Article:
Li, Y, Zhang, J, Zhang, S et al. (2 more authors) (2022) Multi-objective optimization-based adaptive class-specific cost extreme learning machine for imbalanced classification. Neurocomputing, 496. pp. 107-120. ISSN 0925-2312

https://doi.org/10.1016/j.neucom.2022.05.008

Multi-objective optimization-based adaptive class-specific cost extreme learning machine for imbalanced classification

Yanjiao Li[a,b], Jie Zhang[c,d,*], Sen Zhang[c], Wendong Xiao[c], Zhiqiang Zhang[d]

[a]*Institute of Engineering Technology, University of Science and Technology Beijing, Beijing 100083, China*
[b]*National Engineering Research Center for Advanced Rolling and Intelligent Manufacturing, University of Science and Technology Beijing, Beijing 100083, China*
[c]*School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China*
[d]*School of Electronic and Electrical Engineering, University of Leeds, Leeds LS2 9JT, U.K.*

Abstract

Imbalanced classification is a challenging task in the fields of machine learning and data mining. Cost-sensitive learning can tackle this issue by considering different misclassification costs of classes. Weighted extreme learning machine (W-ELM) takes a cost-sensitive strategy to alleviate the learning bias towards the majority class to achieve better classification performance. However, W-ELM may not achieve the optimal weights for the samples from different classes due to the adoption of empirical costs. In order to solve this issue, multi-objective optimization-based adaptive class-specific cost extreme learning machine (MOAC-ELM) is presented in this paper. To be Specific, the initial weights are first assigned depending on the class information. Based on that, the representation of the minority class could be enhanced by adding penalty factors. In addition, a multi-objective optimization with respect to penalty factors is formulated to automatically determine the class-specific costs, in which multiple performance criteria are constructed by comprehensively considering the misclassification rate and generalization gap. Finally, ensemble strategy is implemented to make decisions after optimization. Accordingly, the proposed MOAC-ELM is an adaptive method with good robustness and generalization performance for imbalanced classification problems. Comprehensive experiments have been performed on several benchmark datasets and a real-world application dataset. The statistical results demonstrate that MOAC-ELM can achieve competitive results on classification performance.

Keywords: Imbalanced classification, cost-sensitive learning, extreme learning machine, multi-objective optimization

1. Introduction

Class imbalance means that the number of samples of different classes varies greatly, i.e., data present skewed class distributions [1, 2]. This phenomenon pervasively exists in a variety of real-world applications, such as financial distress prediction [3], human activity recognition [4],

*Corresponding author
Email address: zhangjie_saee@ustb.edu.cn (Jie Zhang)

and abnormal condition detection [5], etc. The regular classifier constructed using imbalanced data usually leads to the decision boundary closer to the minority classes, degrading the classification performance on the minority classes [6, 7]. However, in many scenarios, minority classes are more important and concern ones to be recognized than majority classes, such as disease diagnosis in the medical domain and fault diagnosis in the industrial domain [8].

Several efforts have been made to develop specific classifiers that could classify minority classes more accurately, involving decision tree [9], random forest [10], k-nearest neighbors [11], naive Bayes [12], and support vector machine (SVM) [13]. These algorithms are essentially effective, but encounter stopping criteria, learning rate, local minima and time-consuming issues.

In the past decade, extreme learning machine (ELM) and its variants have been proved to be very powerful tools to deal with classification and regression problems [14, 15, 16, 17, 18]. Unlike other learning algorithms, the most advanced features of ELM are that its hidden layer parameters are independent of training data and the learning procedure is non-iterative. There have been also a lot of interests in solving the imbalanced classification problem using ELMs. For example, total error rate ELM (TER-ELM) [19] assigned adjustable parameters for training samples but could not be applied in multiclass classification directly. Weighted ELM (W-ELM) [20] provided two weighting schemes for training samples from different classes to strengthen the performance of the minority classes. Zhang et al. [21] presented a fuzzy matrix to ELM to highlight the contributions of different inputs. Li et al. [22] explored a modified AdaBoost framework and its distribution weights replaced the training samples weights in W-ELM to enhance the classification performance. Class-specific cost regulation ELM (CCR-ELM) was proposed [23], in which class-specific cost regularization was used for misclassification of each class. Different from CCR-ELM in the computation of output weights, cost-specific kernelized ELM (CSKELM) [24] was presented with lower computational overhead. Label-weighted ELM (LW-ELM) [25] extended the class label of the the minority class samples and two weighing rules were designed based on class information. Li et al. [26] proposed a sparse cost matrix for W-ELM. The aforementioned classifiers are based on the concept of cost-sensitive learning, which has been frequently used for handling imbalanced data, and could achieve satisfactory performance in several tasks. However, a good consistency between the classification performance and the design of the cost is difficult to identify, because the empirical cost only associates with the class imbalance ratio (CIR) and neglects the specific data distribution. Thus, modeling quality of the classifier may not be excellent enough. In addition, with the unknown data distribution, the classifier also suffers from the overfitting problem due to the unclear relationship between the weighting scheme and generalization performance. Therefore, it is desirable to develop a weighting scheme utilizing an adaptive method with the purpose of simultaneously achieving the optimal classification performance and alleviating the overfitting problem.

In view of the aforementioned considerations, we propose a multi-objective optimization-based adaptive class-specific cost ELM (MOAC-ELM) to promote the classification performance of ELM for handling imbalanced classification problems. MOAC-ELM has a flexible and efficient weighting scheme for different tasks. In this scheme, considering the multiple classification performance criteria, a multi-objective optimization problem is formulated to learn the optimal weights to enhance the generalization performance. Through the comparisons with state-of-the-art methods on selected benchmark datasets and a real-world application dataset, MOAC-ELM shows its superiority in solving the imbalanced classification problem. Specifically, the primary contributions of MOAC-ELM include the following aspects:

1) Without any data distribution information, a flexible and efficient weighting scheme is developed. This scheme not only considers the CIR, but also introduces a diagonal penalty

adjustment matrix for different classes to strengthen the representation of data distribution.

2) In order to achieve the penalty adjustment factors in an adaptive manner, a multi-objective optimization problem is developed to learn the parameters to enhance the generalization performance. Misclassification rate and generalization gap are defined as the two conflicting optimization objectives to improve the classification performance and alleviate the overfitting problem simultaneously.

3) The non-dominated sorting genetic algorithm version II (NSGA-II) [27] is applied to globally optimize the penalty factors. Moreover, ensemble strategy is implemented to make a decision from the multi-objective optimization solutions.

The remainder of this paper is organized as follows. Section 2 introduces preliminaries, including solutions for imbalanced classification, details of W-ELM and evaluation metrics for imbalanced classification. Section 3 presents the problem formulation. The proposed MOAC-ELM is detailed in Section 4. Section 5 presents the experimental results and further analysis. This is followed by discussions in Section 6. Finally, Section 7 gives the conclusions of our research work.

2. Preliminaries

This section briefly reviews state-of-the-art methods addressing imbalanced classification, W-ELM, and evaluation metrics for imbalanced classification, to provide the necessary background for the proposed MOAC-ELM.

2.1. Solutions for imbalanced classification

Imbalanced data distribution includes several cases as illustrated in Fig. 1, in which the red diamonds represent the minority class and the green filled stars stand for the majority class. Class imbalance is affected by many factors, including imbalance ratio, class surrounding, class overlapping and small disjuncts, etc. It has been proved that the kind of data complexity is the main determinant of classification performance reduction [6]. When regular classifiers are employed to imbalanced data, the decision boundary usually biased towards the minority classes, since the minority classes are underrepresented. Many techniques have been developed to tackle this issue, which could be grouped as data-level methods and algorithm-level methods.

Data-level methods essentially utilize the preprocessing to rebalance the class distribution. Different forms of resampling methods, i.e., oversampling and undersampling, try to extract balanced samples from different classes. Oversampling methods remove some samples corresponding to the majority class while undersampling methods expand samples corresponding to the minority class [28, 29, 30]. The salient feature of resampling is that it is independent with classifier, making it more versatile [31]. However, the matter is how to determine the appropriate class distribution for a skewed distribution. Oversampling may suffer from the overfitting issue while undersampling may encounter the information loss.

Algorithm-level methods generally attempt to establish a classification algorithm that can provide a better classification performance for class imbalance problems than regular classifier. It can be mainly divided as cost-sensitive learning [32], ensemble strategy [33], and one-class learning [34]. Among them, cost-sensitive learning is the most commonly used strategy, which introduces the misclassification costs and assigns higher misclassification costs to minority samples than majority ones. By utilizing this strategy, there are several improved variants in the ELM community, such as W-ELM [20], AdaBoost W-ELM [22], CCR-ELM [23], CSKELM

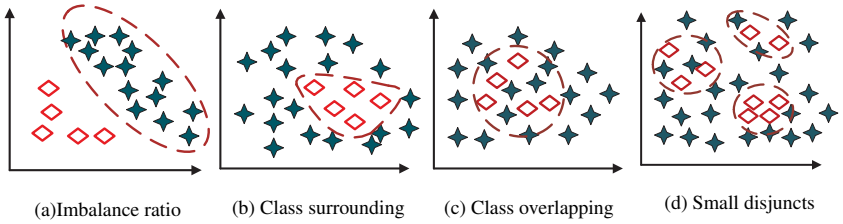


Figure 1: Illustration of different imbalanced data distributions with two-dimensional binary-class data

[24], and Evolutionary ELM [26], etc. However, an appropriate misclassification cost matrix is still difficult to be estimated.

2.2. *Brief review of W-ELM*

ELM, a single hidden layer feedforward neural network shown in Fig. 2, can be used for both regression and classification [35]. The major differences are the assignment of hidden layer parameters and learning process without iteration in comparison with other traditional learning methods [36, 37]. Overall, due to the random generation of hidden layer parameters, ELM has faster training process and better generalization performance. However, ELM treats all the samples equally, making it not suitable for the imbalanced classification problems.

