



Artificial bee colony optimization-based weighted extreme learning machine for imbalanced data learning

Xiaofen Tang^{1,2} · Li Chen¹

Received: 5 November 2017 / Accepted: 10 January 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

The imbalanced datasets are common in real-world application and the problem of imbalanced dataset affect classification performance of many standard learning approaches. To address imbalanced datasets, a weighted extreme learning machine (WELM) solving the L_2 -regularized weighted least squares problem is presented to avoid the generation of an over-fitting model and obtain better generalization ability compared with ELM. However, the weight generated according to class distribution of training data leads to lack of finding optimal weight with good generalization performance and the randomness of input weight and hidden biases of network makes the algorithm produce suboptimal classification model. In this paper, a weighted extreme learning machine based on hybrid artificial bee colony (HABC) is proposed to obtain better performance than WELM, in which input weights and hidden bias of WELM and the weight assigned to training samples are optimized by the hybrid artificial bee colony algorithm. HABC combines the diversities of the perturbed parameter vectors of differential evolution with the best solution information of the artificial bee colony effectively. In the empirical study, different class imbalance data handling methods including four WELM-based methods, weighted support vector machine, four ensemble methods which combine data sampling and the Bagging or Boosting are compared with our method. The experimental results on 15 imbalanced datasets show that the proposed method outperforms most methods, which indicates its superiority.

Keywords Artificial bee colony algorithm · Weighted extreme learning machine · Imbalanced data learning · Single hidden layer feed-forward networks

1 Introduction

Extreme learning machine (ELM) [1,2] is a single layer feed-forward neural network (SLFN) with important features such as elimination of the needed parameter tuning during the training phase, united solutions for regression, binary, and multi-class classification [3,4]. The overall efficiency of ELM has been proved in many areas like specific proteins detection in bioinformatics [5–8], drug discovery problem [9], air pollutant detection [10], medical diagnosis [11,12], corporate life cycle prediction [13], face recognition [14,15], abnormal activity recognition [16], and so on. Although the ELM can classify balanced datasets effectively, it causes bias

towards the majority classes and results in a lower sensitivity in detecting the minority class when dealing with imbalance class [17].

The class imbalance problem are common in real-word domains, such as detecting oil spills from satellite images, anomaly detection, image annotation, software defect prediction, spam filtering [10,16,17]. To deal with imbalance class, fuzzy extreme learning machine (FELM) and weighted extreme learning machine (WELM) have been proposed. According to the results of kernel-based possibilistic fuzzy c-Means clustering, FELM assigns different misclassification cost to the positive samples and negative samples for imbalanced datasets with noise and outliers [18]. WELM assigns different weight values for the training samples of different classes to reflect their importance in respective classes and solves the L^2 -regularized weighted least squares problem to obtain better generalization performance compared with ELM [19]. Similar to traditional ELM, WELM obtains its extremely fast learning speed due to random assignment of input weights and hidden neuron biases. However, this

✉ Xiaofen Tang
xiaofengtang@hotmail.com

¹ School of Information Science and Technology, Northwest University, Xi'an 710127, China

² School of Information Engineering, Ningxia University, Yinchuan 750021, China

also causes the WELM to produce suboptimal classification model [20,21]. The effect of random parameters in ELM imposed on the generalization performance can be quite significant, particularly for imbalanced data. In addition, the weight generated according to class distribution of training samples depends on input data, which leads to the lack of finding optimal weight with good generalization performance [22–25].

In order to reduce the effect of random parameters of ELM, evolutionary extreme learning machine (DE-ELM) has been proposed. DE-ELM adopt the differential evolutionary algorithm (DE) to select the input weights and ELM to calculate the output weight. Experimental results show that the proposed algorithm is able to achieve good generalization performance with much more compact networks [26]. However, this approach developed to reducing the effect of random parameters of the ELM is accuracy-oriented, this causes bias towards the majority classes and results in a lower sensitivity in detecting the minority class, whereas classifying the minority class accurately can be more important than classifying the majority ones accurately in some scenarios as the minority ones often represents the main class of interest. In addition, although the existing external approaches proposed to deal with imbalanced data classification [27–33] can diminish the effect caused by class imbalance by preprocessing the datasets, when it comes to some real applications such as in bioinformatics and medical diagnosis, it is significant that the predictor is evaluated on the distribution of natural samples instead of balanced distribution under artificial construction, since the external approaches artificially modify the background frequencies of different classes [34]. Therefore, it is meaningful to discuss the internal approaches of imbalance learning that can be applied for the WELM model [35].

