemmer, M., Goutier, M., & Satzger, G. (2022). Artificial intelligence and machine learning. Electronic Markets, 32(4), 2235-2244.

[3] Mienye, I. D., & Sun, Y. (2022). A survey of ensemble learning: Concepts, algorithms, applications, and prospects. Ieee Access, 10, 99129-99149.

[4] Alam, K. M. R., Siddique, N., & Adeli, H. (2020). A dynamic ensemble learning algorithm for neural networks. Neural Computing and Applications, 32(12), 8675-8690.

[5] Cao, Y., Geddes, T. A., Yang, J. Y. H., & Yang, P. (2020). Ensemble deep learning in bioinformatics. Nature Machine Intelligence, 2(9), 500-508.

[6] Syamsuddin, A. (2023). Designing Blended Learning Program for Mathematical Economics Using Integrative Learning Design Framework Approach. Galore International Journal of Applied Sciences and Humanities, 7(4), 30-46.

[7] Forouzandeh, S., Berahmand, K., & Rostami, M. (2021). Presentation of a recommender system with ensemble learning and graph embedding: a case on MovieLens. Multimedia tools and applications, 80(5), 7805-7832.

[8] Yu, L., Zhou, R., Tang, L., & Chen, R. (2018). A DBN-based resampling SVM ensemble learning paradigm for credit classification with imbalanced data. Applied Soft Computing, 69, 192-202.

[9] Feng, W., Huang, W., & Ren, J. (2018). Class imbalance ensemble learning based on the margin theory. Applied Sciences, 8(5), 815.

[10] Tanha, J., Abdi, Y., Samadi, N., Razzaghi, N., & Asadpour, M. (2020). Boosting methods for multi-class imbalanced data classification: an experimental review. Journal of Big data, 7, 1-47.

[11] You, G. R., Shiue, Y. R., Yeh, W. C., Chen, X. L., & Chen, C. M. (2020). A weighted ensemble learning algorithm based on diversity using a novel particle swarm optimization approach. Algorithms, 13(10), 255.

[12] Yao, J., Wang, Z., Wang, L., Liu, M., Jiang, H., & Chen, Y. (2022). Novel hybrid ensemble credit scoring model with stacking-based noise detection and weight assignment. Expert Systems with Applications, 198, 116913.

[13] Liu, X., Liu, Z., Wang, G., Cai, Z., & Zhang, H. (2017). Ensemble transfer learning algorithm. Ieee Access, 6, 2389-2396.

[14] Li, L., Hu, Q., Wu, X., & Yu, D. (2014). Exploration of classification confidence in ensemble learning. Pattern recognition, 47(9), 3120-3131.