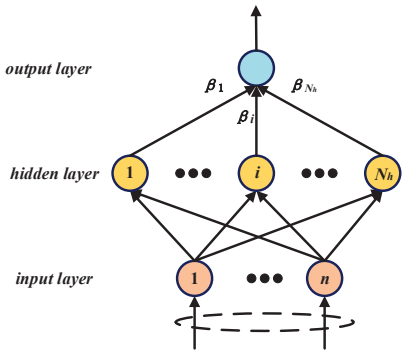


Figure 2: ELM network architecture [38]

W-ELM is an effective solution to deal with the imbalanced data, which can be considered as a cost-sensitive version of ELM. The essence of W-ELM is to assign different weights to the samples from different classes. The minority class employs larger weights to strengthen its influence. On the contrary, the majority class employs smaller weights to weaken its relative influence. Mathematically, we define a $N \times N$ diagonal matrix $\mathbf{W}$ to allocate the weights for each training sample $\mathbf{x}_i$. In this case, W-ELM actually attaches more significance to the training errors of the minority class, making the decision boundary emerge in a more impartial position.

Embedding the diagonal matrix $\mathbf{W}$, we have an optimization problem represented as

$$\begin{aligned} \min :& J_{W-ELM} = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\mathbf{W}\frac{1}{2}\sum_{i=1}^{N}\|\mathbf{e}_i\|^2 \\ s.t.,& \mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} = \mathbf{y}_i^T - \mathbf{e}_i^T \end{aligned} \tag{1}$$

where $(\mathbf{x}_i, \mathbf{y}_i) \in R^n \times R^m$ is a given training dataset with N samples, $\boldsymbol{\beta}$ is the output weight vector, $\mathbf{e}_i$ is the training error, C is a tradeoff parameter and can be determined by users according to the specific tasks, and $\mathbf{h}(\mathbf{x}_i)$ is the mapped feature vector with respect to input $\mathbf{x}_i$.

Similar to ELM, we can achieve the solution of Eq.(1):

$$\boldsymbol{\beta} = \begin{cases} \mathbf{H}^T\left(\frac{\mathbf{I}}{C} + \mathbf{W}\mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{W}\mathbf{Y}, N < N_h \\ \left(\frac{\mathbf{I}}{C} + \mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{W}\mathbf{Y}, N \geq N_h \end{cases} \tag{2}$$

Two empirical weighting schemes of W-ELM are

$$\begin{aligned} W1 &: w_{ii} = \frac{1}{\#n_i} \\ W2 &: w_{ii} = \begin{cases} \frac{0.618}{\#n_i} & if \ \ \#n_i > \bar{A}(\#n_i) \\ \frac{1}{\#n_i} & if \ \ \#n_i \leq \bar{A}(\#n_i) \end{cases} \end{aligned} \tag{3}$$

where $\#n_i$ represents the sample size belonging to class n_i, $\bar{A}(\#n_i)$ is the average sample size of all the classes.

The classification error of the minority class is reduced by allocating a different cost distribution to each class. It should be noted that W-ELM maintains the advantages of ELM, and is competent for both binary classification and multiclass classification tasks.

2.3. Performance metrics for imbalanced classification

Evaluation metrics are indispensable for classification performance evaluation and classifier construction guidance [7]. Considering a binary classification problem, confusion matrix is defined to represent the classification performance, as shown in Table 1. The minority class is considered as positive class and the majority class is called negative class. After classification process, samples are divided into four parts, i.e., TP, FP, FN and TN.

Table 1: Confusion matrix

	Predicted as Positive	Predicted as Negative
Actually Positive	True Positive (TP)	False Negative (FN)
Actually Negative	False Positive (FP)	True Negative (TN)

The most frequently adopted evaluation metrics for the binary imbalanced classification problems include accuracy, precision (P), recall (R), F-measure (F) and G-mean:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4}$$

$$Precision\,(P) = \frac{TP}{TP + FP} \tag{5}$$

$$Recall\,(R) = \frac{TP}{TP + FN} \tag{6}$$

$$F - measure\,(F) = \frac{2PR}{P + R} \tag{7}$$

$$G - mean_b = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}} \tag{8}$$

Different metrics can characterize the classification performance from different aspects. In general, accuracy provides a simple manner to quantify the performance of a classifier while it is sensitive to data changes in some cases. Precision is used to measure the exactness, recall is used to evaluate the completeness. There is an opposite relationship between precision and recall. In particular, F-measure is a more comprehensive measure by combining precision and recall. Another indicator, G-mean, represents the integral assessment of positive accuracy and negative accuracy.

For the multiclass classification problem, racall of the *i*th class is expressed as

$$R_i = \frac{n_{ii}}{\sum_{j=1}^{k} n_{ij}} \tag{9}$$

where n_{ij} denotes the number of samples belonging to class i but classified as class j.

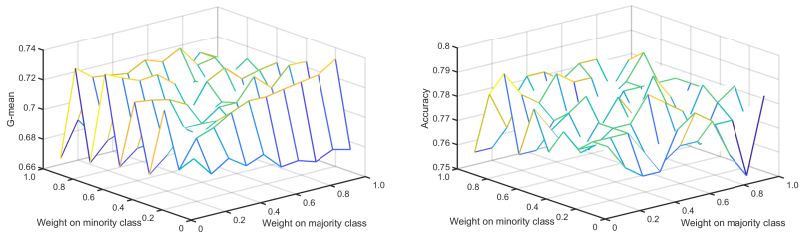
G-mean for the multiclass classification problem is given by

$$G - mean_m = \left(\prod_{i=1}^{k} R_i \right)^{\frac{1}{k}} \tag{10}$$

3. Problem Formulation

Cost matrix is the key of cost-sensitive learning, which encodes the penalty for misclassified samples. Minority class samples are usually assigned higher misclassification costs to enhance their representation. Although many promising progresses have been made utilizing this strategy in the ELM community, the existing works highly rely on empirical weighting schemes without considering data distribution or have a large amount of calculation in finding appropriate cost. More concretely, the weights were determined by the number of elements in each class in W-ELM [20]. CSKELM adopted the class specific regularization parameter, which was designed based on class proportion for handling binary classification problem with imbalanced data [24]. CCR-ELM performed grid search to adjust the regularization parameters, leading to its high computational complexity [23]. LW-ELM designed two types of weight allocation strategies based on CIR [25]. The aforementioned works adopt fixed weight mechanism by considering the percentage of samples belonging to a certain class. The fixed weight is lack of flexibility and will miss a better selection to achieve better classification performance. In addition, CIR is not the only factor deteriorating the classifier performance, and also involving other factors, such as class overlapping and small disjunction (see Fig. 1) [39]. However, we lack the priori information on data distribution in most scenarios. Thus, it is difficult to define the relationship between weight setting mechanism and classification performance. Fig. 3 depicts the relationship between weights and classification performance on a specific imbalanced dataset (the sample number of the majority class is 490 and that of the minority class is 260). From Fig. 3, we

can find that the best accuracy is 79.17% when the weight on the majority class is 0.1 and the weight on the minority class is 0.8. In this sense, the original weight setting mechanism based on the percentage of samples cannot obtain the best performance. Furthermore, the classification performance is constantly changing as weights vary. Accordingly, the costs in cost-sensitive learning play important roles in performance improvement. How to find the appropriate cost matrix for a specific task is a meaningful and challenging issue.



(a) (b)

Figure 3: Relationship between weights and classification performance on a specific imbalanced dataset

Moreover, when building an operational mechanism for evaluating the suitability of cost matrix, we are usually confused by the lack of practical guidelines. One of the frequently used schemes is to construct an optimization problem where accuracy-related function is selected as the objective function, but the classifier may suffer from the overfitting problem in this scenario. Additional criterion should be considered to enhance the generalization performance.

Accordingly, it is worthwhile to develop a more robust and adaptive determination scheme fully considering the class information and data complexity during its modeling process to achieve superior classification performance.

4. Methodology

This section gives the details of the proposed MOAC-ELM for imbalanced classification problems, covering main framework, classifier construction, multi-objective optimization for penalty adjustment factors, decision making via ensemble strategy and MOAC-ELM algorithm description.

4.1. Main framework of MOAC-ELM

Focusing on the aforementioned practical problems in imbalance learning, we propose MOAC-ELM to determine the optimal misclassification loss without the need of prior knowledge and many trials. As depicted in Fig. 4, MOAC-ELM can be achieved via the following three phases: 1) Define a penalty adjustment matrix and derive the output weights; 2) Establish the multi-objective optimization model and search the optimal penalty adjustment factors; 3) Employ the ensemble strategy for decision making from the obtained Pareto optimal solutions to output the optimal output weights.