Artificial bee colony (ABC) has been used increasingly for global optimization problems [36–38]. Compared with existing state-of-the-art population-based approaches, ABC has simple evolutionary operators and fewer parameters need to adjust. Moreover, ABC algorithm is applied to find optimal weight set of the neural network through minimizing output error In the literature [39]. However, Individual position is modified by moving towards a randomly selected solution, which is no guaranty that new candidate position will be better than the last one. So the solution search of ABC algorithm is good at exploration but poor at exploitation [37,38]. It is observed that DE is good at exploring the search space and locating the region of global minimum but it is slow at exploitation of the solution [40]. Combining ABC with DE algorithm can improve the performance of basic ABC. Therefore, the hybrids of ABC and DE should be promising for training feed-forward neural networks [41,42].

In this paper, an imbalance learning algorithm applied for the ELM model, named hybrid artificial bee colony

optimization-based weighted extreme learning machine (HABC-WELM) is proposed, in which, input weights and hidden bias of network and the weight assigned to training samples are optimized by the hybrid artificial bee colony algorithm, whose exploitation and convergence rate is improved by combining DE operator with the ABC operation, and the network output weights are calculated using WELM. The hybrid ABC mainly focused on restricting the input weights and hidden biases to a reasonable range and assigning optimal weight to training data to achieve good generalization performance. In order to make results comparable, we implement five internal approaches WELM, ensemble weighted extreme learning machine (EN-WELM), evolutionary weighted extreme learning machine (DE-WELM), artificial bee colony optimization-based weighted extreme learning machine (ABC-WELM) and weighted support vector machine (WSVM) which is similar to WELM based on cost-sensitive learning [43]. In addition, we also implement four popular external approaches based ensemble methods RUSBoost, EsayEnsemble, OverBagging, UnderBagging and consider SVM as base classifier since it has been the powerful tools for supervised learning and widely used in classification in a variety of real-world problems. The classification performance and stabilization of the above nine algorithms is also analyzed. In this paper, we focus on imbalanced binary classification datasets, referring to the minority class with positive label and the majority class with negative label.

The rest of paper is organized as follows. In Sect. 2 we review the weighted extreme learning machine and the artificial bee colony algorithm. The details of the proposed HABC-WELM algorithm are described in Sect. 3. In Sect. 4, we carry out the comparison of our method with the state-of-the-art methods. The conclusion and future works are provided in Sect. 5.

2 Preliminaries

This section provides brief reviews of weighted ELM and artificial bee colony algorithm.

2.1 Weighted extreme learning machine

According to statistical learning theory [44], Deng et al proposed a regularized extreme learning machine (RELM) based on L_2 penalty which incorporates the structural risk minimization theory and the weighted least squares method into the ELM [45].

Given N distinct samples (x_i, t_i) , where $x_i = [x_{i,1}, x_{i,2}, \dots, x_{i,m}]^T \in R^m$ and $t_i = [t_{i,1}, t_{i,2}, \dots, t_{i,c}]^T \in R^c$. $i = 1, 2, \dots, N$. $j = 1, 2, \dots, L$. m represents the number of

feature. c represents the number of class. The mathematical model of RELM with L hidden nodes and activation function $g(x)$ can be expressed as

$$\text{Minimize : } \frac{1}{2} \|\beta\|^2 + \frac{\gamma}{2} \sum_{i=1}^N \|\varepsilon_i\|^2 \quad (1)$$

$$\text{Subject to : } \beta_j g(a_j x_i + b_j) - t_i = \varepsilon_i$$

where $a_j = [a_{j,1}, a_{j,2}, \dots, a_{j,m}]$ is the weight vector between the j -th hidden node and the input nodes, $\beta_j = [\beta_{j,1}, \beta_{j,2}, \dots, \beta_{j,c}]^T$ is the weight vector between the j -th hidden node and the output nodes, b_j is threshold of the j -th hidden nodes, $\varepsilon_i = [\varepsilon_{i,1}, \dots, \varepsilon_{i,m}]$ is the training error vector of the m output nodes with respect to the training sample x_i , γ represents the trade-off between the minimization of training errors and the maximization of the marginal distance, The parameters (a_j, b_j) in the hidden layer nodes function $h(x_i) = g(a_j, b_j, x_i)$ are randomly generated according to any continuous probability distribution.