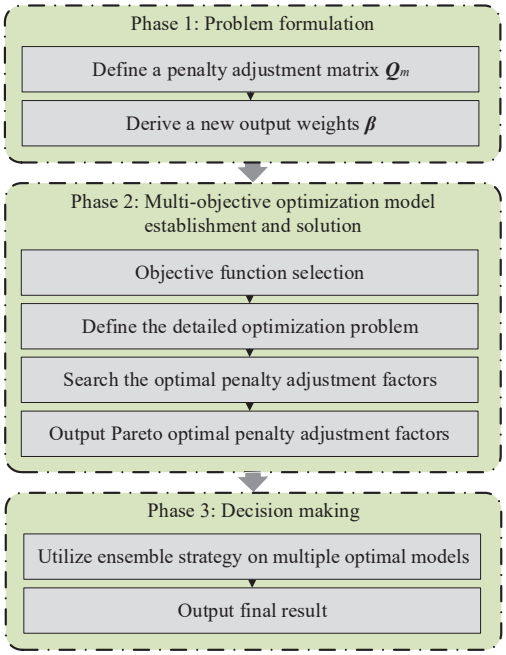


Figure 4: Main framework of the proposed MOAC-ELM

4.2. Classifier construction

As previously mentioned, existing works mainly depend on empirical weighting schemes, which are determined by the number of elements in each class without considering data complexity and other factors. In a sense, they can be called experienced domain and the global optimum cannot be guaranteed. In order to effectively address class imbalance problem, we embed an imbalance representation matrix $\mathbf{W}_m$ and a penalty adjustment matrix $\mathbf{Q}_m$ into ELM to make the decision boundary locate in the ideal position. $\mathbf{Q}_m$ is a $N \times N$ diagonal matrix, in which the element Q_m^{ii} denotes the correction value associated with the ith sample and $Q_m^{ij} = 0$ $(i \neq j)$. The number of different element types in penalty adjustment matrix is consistent with the number of classes. Define the set of penalty adjustment factors as $P = \{p_1, p_2, \ldots, p_k\}$, k is the number of classes, $Q_m^{ii} = p_k$ means that the ith sample belongs to the kth class. $\mathbf{W}_m$ is a $N \times N$ diagonal matrix, in which the element W_m^{ii} denotes the class proportion. Based on $\mathbf{W}_m$ and $\mathbf{Q}_m$, the objective function of MOAC-ELM is mathematically represented as

$$\begin{aligned} \min :& J_{MOAC-ELM} = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\left(\mathbf{W}_m + \mathbf{Q}_m\right)\frac{1}{2}\sum_{i=1}^{N}\|\mathbf{e}_i\|^2 \\ s.t.,& \mathbf{h}\left(\mathbf{x}_i\right)\boldsymbol{\beta} = \mathbf{y}_i^T - \mathbf{e}_i^T \end{aligned} \tag{11}$$

Here we use $\mathbf{W}_m$ and $\mathbf{Q}_m$ as the regulation costs for misclassification caused by class proportion and data complexity.

Based on Karush-Kuhn-Tucker (KKT) theorem [40], the Lagrangian function of Eq.(11) is

$$L = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\left(\mathbf{W}_m + \mathbf{Q}_m\right)\frac{1}{2}\sum_{i=1}^{N}\|\mathbf{e}_i\|^2 - \sum_{i=1}^{N}\alpha_i\left(\mathbf{h}\left(\mathbf{x}_i\right)\boldsymbol{\beta} - \mathbf{y}_i^T + \mathbf{e}_i^T\right) \tag{12}$$

where α_i is the Lagrangian multiplier.

The KKT optimization conditions of Eq.(12) are as follows:

$$\frac{\partial L}{\partial \boldsymbol{\beta}} = 0 \rightarrow \boldsymbol{\beta} = \sum_{i=1}^{N}\boldsymbol{\alpha}_i\mathbf{h}(\mathbf{x}_i)^T = \mathbf{H}^T \tag{13a}$$

$$\frac{\partial L}{\partial \mathbf{e}_i} = 0 \rightarrow \boldsymbol{\alpha}_i = C\left(\mathbf{W}_m + \mathbf{Q}_m\right)\mathbf{e}_i, i = 1, \ldots, N \tag{13b}$$

$$\frac{\partial L}{\partial \boldsymbol{\alpha}_i} = 0 \rightarrow \mathbf{h}\left(\mathbf{x}_i\right)\boldsymbol{\beta} - \mathbf{y}_i^T + \mathbf{e}_i^T = 0, i = 1, \ldots, N \tag{13c}$$

Based on Eq.(13), we can obtain the output weights:

$$\boldsymbol{\beta} = \begin{cases} \mathbf{H}^T\left(\frac{\mathbf{I}}{C} + \left(\mathbf{W}_m + \mathbf{Q}_m\right)\mathbf{H}\mathbf{H}^T\right)^{-1}\left(\mathbf{W}_m + \mathbf{Q}_m\right)\mathbf{Y}, N < N_h \\ \left(\frac{\mathbf{I}}{C} + \mathbf{H}^T\left(\mathbf{W}_m + \mathbf{Q}_m\right)\mathbf{H}\right)^{-1}\mathbf{H}^T\left(\mathbf{W}_m + \mathbf{Q}_m\right)\mathbf{Y}, N \geq N_h \end{cases} \tag{14}$$

Finally, MOAC-ELM classifier can be constructed:

$$f\left(\mathbf{x}\right) = \begin{cases} sign\ \mathbf{h}\left(\mathbf{x}\right)\mathbf{H}^T\left(\frac{\mathbf{I}}{C} + \left(\mathbf{W}_m + \mathbf{Q}_m\right)\mathbf{H}\mathbf{H}^T\right)^{-1}\left(\mathbf{W}_m + \mathbf{Q}_m\right)\mathbf{Y}, N < N_h \\ sign\ \mathbf{h}\left(\mathbf{x}\right)\left(\frac{\mathbf{I}}{C} + \mathbf{H}^T\left(\mathbf{W}_m + \mathbf{Q}_m\right)\mathbf{H}\right)^{-1}\mathbf{H}^T\left(\mathbf{W}_m + \mathbf{Q}_m\right)\mathbf{Y}, N \geq N_h \end{cases} \tag{15}$$

The essence of MOAC-ELM is to determine $\mathbf{W}_m$ and $\mathbf{Q}_m$ to push the decision boundary tend to the majority class and further minimize the weighted cumulative training error. Hence, we provide $\mathbf{W}_m$ based on class information, which can be represented as

$$w_{ii} = \frac{\#n_i}{\sum_{i=1}^{k}\left(\#n_i\right)} \tag{16}$$

where $\#n_i$ denotes the number of the samples corresponding to class n_i, k is the number of classes. In this case, the weights between the different classes reflect the CIR. Meanwhile, data complexity has its own characteristics in different tasks. Thus, $\mathbf{Q}_m$ should be set according to the case-specific situations, which will be discussed below.

Under the unknown feature mapping $\mathbf{h}\left(\mathbf{x}\right)$ circumstances, ELM kernel matrix is defined as

$$\boldsymbol{\Omega}_{ELM} = \mathbf{H}\mathbf{H}^T : \Omega_{ELMi,j} = h\left(\mathbf{x}_i\right)\cdot h\left(\mathbf{x}_j\right) = K\left(\mathbf{x}_i, \mathbf{x}_j\right) \tag{17}$$

Inspired by the definition of ELM kernel matrix, the output function of kernel-based MOAC-ELM can be represented as

$$\begin{aligned} f\left(\mathbf{x}\right) &= sign\ \mathbf{h}\left(\mathbf{x}\right)\mathbf{H}^T\left(\frac{\mathbf{I}}{C} + \left(\mathbf{W}_m + \mathbf{Q}_m\right)\mathbf{H}\mathbf{H}^T\right)^{-1}\left(\mathbf{W}_m + \mathbf{Q}_m\right)\mathbf{Y} \\ &= sign\begin{bmatrix} K\left(\mathbf{x}, \mathbf{x}_1\right) \\ \vdots \\ K\left(\mathbf{x}, \mathbf{x}_N\right) \end{bmatrix}^T\left(\frac{\mathbf{I}}{C} + \left(\mathbf{W}_m + \mathbf{Q}_m\right)\boldsymbol{\Omega}_{ELM}\right)^{-1}\left(\mathbf{W}_m + \mathbf{Q}_m\right)\mathbf{Y} \end{aligned} \tag{18}$$

***Remark* 1 :** CIR is generally easy to obtain. It is suppose that we have priori information about the CIR in the proposed method. Besides, data distribution information is difficult to be identified and the penalty adjustment matrix is added to capture this issue. Thereby, MOAC-ELM fully considers both class proportion and data complexity.

***Remark* 2 :** The number of different element types in $\mathbf{Q}_m$ is consistent with the number of classes, thus there are less tuned parameters, making the proposed method more convenient to be implemented for multiclass classification problem. Taken altogether, MOAC-ELM is suitable for both binary classification and multiclass classification problems. Moreover, hidden layer activation function or kernel function are free to select to map the raw data into the feature space.

4.3. Multi-objective optimization model for penalty adjustment factors

In order to adapt to data complexity for different applications and determine the penalty adjustment factors in an adaptive manner, a multi-objective optimization-based determination mechanism is developed. In this mechanism, two main points should be considered for parameter optimization before applying any optimizer. The first one is how to design individual encoding using penalty adjustment factors and the second one is how to select the appropriate fitness function.

As previously mentioned, the individual of the multi-objective optimization model consists of the set of penalty adjustment factors $P = \{p_1, p_2, \ldots, p_k\}$. Then the individual can be defined as

$$\chi = [p_1, p_2, \ldots, p_k] \tag{19}$$

where k is the number of classes. All parameters are randomly initialized in the reasonable bounds. Then, the iterative process is triggered to evolve the initialized population.

In order to evaluate the quality of each generated solution by the optimization algorithm, a suitable optimization objective (i.e., fitness function) should be selected. As the widely used model performance measure, misclassification rate refers to the proportion of all samples with falsely classified samples. It is utilized as one of the objective:

$$f_1 = 1 - \frac{NCC}{N} \tag{20}$$

where N is the number of samples, and NCC indicates the number of correctly classified samples, which can be expressed as

$$NCC = \sum_{i=1}^{N} \sum_{j=1}^{k} f(i, j)\kappa(i, j) \tag{21}$$

where $f(i, j)$ and $\kappa(i, j)$ are the indicator functions that return 0 or 1. If the sample i belongs to class j, then $f(i, j) = 1$, if the predicted class of sample i is j, $\kappa(i, j) = 1$.

In addition, a good algorithm is expected to achieve good generalization performance (i.e., perform well on unseen data) [26, 41]. Kawaguchi et al. [41] identified the practical roles for generalization theory, in which the most crucial point is that it can provide theoretical insights to guide of searching model parameters. Therefore, we take into account the generalization gap during the optimization. Generalization gap is defined as the difference in the model performance on training samples versus testing samples, which is given by

$$GG = |\ell(f, s_{train}) - \ell(f, s_{test})| \tag{22}$$

where $\ell(f, s_{train})$ represents the training accuracy of the model, and $\ell(f, s_{test})$ is the corresponding testing accuracy.

A model with a larger generalization gap means overfitting training samples. Inspired by this, two nonoverlapped subsets are extracted from the training samples, in which one is for training and the other for validation. Generalization gap is considered as the other objective, which is defined as

$$f_2 = |\ell(f, s_{train}) - \ell(f, s_{val})| \tag{23}$$

where $\ell(f, s_{val})$ is the validation accuracy.

We propose a multi-objective optimization model for penalty adjustment factors by minimizing the above objectives:

$$\min F(P) = \{f_1(P), f_2(P)\} \tag{24}$$

Multi-objective evolutionary algorithms (MOEAs) are the mainstream algorithm for solving the multi-objective problem by providing the Pareto optimal solutions. Among them, NSGA-II [27] is one of the most famous and successful approaches, which has excellent multi-objective search capability with elitist strategy and diversity preservation mechanism. Its distinguishing features are computationally efficient and less dependent on the sharing parameters. Accordingly, NSGA-II is adopted to solve this multi-objective optimization model to generate the Pareto optimal solutions. The details are described in Algorithm 1.

Algorithm 1 NSGA-II Algotirhm

Input: Initial parent population P_0, N_{pop} size of population and maximal number of iterations $It_{\max}$.
Output: Pareto optimal solutions

1: Sort P_0 based on non-domination criteria.
2: Evaluate the fitness of the initialized population by the defined objective function.
3: Create offspring population Q_0 after selection, recombination and mutation.
4: Set iteration times $t = 0$.
5: **while** $t < It_{\max}$ **do**
6: Generate a new population R_t with $2N_{pop}$ size ($R_t = P_t \cup Q_t$).
7: Apply non-dominated sorting on R_t to identify different non-dominated fronts of objective functions F_i.
8: Generate a new population P_{t+1} from F_i by crowding selection operator.
9: Create a new offspring population Q_{t+1} after selection, recombination and mutation.
10: $t = t + 1$.
11: **end while**

***Remark* 3 :** The main purpose of a learning task is to achieve both the satisfactory accuracy and good generalization performance. In the multi-objective function shown in Eq.(24), misclassification rate is to enhance classification accuracy while generalization gap is to guarantee the generalization performance. Misclassification rate is mainly used to demonstrate the difference between the actual value and the corresponding predicted output of the created model, which could describe classification or regression capability of the learning algorithm. Generalization gap is defined as a quantitative measurement for evaluating the identification capability of the model on the unseen data. In general, the created model will achieve different prediction results under different given data. If we obtain a relatively small training error, it indicates that the model can well represent the distribution characteristics of the training data. However, due to the fluctuation of data, the performance of the model on the unseen data may be declined. This

phenomenon shows that the trained model only learns the unique data pattern and not the universal one, and is defined as overfitting. It is the main barrier faced by a learning task. Therefore, misclassification rate and generalization gap are conflicting, i.e., one specific solution is best for one objective, but may be worst in the sense of the other objective.

4.4. Decision making via ensemble strategy

Through the above procedure, we can obtain a Pareto optimal solution set with N_{pop} solutions by multi-objective optimization. Actually, too many solutions may make it difficult for decision makers to make choice. Therefore, voting method is implemented to make a decision from the predicted results of all independent classifiers on the testing samples. Specifically, the class label vector is first expressed as $\{l_1, l_2, \ldots, l_k\}$. Then, for each testing sample x^{test}, the predicted results of N_{pop} independent classifiers can be obtained, defined as $\left\{\psi_i^1\left(x^{test}\right), \psi_i^2\left(x^{test}\right), \ldots, \psi_i^k\left(x^{test}\right)\right\}$, where $\psi_i^j\left(x^{test}\right)$ indicates that the predicted label of classifier $\psi_i\left(i = 1, \ldots, N_{pop}\right)$ is l_j. The final predicted label of x^{test} is determined by conducting a majority voting on all results obtained by these independent classifiers:

$$\Psi\left(x^{test}\right) = l_{\underset{j\in[1,\ldots,k]}{\arg\max} \sum_i^{N_{pop}} \psi_i^j(x^{test})} \tag{25}$$

4.5. MOAC-ELM algorithm description

Considering class proportion and data distribution information, MOAC-ELM introduces imbalance representation matrix $\mathbf{W}_m$ and penalty adjustment matrix $\mathbf{Q}_m$ for misclassification of each class to minimize the weighted cumulative training error, as shown in Eq.(11). $\mathbf{W}_m$ is represented by ratio of the number of the samples belonging to each class to total samples. In addition, duo to the difficulty of identification for data distribution information, a multi-objective optimization-based determination scheme with respect to penalty factors in penalty adjustment matrix is presented to adaptively discover the class-specific costs. In order to solve the optimization problem, NSGA-II is adopted to generate the Pareto optimal solutions. It should be noted that misclassification rate and generalization gap are expected to be minimized, which are defined by Eq.(20) and Eq.(23), respectively. The fitness values calculation can be achieved through the steps below: 1) construct the classifier using training dataset to estimate the output weights and compute the output function; 2) using the data from the validation dataset to perform the created classifier; 3) calculate the misclassification rate and generalization gap by Eq.(20) and Eq.(23). Finally, ensemble strategy is performed to make a decision from the obtained solutions to achieve the optimal output weights of MOAC-ELM. In summary, the description of MOAC-ELM is presented in Algorithm 2.

Algorithm 2 MOAC-ELM Algotirhm

Input: A training dataset $\mathbb{D} = \{(\mathbf{x}_i, \mathbf{y}_i) | \mathbf{x}_i \in \Re^n, \mathbf{y}_i \in \Re^m, i = 1, \ldots, N\}$, validation dataset $\mathbb{V}$, number of hidden nodes N_h, activation function $g(\cdot)$, initial parent population P_0, N_{pop} size of population, iteration times $t = 0$, and maximal number of iterations $It_{\max}$.

Training phase
Randomly initialize the population;
Estimate the output weight $\boldsymbol{\beta}$ by Eq.(14) and compute the output function by Eq.(15);
Compute the fitness of the initialized population by Eq.(20) and Eq.(23);
Determine the new population using NSGA-II algorithm;
Repeat the NSGA-II process until the goal is met or the preseted maximal number of iterations is reached;
Get the optimal penalty adjustment factors;
Testing phase
for any testing sample x^{test} **do**
    Set $i = 1$;
    while $i < N_{pop}$ **do**
        Using the ith independent classifier to predict the label of the testing sample x^{test};
        $i = i + 1$;
    end while
    Calculate the final predicted label of x^{test} by Eq.(25);
end for

5. Performance evaluation

In this section, comparisons are made with several state-of-the-art methods on both binary classification and multiclass classification tasks to evaluate the performance of MOAC-ELM. All experiments are performed in Matlab 2019a environment running on a computer equipped with an Intel Core i7-9750H CPU at 2.60 GHz.

5.1. Dataset description

Experiments are performed on selected benchmark datasets and a real-world application dataset to test the effectiveness of MOAC-ELM. For benchmark datasets, we use 10 binary classification datasets and 2 multiclass classification datasets, which have different CIRs. CIR is defined as

$$CIR = \begin{cases} \frac{\#(N^-)}{\#(N^+)}, & for\ k = 2 \\ \frac{\min(\#(y_i))}{\max(\#(y_i))}, & for\ k > 2 \end{cases} \tag{26}$$

where $\#(N^-)$ and $\#(N^+)$ represent the number of samples of the minority class and that of the majority class, $\#(y_i)$ is the sample number of class i, and k represents the number of classes.

Table 2 presents the detailed characterization of the benchmark datasets used in the experiments, including attributes number, class number, training data number, testing data number and CIR. According to Table 2, those datasets cover several different CIRs. The smaller its value, the higher the skew of the data. For binary classification, CIR ranges from 0.1 to 0.55. The CIRs of the two multiclass classification datasets are also different. In addition, in order to explore the performance of MOAC-ELM for handling the real-world imbalanced dataset, we also apply it on the blast furnace status diagnosis dataset. In this case, two datasets with different CIRs (i.e.,

BFSD1 and BFSD2) are formed from the original dataset, as shown in Table 3. Before the experiments, we normalize all the attributes to the range $[-1, 1]$ to remove the influence of dimension. The data normalization processing method is

$$\tilde{x} = \frac{(\tilde{x}_{\max} - \tilde{x}_{\min})(x - x_{\min})}{x_{\max} - x_{\min}} + \tilde{x}_{\min} \tag{27}$$

where $x_{\max}$ and $x_{\min}$ represent the maximum and minimum values of the variable before normalized, $\tilde{x}_{\max}$ and $\tilde{x}_{\min}$ are the maximum and minimum values of variable after normalized, x and $\tilde{x}$ are the variables values before and after normalized, respectively.