The regularized extreme learning machine avoids the generation of an over-fitting model, provides more robust estimates and owns better generalization ability than ELM. But it can leave ELM algorithms with a performance bias, when using a dataset with imbalanced class distribution in the learning process. In order to alleviate the bias in performance caused by imbalanced class distribution, an extra weight $W_{ii} = \frac{1}{\#(t_i)}$ or $W_{ii} = \begin{cases} \frac{0.618}{\#(t_i)}, & \text{if } t_i > \text{AVG}(t_i) \\ \frac{1}{\#(t_i)}, & \text{if } t_i \leq \text{AVG}(t_i) \end{cases}$ is assigned to each sample to strengthen the impact of minority class while weaken the relative impact of majority class in literature [19], where $\#(t_i)$ is the number of samples belonging to class c , $c \in 1, 2$, So the weighted extreme learning machine(WELM) is mathematically modeled as

$$\text{Minimize : } \frac{1}{2} \|\beta\|^2 + \frac{\gamma}{2} W \sum_{i=1}^N \|\varepsilon_i\|^2 \quad (2)$$

$$\text{Subject to : } \beta_j g(a_j x_i + b_j) - t_i = \varepsilon_i$$

Substituting the constraints to the objective function yields, the Lagrangian for (2) can be written as follows

$$L(\beta, \varepsilon, \alpha) = \frac{1}{2} \|\beta\|^2 + \frac{\gamma}{2} W \sum_{i=1}^N \varepsilon_i^2 - \sum_{i=1}^N \alpha_i \left(\sum_{j=1}^L \beta_j g(a_j x_i + b_j) - t_i - \varepsilon_i \right) \quad (3)$$

Where $\alpha_i \in R(i = 1, 2, \dots, N)$ is the Lagrangian multiplier with the equality constraints of (2). Furthermore, by making the partial derivatives with respect to variables $(\beta, \varepsilon, \alpha)$ all

equal to zero, we can have the KKT corresponding optimality conditions as follows:

$$\frac{\partial L}{\partial \beta} = 0 \rightarrow \beta = H^T \alpha \quad (4.1)$$

$$\frac{\partial L}{\partial \varepsilon_i} = 0 \rightarrow \alpha_i = \gamma W \varepsilon_i \quad (4.2)$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \rightarrow h(x_i) \beta - t_i + \varepsilon_i = 0, i = 1, 2, \dots, N \quad (4.3)$$

By substituting (4.1) and (4.2) into (4.3), we obtain the closed form solution to β

$$\beta \begin{cases} = \left(\frac{1}{\gamma} + H^T W H \right)^{-1} H^T W T, & N \geq L \\ = H^T \left(\frac{1}{\gamma} + W H H^T \right) W T, & N < L \end{cases} \quad (5)$$

where I is identity matrix and $T = [t_1, t_2, \dots, t_N]$. L is the number of hidden nodes and N represents the number of features of input samples. For binary classification problems, WELM needs only one output node, and the decision function is

$$f(x) = \text{sign}(h(x) \beta) = \text{sign} \left(\begin{cases} h(x) \left(\frac{1}{\gamma} + H^T W H \right)^{-1} H^T W T \\ h(x) H^T (1/\gamma + W H H^T) W T \end{cases} \right) \quad (6)$$

Similar to traditional ELM, the regularized weighted extreme learning machine produce suboptimal classification model due to random assignment of input weights and hidden neuron biases.

2.2 Artificial bee colony (ABC) algorithm

Artificial bee colony algorithm, proposed by Karaboga for numeric function optimization [36,37], is inspired by the foraging behavior of bee swarms, which is a competitive and the latest swarm intelligence algorithm. Artificial bee colony consists of three groups: employed bees, onlooker bees and scout bees, in which the first two are of equal amount. To be specific, employed bees exploit the nectar sources in the hive and give information about the quality of the food source to the onlooker bees. Onlooker bees, according to the information shared by the employed bees, decide on a food source to exploit. Scout bees randomly search the environment in order to find a new food source when any food source is abandoned. After initialization phase, the search process of the employed bee phase, the onlooker bee phase and the scout bee phase is repeated until the goal is met or the iteration times are over maximum generation (MaxGen). In ABC algorithm, a possible solution to the optimization problem is expressed by the position of a food source.

ABC works by randomly initializing the positions of food sources searched by the employed bee or its corresponding onlooker bee. Each employed bee is associated with only

one food source position. Food source x_i is generated using a uniform distribution as:

$$x_{i,j} = x_j^{\min} + \text{rand}(0, 1) \cdot (x_j^{\max} - x_j^{\min}) \quad (7)$$

where $i \in (1, 2, \dots, SN)$ and $j \in (1, 2, \dots, D)$. SN represents the number of food source and D represents the dimension of optimization parameters. x_j^{\min} and x_j^{\max} denote boundaries of j th dimension and $\text{rand}(0,1)$ is a uniformly distributed random number in the range $[0,1]$.