Table 2: Details of benchmark datasets

Datasets	#Attributes	#Category	#Training Data	#Testing Data	CIR
Abalone19	8	2	3339	835	0.01
Ecoli1	7	2	268	68	0.30
Ecoli2	7	2	268	68	0.18
Glass0	9	2	171	43	0.49
Glass1	9	2	171	43	0.55
Glass2	9	2	171	43	0.09
Pima	8	2	614	154	0.54
Yeast3	8	2	1187	297	0.12
DNA	180	3	2000	1186	0.44
Wine	13	3	118	60	0.68

Table 3: Details of blast furnace status diagnosis datasets

Code	#Attributes	#Category	CIR
BFSD1	12	2	0.2
BFSD2			0.5

5.2. Experimental settings

During the experiments, we divide all the datasets into training samples and testing samples. Moreover, two nonoverlapped subsets are extracted from the training samples, in which one is for training and the other for validation. For fair comparison, all the ELM-based algorithms adopt Sigmoid nodes $G(\mathbf{a}, b, \mathbf{x}) = 1/\left(1 + \exp\left(-\left(\mathbf{a} \cdot \mathbf{x} + b\right)\right)\right)$. Due to the utilization of random parameters between the input and hidden layers, grid search method is conducted to get the optimal combination of the number of hidden nodes N_h and the regularization factor C from the set $\{10, 20, \ldots, 1000\}$ and $\left\{2^{-24}, 2^{-23}, \ldots, 2^{25}\right\}$. For kernel-based ELM, we test the algorithm on Gaussian kernel $K(\mathbf{u}, \mathbf{v}) = \exp\left(-\sigma\|\mathbf{u} - \mathbf{v}\|^2\right)$. Therein two parameters to be adjusted are the regularization factor C and kernel width σ, and they are searched in the range of $\left\{2^{-24}, 2^{-23}, \ldots, 2^{25}\right\}$. In addition, for NSGA-II in MOAC-ELM, the population size is selected as 200, the crossover probability is set as 0.95, and the mutation probability is selected as 0.1. The range of decision variable is $[-1, 1]$. The different maximal iteration times are set for each dataset.

Fig. 5 illustrates the relationship between user specified parameters and classification performance. CIR is first considered to determine the parameters in MOAC-ELM. It can be seen that the performance in terms of accuracy fluctuates frequently when C and N_h in Sigmoid node change. With the variety of C and σ in Gaussian kernel, accuracy appears stable interval. On the Pima dataset, we see from Fig. 5(a1) that the distribution of validation accuracy for Sigmoid node has a relative smooth range with the best performance. In this case, when getting consistent boundary under circumstantiates of different parameter combinations, smaller N_h is preferred. Therefore, we select the best combination of (C, N_h) in this range with smaller N_h. As observed from Fig. 5(b1), the best performance of Gaussian kernel falls within a limited range. We prefer smaller σ than C when getting consistent boundary under circumstantiates of different parameter combinations. Similarly, all the parameters for each dataset are set based on the aforementioned observations. The parameter setting for W-ELM, kernel-based W-ELM, CCR-ELM, Boosting W-ELM also use the grid search manner.

After the model parameters are determined, we use NSGA-II to optimize the two defined objectives for the setting of penalty adjustment factors in MOAC-ELM. Fig. 6 presents the Pareto optimal solutions with 50, 100, 200, and 300 iterations on Pima dataset, Ecoli2 dataset, Yeast3 dataset and Glass1 dataset, respectively. In this figure, the vertical and horizontal axes show the first objective and the second objective, respectively. According to Fig. 6, the two objectives are mutually incompatible, which is the primary prerequisite for better convergence of the proposed multi-objective optimization model. It is obvious that the convergence results are not sensitive to the generations, but it also can be seen that as the maximal iterations increase, the solutions gradually become stable. Considering the computational complexity, the maximal iteration times is selected as 200 as a trade-off. Overall, the multi-objective optimization model can obtain the Pareto optimal solutions from the viewpoint of optimization results.

5.3. Performance evaluation on benchmark datasets

We divide all the datasets into three parts, including binary classification with high imbalance degree (CIR $\leq$ 0.2), binary classification with low imbalance degree (CIR $>$ 0.2) and multiclass classification. G-mean, accuracy and computation time are utilized as measurements of the performance of MOAC-ELM. Tables 4, 5, 6 and Fig. 7 illustrate the detailed comparison results. The best results achieved by each dataset appear as bold. According to Table 4 and Fig. 7(a), in terms of G-mean, as CIR increases, the results get worse, and the performance of ELM is pretty bad. MOAC-ELM achieves better results on Abalone19, Ecoli2 and Yeast3 datasets, especially when kernel trick is utilized. For Glass2 dataset, Boosting W-ELM achieves the best G-mean. In addition, the performance of MOAC-ELM and kernel-based MOAC-ELM are better than W-ELM, indicating that data distribution information has crucial influence on classification performance. As observed from Table 5 and Fig. 7(b), MOAC-ELM performs better than other comparison methods on Glass0 and Glass1 datasets. For Pima dataset, the performance of CCR-ELM and MOAC-ELM are with minor difference. The CIR of these datasets are not much different. Thus, it is likely that strategy differences between methods contribute to different results. The results presented here indicate that data complexity plays a more important role in the performance improvement than CIR. Furthermore, MOAC-ELM is also compared with non-ELM classifier, i.e., fuzzy SVM (FSVM) [42], and the comparison results are given. The accuracy of FSVM on Ecoli1, Glass0, Glass1 and Pima datasets are 88.12, 83.83, 64.52, 70.24, respectively. It can be observed that MOAC-ELM is superior to FSVM.

The comparison results of two multiclass classification datasets are listed in Table 6. It can be observed that kernel-based MOAC-ELM presents more accurate classification results. It is

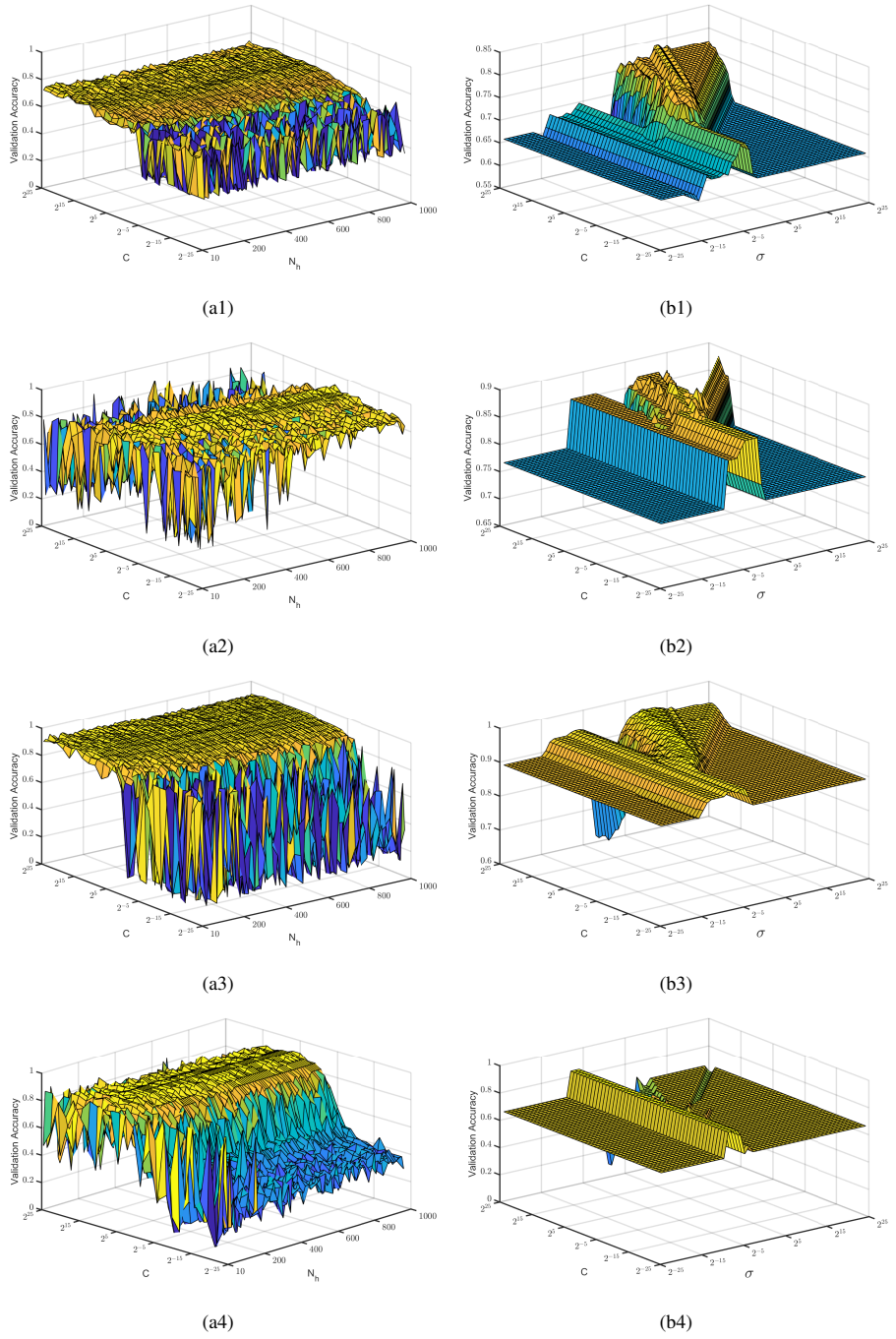
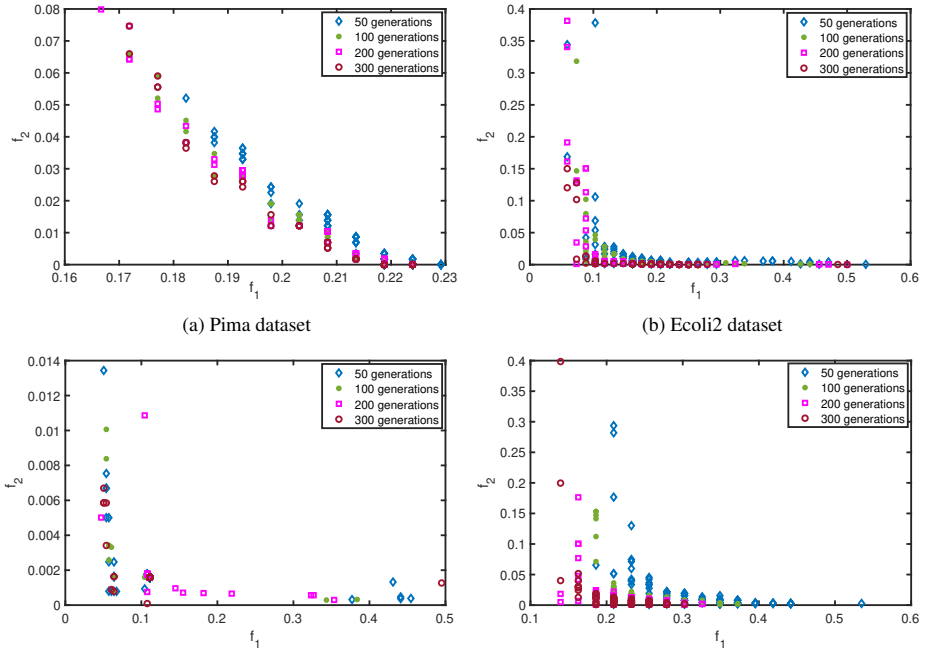


Figure 5: Validation accuracy distribution with varying user specified parameter (C, N_h) for Sigmoid node or (C, σ) for Gaussian kernel on four different datasets: (a1), (a2), (a3) and (a4) are the different choices of C and L for Sigmoid node on Pima dataset, Ecoli1 dataset, Yeast3 dataset and BFSD2 dataset; (b1), (b2), (b3) and (b4) are the different choices of C and σ for Gaussian kernel on Pima dataset, Ecoli1 dataset, Yeast3 dataset, and BFSD2 dataset, respectively.



(a) Pima dataset

(b) Ecoli2 dataset

(c) Yeast3 dataset

(d) Glass1 dataset

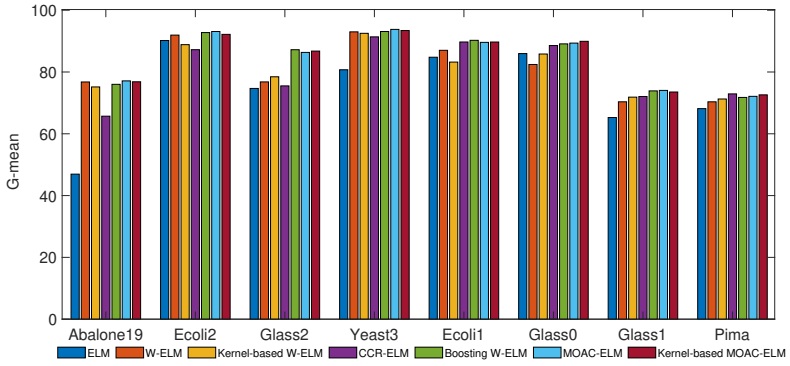
Figure 6: Pareto optimal solutions obtained by 50, 100, 200, and 300 iterations on different datasets.

Table 4: Experimental results of binary classification with high imbalance degree (CIR ≤ 0.2)

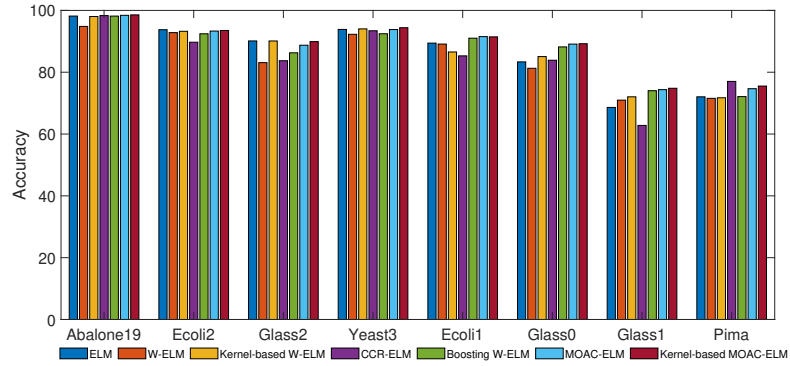
Approaches	Abalone19 (0.0078)		Ecoli2 (0.1831)		Glass2 (0.0947)		Yeast3 (0.1234)	
	G-mean	Accuracy	G-mean	Accuracy	G-mean	Accuracy	G-mean	Accuracy
ELM	46.94	98.17	90.19	**93.75**	74.68	**90.12**	80.72	93.85
W-ELM	76.77	94.82	91.94	92.81	76.79	83.10	92.99	92.27
kernel-based W-ELM	75.16	98.02	88.85	93.24	78.45	90.11	92.52	94.02
CCR-ELM	65.70	98.33	87.22	89.71	75.50	83.72	91.37	93.41
Boosting W-ELM	76.01	98.16	92.76	92.44	**87.21**	86.28	93.11	92.45
MOAC-ELM	**77.12**	98.41	**93.11**	93.32	86.34	88.73	**93.79**	93.81
kernel-based MOAC-ELM	76.83	**98.55**	92.18	93.50	86.77	89.90	93.42	**94.40**

Table 5: Experimental results of binary classification with low imbalance degree (CIR > 0.2)

Approaches	Ecoli1 (0.2973)		Glass0 (0.4861)		Glass1 (0.5507)		Pima (0.5350)	
	G-mean	Accuracy	G-mean	Accuracy	G-mean	Accuracy	G-mean	Accuracy
ELM	84.78	89.42	85.96	83.34	65.26	68.61	68.14	72.04
W-ELM	87.02	89.11	82.43	81.27	70.35	70.97	70.35	71.55
kernel-based W-ELM	83.21	86.55	85.82	85.06	71.86	72.04	71.26	71.73
CCR-ELM	89.71	85.29	88.56	83.88	72.07	62.79	**72.91**	**77.03**
Boosting W-ELM	**90.25**	91.01	89.10	88.19	73.88	74.02	71.77	72.11
MOAC-ELM	89.59	**91.52**	89.35	89.10	**74.04**	74.36	72.12	74.67
kernel-based MOAC-ELM	89.70	91.44	**89.93**	**89.22**	73.52	**74.81**	72.60	75.52



(a)



(b)

Figure 7: Comparison results of different datasets: (a) G-mean and (b) Accuracy.

Table 6: Experimental results of multiclass classification

Approaches	DNA		Wine	
	G-mean	Accuracy	G-mean	Accuracy
ELM	93.78	93.84	95.64	97.83
W-ELM	93.54	93.44	97.58	97.67
kernel-based W-ELM	94.24	94.05	97.75	97.88
CCR-ELM	94.88	93.16	96.10	98.33
Boosting W-ELM	**94.91**	94.65	97.80	98.69
MOAC-ELM	94.79	94.72	98.15	99.02
kernel-based MOAC-ELM	94.80	**94.85**	**98.22**	**99.21**

Table 7: Computation time of different methods on 4 selected datasets

Datasets	Ecoli2	Pima	Glass1	Yeast3
ELM	**0.0156**	**0.0128**	0.1404	**0.0312**
W-ELM	0.0374	0.1145	0.0425	0.6510
kernel-based W-ELM	0.0343	0.1045	**0.0274**	0.4619
CCR-ELM	0.0410	0.1288	0.0653	0.6273
Boosting W-ELM	0.2722	0.2751	0.1425	0.9771
MOAC-ELM	182.29	400.45	355.59	640.32
kernel-based MOAC-ELM	185.44	398.38	356.03	621.35

easier to explain this appearance: kernel trick can map the raw data to the higher dimensional space. Accordingly, MOAC-ELM is also competent for multiclass classification tasks.

Table 7 shows the average training time on 4 selected datasets. The comparison results show that ELM, W-ELM, and kernel-based W-ELM take less training time than MOAC-ELM. Boosting W-ELM needs several iteration steps, making its training speed slower than ELM and W-ELM. Comparing with these methods, MOAC-ELM is relatively time-consuming due to the time spending in training multi-objective optimization model, but it still can meet requirements of the classification tasks.

5.4. Performance evaluation on real-world application

In this section, the performance of MOAC-ELM in dealing with real-world application is tested using blast furnace diagnosis dataset. In addition, the effectiveness and feasibility of MOAC-ELM is verified through comparative experiments.

Blast furnace is the smelting equipment used to produce hot metal in metallurgical industry, as shown in Fig. 8 [43, 44]. Safety and reliability is the primary concern of ironmaking production. There is no doubt that it is essential to monitor the production status of blast furnace accurately. In actual production, there are often more samples of normal operating than samples of abnormal operating, thereby status diagnosis of the blast furnace can be viewed as a imbalanced classification problem.

To be specific, majority class is composed of the normal samples while minority class is represented by abnormal samples, thus status diagnosis of the blast furnace is designed as a bi-

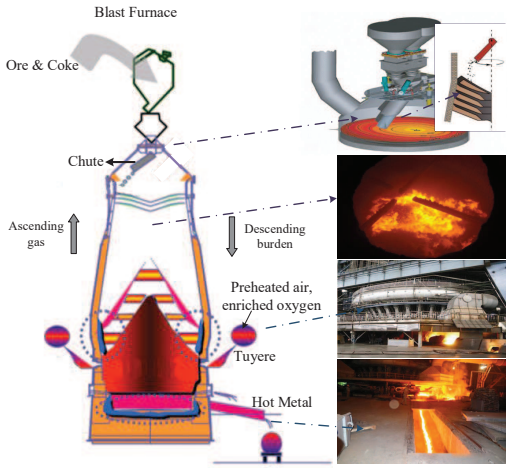


Figure 8: Blast furnace system overview

Table 8: Characteristics of two blast furnace status diagnosis datasets with different CIRs

No.	CIR	Training samples	Testing samples
BFSD1	0.2	1200	350
BFSD2	0.5	900	200

nary classification problem. The data were collected from a medium-size blast furnace. Though the analysis and experts' experience, the related variables are treated as characteristic feature to determine whether the status is normal or not, including blast pressure, top pressure, blast temperature, top temperature (including four-point temperature), blast volume, differential pressure, permeability index, cross temperature (including center and edge temperature). The simulation is carried out based on a dataset containing 1800 samples, including 1400 normal samples and 400 abnormal samples. In order to verify the validity of MOAC-ELM, datasets with two different CIRs are formed by randomly selecting normal and abnormal samples from the original dataset, as shown in Table 3. Table 8 details the characteristics of the two datasets with different CIRs .