At first, all employed bees are set out to explore the food source positions. According to local information in its memory, an employed bee updates the position at v_i of a neighboring food source using following equation

$$v_{i,j} = x_{i,j} + \phi_{i,j}(x_{i,j} - x_{k,j}) \quad (8)$$

$k \in \{1, 2, \dots, SN\}$ and $j \in \{1, 2, \dots, D\}$ are randomly chosen index that has to be different from i . $\phi_{i,j}$ is a uniformly distributed random number in the range $[-1, 1]$.

Afterwards, a greedy selection strategy is applied. If the quality of the food source at v_i is superior to that of x_i , the employed bee will memorize the new position v_i and abandon the old one x_i , or the employed bee remains at the previous position. According to the information about the quality of the food source positions shared by the employed bees, onlooker bees either randomly decide on a food source to exploit or ignore according to probability (roulette selection strategy) defined as follows

$$p_i = \frac{0.9 * \text{fitness}_i}{\text{fitness}_{\text{best}}} + 0.1 \quad (9)$$

where fitness_i represents the quality of the food source at positions v_i (the objective function of solution) and $\text{fitness}_{\text{best}}$ represents the best quality of all food source. It is obvious that higher fitness_i enjoys higher probability of being selected by the corresponding onlooker bee.

These onlooker bees update the food source position around corresponding employed bees by Eq. (8) and randomly decide on a food source to exploit by Eq. (9). The number of inefficient searching iterations in the process of iterative optimization is written into the control parameter $\text{trial}(i)$ $\{i \in (1, 2, \dots, SN)\}$ before better position of the each employed bee and corresponding onlooker bee is derived. If the i -th position of employed bee or onlooker bee is updated, $\text{trial}(i)$ is set to zero; otherwise, it is added by one. The food source is abandoned and the corresponding bee becomes a scout bee if a food source is not updated for a fixed number limit. The abandoned food source is replaced by a randomly chosen food source by the Eq. (7).

3 Hybrid artificial bee colony optimization-based weighted extreme learning machine

An important paradigm of parameter optimizations for single layer feed-forward neural networks (SLFN) is evolutionary computation [46–49]. The representative methods using evolutionary computation to optimize ELM parameter is E-ELM [26], which adopts differential evolutionary method to select the hidden node parameters and uses ELM to calculate the output weights. However, E-ELM relies on the error function between the network approximate output and the expected output to evaluate all the populations. For balanced class distribution, these approaches work very well, but when faced with imbalanced data, error can cause the evolution of solutions biased toward the majority class. Yet, it is the performance on the minority class data that is usually of utmost importance. Moreover, while the WELM solves the L^2 -regularized weighted least squares problem and assigns more weight to minority class and less weight to majority to obtain better generalization performance, the randomly-chosen input weights and hidden neuron biases remain unchanged during the training phase and many non-optimal nodes may exist and contribute less to minimizing the cost function and the weight generated according to class distribution of training samples depends on input samples.

As mentioned above, combining ABC with DE algorithm can improve the performance of basic ABC. This paper aims to propose the HABC-WELM approach that can: (1) use HABC to train WELM to find optimal input weight, hidden bias and the weight assigned to training samples. (2) evolve classifiers in an imbalanced environment without relying on error (the hybrid artificial bee colony is driven by maximizing AUC rather than minimizing overall classification error rate during the training phase), (3) obtain better classification performance than WELM and (4) based on commonly used classification benchmark datasets, present a quantitative performance comparison with WELM, EN-WELM, DE-WELM, ABC-WELM, WSVM, RUSBoost, EsayEnsemble, OverBagging and UnderBagging.

DE algorithm relies on mutation operation constructing better solutions. After generating all the mutant vectors, a crossover operator is used to increase the diversities of the perturbed parameter vectors. The mutation operation of DE is based on the weighted difference of randomly sampled pairs of solutions in the population, and DE algorithm might lead to better perturbation than the strategies with only one weighted difference vector when two weighted difference vectors are added to the base vector. In addition, it is observed that the exploitation capability of ABC can be improved by compelling less fit solutions to follow the best ever found solution [47,48]. We get the solution search equation of hybrid artificial bee colony (HABC) by combining crossover operation

of DE with the best ever found solution of ABC.

$$v'_{ij} = \begin{cases} x_{i,j} + F(x_{best,j} - x_{i,j}) \\ + F(x_{k,j} - x_{m,j}) & \text{if } (rand < CR) \text{ or } (j = j_{rand}) \\ v_{i,j} & \text{otherwise} \end{cases} \quad (10)$$

where i, k, m is distinct integers randomly selected from the range $[1, SN]$. $x_{best,j}$ is the best solution in the current swarm. F is positive amplification factor and CR is the crossover rate. j_{rand} is a randomly chosen integer from $[1, D]$.

The hybrid artificial bee colony (HABC) described above is now applied to optimize input weights, hidden bias of network and the weight assigned to training samples in WELM. HABC is directly adapted as a training method for feed-forward networks where input weights, hidden bias of network and the weight assigned to training samples are encoded into position vector of food source. The detailed steps of the proposed method are as follow:

Firstly, food sources are randomly generated and composed of a set of NP vectors:

$$F_q = [a_{11}, a_{12}, \dots, a_{1N}, a_{21}, a_{22}, \dots, a_{2N}, \dots, a_{L1}, a_{L2}, \dots, a_{LN}, b_1, b_2, \dots, b_L, w_{11}, w_{22}, w_{33}, \dots, w_{NN}] \quad (11)$$

where a_j and $b(j = 1, \dots, L)$ are input weights vector and biases vector generated randomly from the uniform distribution within the range of $[-1, 1]$ and $[0, 1]$ respectively. $w_{ii}(i = 1, \dots, N)$ are the diagonal element of weight matrix generated according to class distribution of training samples. L denotes the number of hidden neurons and N denotes the number of samples, q is integer within the range of $[1, NP]$.

Secondly, for each food source, the corresponding output weights of SLFN are computed according to Eq. (5). All employed bees are set out to explore the food source positions. According to local information in its memory, an employed bee updates the position at v_i of a neighboring food source according to Eq. (10). The quality of the food source is evaluated by a greedy selection strategy. In order to evaluate the fitness of each food source, it requires to select an appropriate evaluation metric as fitness function. G-mean is a measure of the ability of a classifier to balance sensitivity and specificity. Due to this symmetric nature of the distribution of the G-man over sensitivity and specificity, it's hard to contrast different models in accordance with their precision on each class [49]. F-measure is the weighted harmonic mean of the sensitivity and precision. So the G-mean and F-measure criterion can lead to the selection of sub-optimal models. AUC is defined as a plot of a model's true positive rate on the y-axis against its false positive rate on the x-axis, offering an overall measure of model performance, regardless of the different thresholds used. When class distribution

is imbalanced, the area under the curve (AUC) is the better metric to measure performance of fitness function [50]. So the AUC on the validation set is used as the fitness function to evaluate all the population and then WELM is used to tune better the candidate parameters obtained by HABC and move forward to the optimum.

Thirdly, with the fitness value of the i -th food source, the fitness of new position $v'_{i,j}$ corresponding to the current position $x_{i,j}$ was computed according to Eq. (10). Based on the knowledge that smaller norm of output weights could lead to better generalization ability for ELMs [21,51], the norm of output weights along with the AUC on the validation set are used as one more criterion evaluating food source. The work mainly focused on restricting the input weights and hidden biases to a reasonable range and assigning optimal weight to training data to improve the classification performance of WELM, thus the greedy selection operation can be expressed as Eq. (12).

$$v'_{i,j} = \begin{cases} v'_{i,j}, & \text{if } (f(v'_{i,j}) - f(x_{i,j})) > \epsilon \cdot f(x_{i,j}), \\ v'_{i,j}, & \text{if } |f(v'_{i,j}) - f(x_{i,j})| < \epsilon \cdot f(x_{i,j}) \text{ and } \|\beta_{v'_{i,j}}\| < \|\beta_{x_{i,j}}\|, \\ x_{i,j}, & \text{otherwise.} \end{cases} \quad (12)$$

where $f(v'_{i,j})$ and $f(x_{i,j})$ are the corresponding fitness values for new position $v'_{i,j}$ and the current position $x_{i,j}$ respectively. $\|\beta_{v'_{i,j}}\|$ and $\|\beta_{x_{i,j}}\|$ are the norm of the corresponding output weights obtained by ABC-WELM when the input weights are set as the i -th food source, $\epsilon > 0$ is a tolerance rate.

Finally, the above optimization process is repeated and parameters are progressively updated until the convergence criterion is reached or the maximum learning iterations are completed. Thus, the WELM network with the optimal input weights, hidden biases and the weight assigned to each class sample are obtained, and then the optimal WELM is applied to the test data.

4 Experiments

In this section, we observe the variance of AUC and norm of output weights during the training phase of proposed HABC-WELM approach. Afterwards, we present details of the comparison with WELM, EN-WELM, DE-WELM, ABC-WELM, WSVM, RUSBoost, EsayEnsemble, OverBagging and UnderBagging for 15 imbalanced datasets.