The Pareto optimal solutions obtained after different generations on BFSD1 dataset are presented in Fig. 9. As can be observed, with the generations increase, the convergence results do not change much. Thus, we set the maximal iteration times equal to 100 in this experiment.

Table 9 illustrates the detailed results on G-mean and accuracy among different methods. The results are also illustrated in Fig. 10 to make the comparison easier to recognize. According to Table 9 and Fig. 10, both MOAC-ELM and kernel-based MOAC-ELM outperform the compared methods. For BFSD1, from the point of view of G-mean, kernel-based MOAC-ELM is 16.3% higher than ELM, 8.6% higher than W-ELM, 7.0% higher than kernel-based W-ELM, 6.7% higher than CCR-ELM, and 5.9% higher than boosting W-ELM. Similarly, for BFSD2, kernel-based MOAC-ELM is 7.7% higher than ELM, 6.4% higher than W-ELM, 3.8% higher than kernel-based W-ELM, 5.7% higher than CCR-ELM, and 3.6% higher than Boosting W-ELM. From the above analysis, we can see that MOAC-ELM has better classification performance under lower

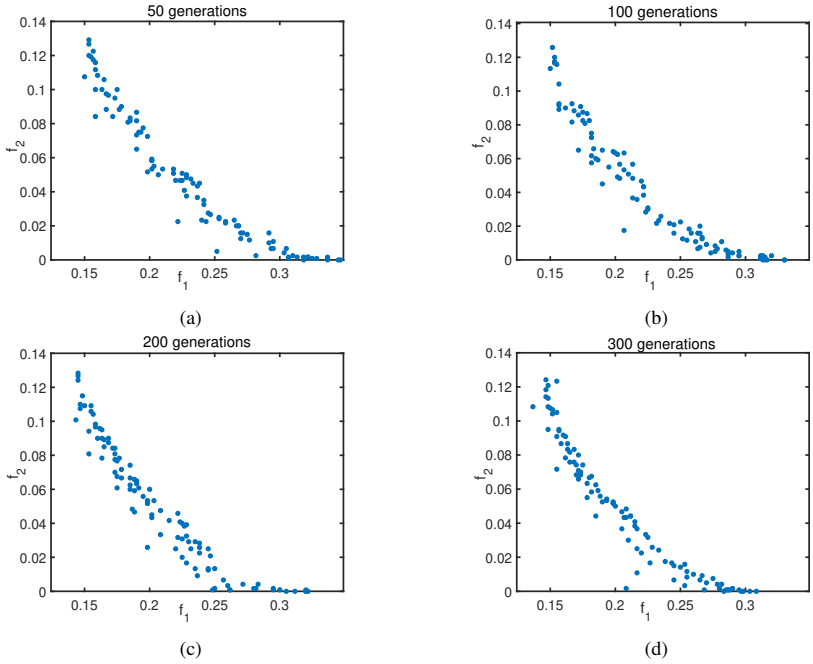


Figure 9: Pareto optimal solutions obtained by 50, 100, 200, and 300 generations on BFSD1.

CIR. Accordingly, MOAC-ELM is more suitable for BF status diagnosis implementation under imbalanced data characteristics. In addition, the training time of ELM, W-ELM and MOAC-ELM are 0.0845s, 0.7338s, and 810.2874s, respectively. It is obvious that MOAC-ELM takes more time than other methods because multi-objective optimization is involved. We also report the parameter sensitivity analysis and give the experimental results of different population sizes. For MOAC-ELM, the G-mean with 100, 150, 200 population sizes on BFSD1 dataset are 84.18, 85.41, 85.71, respectively. The statistical standard deviation is 0.8107. The results are similar when the population sizes are 150 and 200. It further indicates that MOAC-ELM is not sensitive to the population size.

6. Discussions

In this section, we discuss the differences between MOAC-ELM and related optimization-based ELMs, and other cost-sensitive processing manner.

6.1. Differences with respect to related optimization-based ELMs

The multi-objective evolutionary algorithm is regarded as the most popular and effective algorithm for solving multi-objective optimization problem, which has been used to enhance the performance of machine learning-based model and the quality of their results [45]. As the randomly assigned parameters degrade the performance, many researchers tried to optimize the network parameters to achieve more compact network structure by using several optimization algorithms [46, 47]. Meanwhile, heuristic feature extraction methods were developed [48, 49,

Table 9: Experimental results of blast furnace status diagnosis

Approaches	BFSD1		BFSD2	
	G-mean	Accuracy	G-mean	Accuracy
ELM	73.85	84.87	84.63	86.84
W-ELM	79.12	86.42	85.62	86.01
kernel-based W-ELM	80.32	87.69	87.79	88.90
CCR-ELM	80.55	88.07	86.21	85.94
Boosting W-ELM	81.12	88.20	87.95	87.54
MOAC-ELM	85.71	**90.33**	89.31	88.92
kernel-based MOAC-ELM	**85.92**	90.17	**91.12**	**90.01**

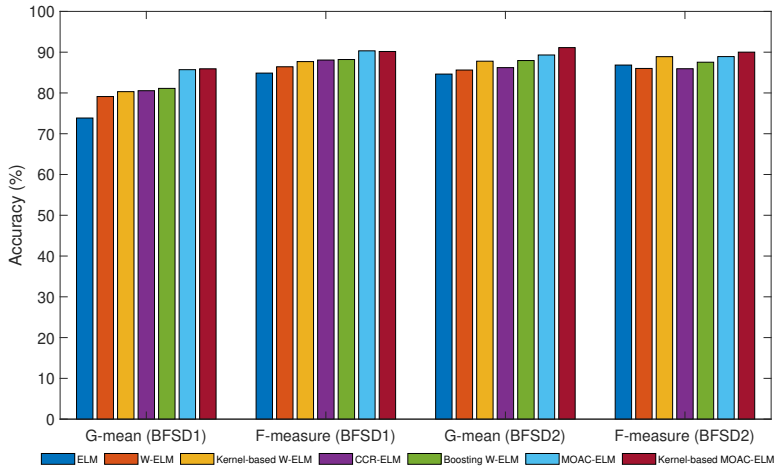


Figure 10: Bar chart for blast furnace status diagnosis

50]. However, such kind of works only consider RMSE as the performance criterion, which only can evaluate the error between the actual value and the corresponding predicted value of the created model. In addition, a multi-objective genetic programming approach was developed to evolving accurate and diverse ensembles of genetic program classifiers on imbalanced data [51]. The major differences of MOAC-ELM with respect to related optimization-based ELMs are that 1) A multi-objective optimization model is investigated to achieve a flexible and efficient weights determination scheme under the framework of W-ELM for imbalanced classification problem; 2) Generalization gap is defined as one of the optimization objectives, which represents the ability of the algorithm to generalize well from the learning data to the unseen data, for producing insightful and effective generalization bounds. Then, the misclassification rate and generalization gap are jointly taken into consideration as the two conflicting objectives to ensure the classification accuracy and alleviate the overfitting issue. It should be pointed out that the training process of MOAC-ELM is relatively time-consuming compared to classic ELM and W-ELM due to the multi-objective optimization involved, but it still can meet the requirements of the classification tasks.

6.2. Differences with respect to other cost-sensitive processing manner

Cost-sensitive learning is the most common strategy to deal with imbalanced data, which takes the misclassication costs into consideration and gives higher misclassication costs to minority class samples than majority class samples. For many cases, the prior information of cost distribution is insufficient and the relationship between cost matrix and classification performance is unclear. CCR-ELM [23] and CSKELM [24] use class-specific regularization parameters whose value are fixed by performing a grid search and using the proportion of class samples, respectively. In order to derive cost matrix in an adaptive manner, Boosting method is introduced to obtain weighting schemes in W-ELM though several iteration steps in [22]. Li et al. [26] established an upper error bound model for W-ELM to provide some useful guidelines for assigning case-weighting factors. Compared with such methods, MOAC-ELM embeds an imbalance representation matrix and a penalty adjustment matrix into ELM by fully considering class proportion and data complexity. In addition, penalty adjustment factors are regarded as the solutions of a multi-objective optimization problem to capture data complexity without the need of prior knowledge and many trials. From the experimental results, MOAC-ELM achieves better performance than W-ELM in both benchmark datasets and a real-world application dataset, revealing that using penalty adjustment matrix to describe data distribution information has a great impact on classification performance.