4.1 AUC and norm of output weights during the training process

In this subsection, different from the evaluation approaches adopted by most methods, where random permutation is used

Table 1 Specification of imbalanced datasets

Datasets	#Ex.	#Atts.	Minority class	IR	#Training	#Test	#Validating
Glass0	214	9	Building windows	0.4861	111	58	55
Glass2	214	9	Vehicle windows	0.0862	107	50	57
Glass6	214	9	Headlamps	0.1567	103	54	57
Ionosphere	351	34	Bad	0.5600	164	92	95
Ecoli0	336	7	cp	0.7409	168	83	85
Ecoli1	336	7	im	0.2972	166	81	89
Ecoli3	336	7	imU	0.1162	170	86	80
Vehicle0	846	18	Van	0.3075	448	179	219
Wisconsin	683	9	Malignant	0.5382	328	170	185
Yeast1	1484	8	nuc	0.4066	747	311	366
Yeast3	1484	8	ME3	0.1233	741	370	373
Pima	768	8	Class0	0.5360	379	189	200
Abalone9vs18	731	8	Class9	0.0609	356	196	179
Abalone19	4174	8	Class19	0.0077	2117	1013	1044
Pageblock1	5473	10	Horiz. line	0.0639	2767	1337	1369

for each run, we use the same training, testing and validation data set in order to solely analyze the norm of output weights and AUC of four approaches where AUC is used as the fitness function in this subsection to better observe the variance of AUC and norm of output weights. 15 benchmark datasets [52–54] used in imbalanced problem are selected. Table 1 gives the characteristics of these datasets. The imbalance ratio (IR, defined as the number of minority class examples divided by the number of majority class examples) of datasets varies from 0.0077 to 0.7409. 10 runs are done to study the stability of three approaches with data unchanged, thus eliminating the performance fluctuation caused by different data partition.

All simulations are carried out by Matlab R2013b on a windows 10 with an Intel Core i7 CPU 4510U @ 2.80GHz and 8G RAM. The performance of the proposed HABC-WELM is evaluated in details with comparison to three WELM-based methods, namely the WELM, ABC-WELM and DE-WELM. The input weights, hidden bias and the weight assigned to training samples of DE-WELM are optimized by differential evolution algorithm. The input weights, hidden bias and the weight assigned to training samples of ABC-WELM are optimized by basic artificial bee colony algorithm. All WELM-based approaches were implemented with sigmoid activation function. There are two parameters to tune for WELM-based approaches, the trade-off constant γ and the number of hidden nodes L . γ and L are searched in a wide range $\{2^{-50}, 2^{-11}, \dots, 2^{50}\}$ and $\{10, 20, \dots, 500\}$ respectively in seek of the optimal result. For DE-WELM, F (stepsize) and CR (crossover probability) are set to 1 and 0.8. Size of population is 50, and the maximum number of cycles is 100. For ABC-WELM and HABC-WELM, SN (the number of food sources) is

20, limit is 100 and the maximum number of cycles is 100.

We use G-mean as the evaluation metric to compare models in our case because of its balanced nature and usage in previous works [18]. In addition, G-mean is the trade-off between sensitivity and specificity, whereas the minority ones often represent the main class of interest in some scenarios. Therefore, to evaluate the effectiveness of the competing methods, we also give the results of sensitivity and specificity. The best average classification results in terms of G-mean and corresponding AUC, SN, SP are given in Table 2. For each evaluation metric, the standard deviation is also given after the \pm sign.

Figures 1 and 2 shows the effectiveness of the evaluation metric (AUC) and the variance of norm in DE-WELM, ABC-WELM and HABC-WELM during the training process. As the differential evolution algorithm, artificial bee colony algorithm and hybrid artificial bee colony algorithm are stochastic learning algorithms, each experiment is repeated ten runs using a different random seed. Therefore, Figs. 1 and 2 report the average AUC and norm of the fittest solution in the population over the 100 generations. The best results in terms of the G-mean, and corresponding AUC, SN, SP and standard deviation (STD) respectively.

Figure 1 shows that the AUC of three approaches increase smoothly over all datasets except for Ecoli0, which suggests that training is not an issue for most datasets. As can be seen from Fig. 1 and Table 2, the G-mean of these three methods are higher than those of WELM despite the fact that there is no variance for AUC of Ecoli0 over 100 generations in HABC-WELM, ABC-WELM and DE-WELM, which illustrates that the evaluation metric (AUC) evolves classifiers with best solution at the first generation. Although

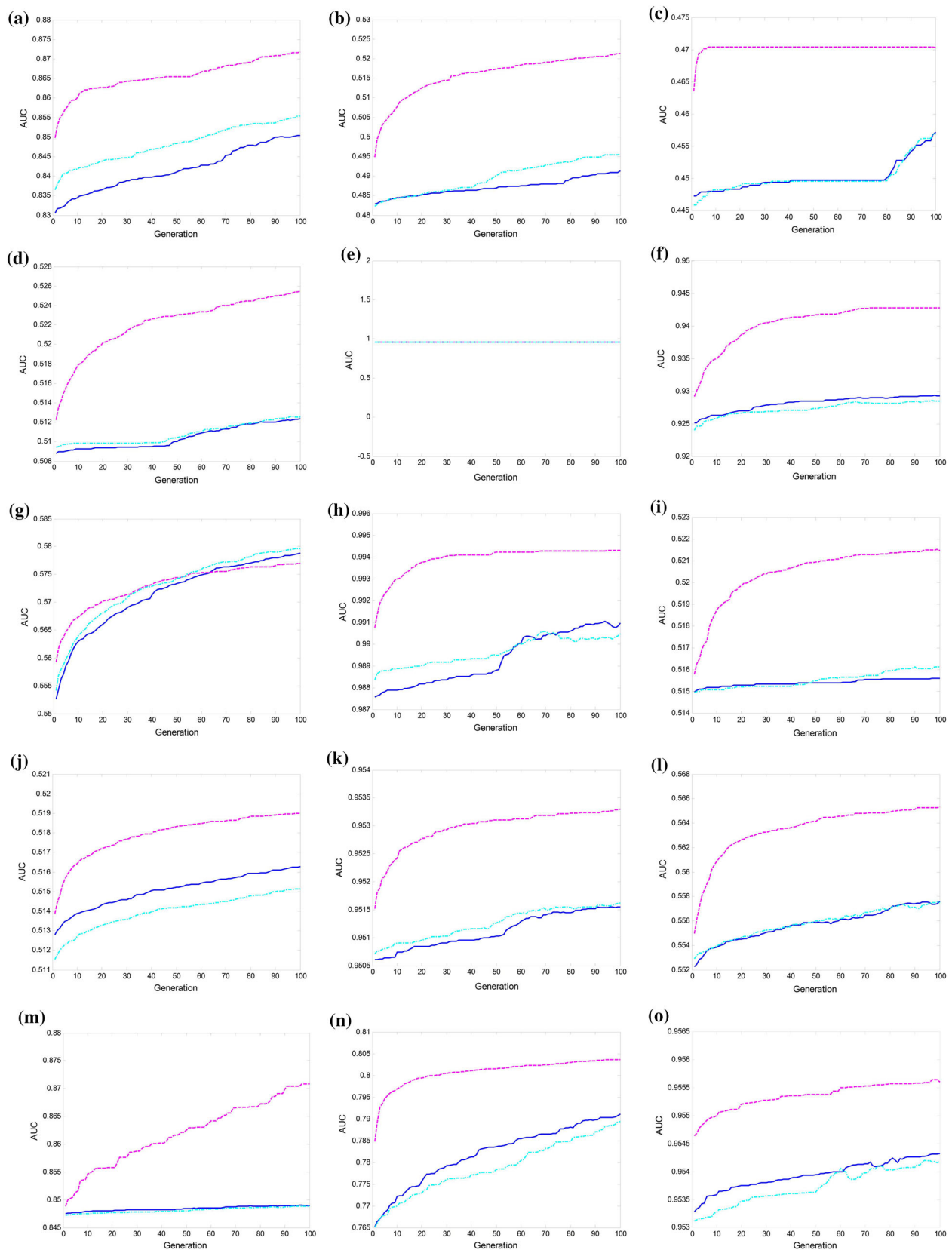


Fig. 1 Average AUC during the fittest evolved solution with DE-WELM (magenta line), ABC-WELM (blue line) and HABC-WELM (cyan line) over 100 independent runs. **a** Glass0, **b** Glass2, **c** Glass6,

d Ionosphere, **e** Ecoli0, **f** Ecoli1, **g** Ecoli3, **h** Vehicle0, **i** Wisconsin, **j** Yeast1, **k** Yeast3, **l** Pima, **m** Abalone9vs18, **n** Abalone19, **o** Page-block1 (Color figure online)