7. Conclusions

In this paper, we focus on the problem of robust and adaptive weighting determination scheme in class imbalance field, and present a solution of MOAC-ELM. Specifically, an imbalance representation matrix and a penalty adjustment matrix are embedded into ELM to make the decision boundary locate in the ideal position. In order to learn the penalty adjustment factors in an adaptive manner for the case-specific situations, we establish a multi-objective optimization model by considering comprehensive optimization criteria. In addition, NSGA-II is adopted to optimize the objectives to get the compromise solutions and make a decision by ensemble strategy. The proposed MOAC-ELM can determine the costs adaptively for different tasks, enabling to enhance the representation of data distribution, so as to overcome the limitations of the

fixed weight mechanism in the existing methods. As such, W-ELM with uniform weights can be regards as a special case of MOAC-ELM. Comprehensive comparisons with several state-of-the-art methods have been performed on selected benchmark datasets and a real-world application dataset. The experimental results demonstrate the good robustness of MOAC-ELM. Multi-label imbalanced learning in the ELM community is still challenging. How to deal with multi-label imbalanced classification task using this strategy is a future research-worthy problem.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 62003038, and in part by National Key Research and Development Program of China under Grant 2017YFB1401203.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. Wang, L. Minku, X. Yao. A systematic study of online class imbalance learning with concept drift. IEEE trans Neural Netw Learn Syst 29(10) (2018) 4802-4821.
[2] S. Yu, Z. Abraham, H. Wang, M. Shah, Y. Wei, J.C. Principe. Concept drift detection and adaptation with hierarchical hypothesis testing. J Franklin Inst 356(5) (2019) 3187-3215.
[3] J. Sun, H. Li, H. Fujita, B. Fu, W. Ai. Class-imbalanced dynamic financial distress prediction based on Adaboost-SVM ensemble combined with SMOTE and time weighting. Inform Fusion 54 (2020) 128-144.
[4] D. Wu, Z. Wang, Y. Chen, H. Zhao. Mixed-kernel based on weighted extreme learning machine for inertial sensor based human activity recognition with imbalance dataset. Neutocomputing 190 (2016) 35-49.
[5] D. Wang D, X. Zhang, H. Chen, Y. Zhou, F. Cheng. Sintering conditions recognition of rotary kiln based on kernel modification considering class imbalance. ISA Trans 106 (2020) 271-282.
[6] H. He, E. Garcia. Learning from Imbalanced data. IEEE Trans Knowl Data Eng 21(9) (2009) 1263-1283.
[7] H. Guo, Y. Li, J. Shang, M. Gu, Y. Huang, B. Gong. Learning from class-imbalanced data: Review of methods and applications. Expert Syst Appl 73 (2017) 220-239.
[8] S. Wang, X. Yao. Relationships between diversity of classification ensembles and single-class performance measures. IEEE Trans Knowl Data Eng 25(1) (2013) 206-219.
[9] J. Obregon, A. Kim, J. Jung. RuleCOSI: Combination and simplification of production rules form boosted decision trees for imbalanced classification. Expert Syst Appl 126 (2019) 64-82.
[10] Y Li, H. Chi, X. Shao, M. Qi, B. Xu. A novel random forest approach for imbalance problem in crime linkage. Knowl-Based Syst 195 (2020) 105738.
[11] S.S. Mullick, S. Datta, S. Das. Adaptive learning-based k-nearest neighbor classifier with resilience to class imbalance. IEEE Trans Neural Netw Learn Syst 29(11) (2018) 5713-5725.
[12] G. Chen, Y. Liu, Z. Ge. K-means Bayes algorithm for imbalanced fault classification and big data application. J Process Contr 81 (2019) 54-64.
[13] Y. Wang, L. Yang. A robust loss function for classification with imbalanced datasets. Neurocomputing 331 (2019) 40-49.
[14] G.B. Huang, Q.Y. Zhu, C.K. Siew. Extreme learning machine: Theory and applications. Neurocomputing 70(1-3) (2006) 489-501.
[15] J. Zhang, W. Xiao, Y. Li, S. Zhang, Z. Zhang. Multilayer probability extreme learning machine for device-free localization. Neurocomputing 396 (2020) 383-393.
[16] J. Cao, J. Hao, X. Lai,C.M. Vong, M. Luo. Ensemble extreme learning machine and sparse representation classification. J Franklin Inst 353(17) (2016) 4526-4541.
[17] J. Zhang, W. Xiao, Y. Li, S. Zhang. Residual compensation extreme learning machine for regression. Neurocomputing 311 (2018) 126-136.

[18] J. Zhang, Y. Li, W. Xiao, Z. Zhang. Robust extreme learning machine for modeling with unknown noise. J Franklin Inst 357(14) (2020) 9885-9908.
[19] K.A. Toh. Deterministic neual classification. Neural Computat 20(6) (2008) 1565-1595.
[20] W. Zong, G.B. Huang, Y. Chen. Weighted extreme learning machine for imbalance learning. Neurocomputing 101 (2013) 229-242.
[21] W. Zhang, H. Ji. Fuzzy extreme leanrning machine for classification. Electron Lett 49(7) (2013) 448-450.
[22] K. Li, X. Kong, Z. Lu, W. Liu, J. Yin. Boosting weighted ELM for imbalanced learning. Neurocomputing 128 (2014) 15-21.
[23] W. Xiao, J. Zhang, Y. Li, S. Zhang, W. Yang. Class-specific cost regulation extreme learning machine for imbalanced classification. Neurocomputing 261 (2017) 70-82.
[24] B.S. Rahuwanshi, S. Shukla. Class-specific kernelized extreme learning machine for binary class imbalance learning. Appl Soft Comput J 73 (2018) 1026-1038.
[25] H. Yu, C. Sun, X. Yang, S. Zheng, Q. Wang, X. Xi. LW-ELM: A fast and flexible cost-sensitive learning framework for classifying imbalanced data. IEEE Access 6 (2018) 28488-28500.
[26] H. Li, X. Yang, Y. Li, L. Hao, T. Zhang. Evolutionary extreme learning machine with sparse cost matrix for imbalanced learning. ISA Trans 100 (2020) 198-209.
[27] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan. A fast and elitist multiobjective genetic algorithm NSGA-II. IEEE Trans Evolut Comput 6(2) (2002) 182-197.
[28] X. Liu, J. Wu, Z. Zhou. Exploratory undersampling for class-imbalance learning. IEEE Trans Cybern 39(2) (2009) 539-550.
[29] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. J Artif Intell Res 16 (2002) 321C357.
[30] A. Sen, M.M. Islam, K. Murase, X. Yao. Binarization with boosting and oversampling for multiclass classification. IEEE Trans Cybern 46(5) (2016) 1078-1091.
[31] V. Lopez, A. Fernandez, S. Garcia, V. Palade, F. Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Informa Sciences 250 (2013) 113-141.
[32] V. Lopez, A. Fernandez,J.G. Moreno-Torres, F. Herrera. Analysis of preprocessing vs cost-sensitive learning for imbalanced classification, open problems on intrinsic data characteristics. Expert Syst Appl 39(7) (2012) 6585C6608.
[33] Y. Sun, K. Tang, L.L. Minku, S. Wang, X. Yao. Online ensemble strategy of data streams with gradually evolved classes. IEEE Trans Knowl Data Eng 28(6) (2016) 1532C1545.
[34] Y. Li, S. Zhang, Y. Yin, W. Xiao, J. Zhang. Parallel one-class extreme learning machine for imbalance learning based on Bayesian approach. J Amb Intell Hum Comp (2018) https://doi.org/10.1007/s12652-018-0994-x.
[35] G.B. Huang, H. Zhou, X. Ding, R. Zhang. Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern Part B: Cybern 42 (2) (2012) 513-529.
[36] Y. Li, S. Zhang, J. Zhang, Y. Yin, W. Xiao, Z. Zhang. Data-driven multi-objective optimization for burden surface in blast furnace with feedback compensation, IEEE Trans Ind Informat 16(4) (2020) 2233-2244.
[37] J. Zhang, Y. Li, W. Xiao, Z. Zhang. Non-iterative and fast deep learning: multilayer extreme learning machines. J Franklin Inst 357(13) (2020) 8925-8955.
[38] J. Zhang, Y. Li, W. Xiao. Integrated multiple kernel learning for device-free localization in cluttered environments using spatiotemporal information. IEEE Internet Things J 8(6) (2021) 4749-4761.
[39] H. Yu, X. Yang, S. Zheng, C. Sun. Active learning from imbalanced data: A solution of online weighted extreme learning machine. IEEE Trans Neural Netw Learn Syst 30(4) (2019) 1088-1103.
[40] R. Fletcher. Practical methods of optimization: Volume 2 Constrained optimization, Wiley, NewYork, 1981.
[41] K. Kawaguchi, L.P. Kaelbling, Y. Bengio. Generalization in deep learning. arXiv:1710.05468, 2020.
[42] R. Batuwita, V. Palade. FSVM-CIL: Fuzzy support vector machines for class imbalance learning. IEEE Trans Fuzzy Syst 18(3) (2010) 558C571.
[43] Y. Li, S. Zhang, Y. Yin, J. Zhang, W. Xiao. A soft sensing scheme of gas utilization ratio prediction for blast furnace via improved extreme learning machine. Neural Process Lett 50(2) (2019) 1191-1213.
[44] Y. Li, H. Li, J. Zhang, S. Zhang, Y. Yin. Burden surface decision using MODE with TOPSIS in blast furnace ironmaking. IEEE Access 8 (2020) 35712-35725.
[45] A. Telikani, A. Tahmassebi, W. Banzhaf, A. Gandomi. Evolution machine learning: A survey. ACM Comput Surv 54 (8) (2021) 161:1-161:35.
[46] Y. Wu, Y. Zhang, X. Liu, Z. Cai, Y. Cai. A multiobjective optimization-based sparse extreme learning machine algorithm. Neurocomputing 317 (2018) 88-100.
[47] M. Eshtay, H. Faris, N. Obeid. Improving extreme learning machine by competitive swarm optimization and its application for medical diagnosis problems. Expert Syst Appl 104 (2018) 134-152.
[48] M Luo, C. Li, X. Zhang, R. Li, X. An. Compound feature selection and parameter optimization of ELM for fault diagnosis of rolling element bearbings. ISA Trans 65 (2016) 556-566.
[49] Y. Xue, T. Tang, W. Pang, A.X. Liu. Self-adaptive parameter and strategy based particle swarm optimization for

large-scale feature selection problem with multiple classifiers. Appl Soft Comput 88 (2020) 106031-106042.
[50] Y. Bi, B. Xue, M. Zhang. Multi-objective genetic programming for feature learning in face recognition. Appl Soft Comput 103 (2021) 107152-107165.
[51] U. Bhowan, M. Johnston, M. Zhang, X. Yao. Evolving diverse ensembles using genetic programming for classification with unbalanced data. IEEE Trans Evol Comput 17 (3) (2013) 368-386.