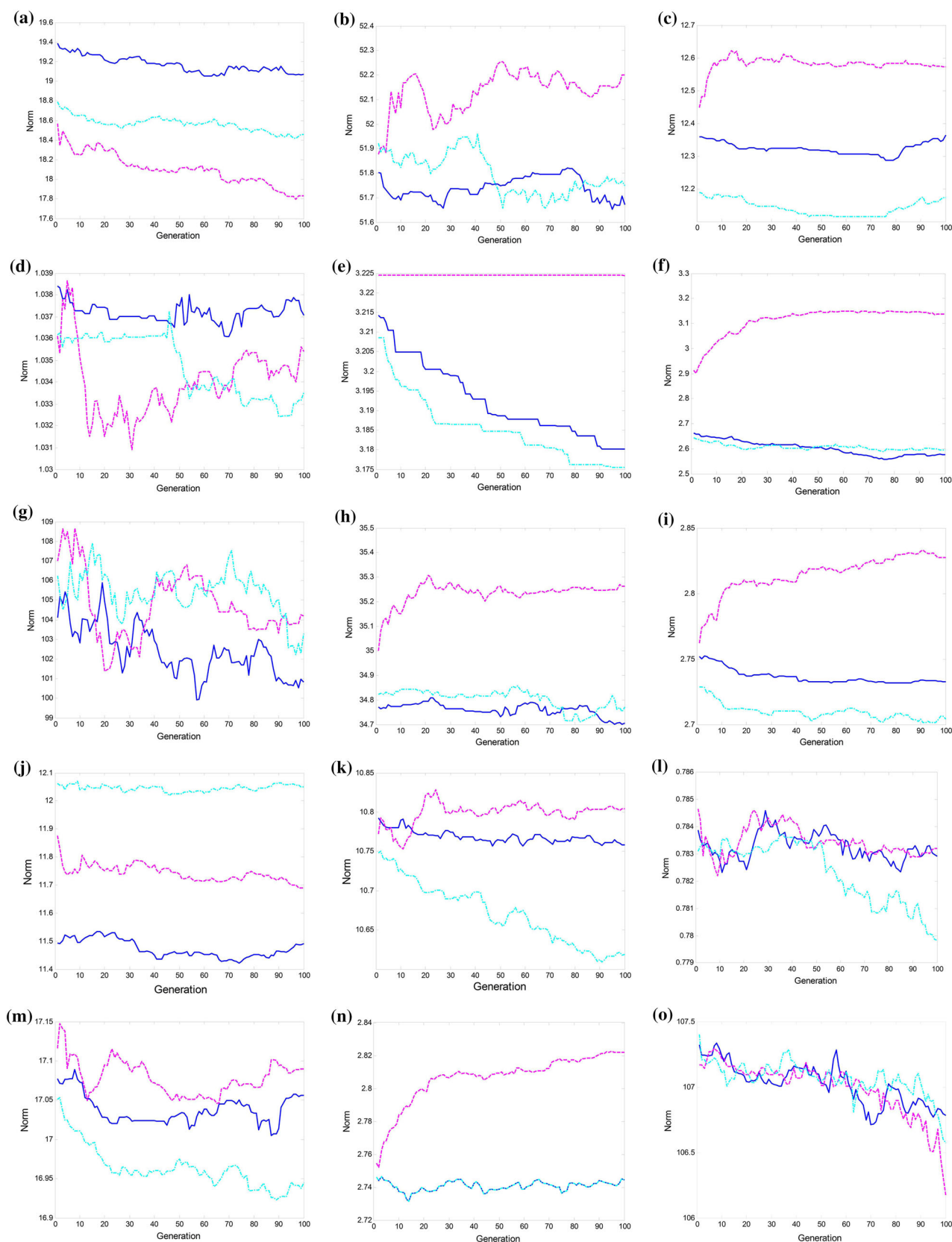


Fig. 2 Average norm of output weights with DE-WELM (magenta line), ABC-WELM (blue line) and HABC-WELM (cyan line) over 100 independent runs. **a** Glass0, **b** Glass2, **c** Glass6, **d** Ionosphere, **e** Ecoli0,

f Ecoli1, **g** Ecoli3, **h** Vehicle0, **i** Wisconsin, **j** Yeast1, **k** Yeast3, **l** Pima, **m** Abalone9vs18, **n** Abalone19, **o** Pageblock1 (Color figure online)

there are some fluctuations, a trend can still be observed from the curves in Fig 2, i.e., the norm of the output weights of the SLFN are decreasing in HABC-WELM and ABC-WELM for all datasets as well as in DE-WELM for seven datasets during the training process, which is consistent with the theory that the neural networks tend to have a better generalization performance with smaller norm of output weights. Besides, the norm of the output weights of the SLFN are increasing over 8 datasets in DE-WELM, but the testing G-mean are higher than those of the WELM over all datasets from Table 2, which suggests that the optimization of the network hidden node parameters is determined not only by

the area under curve(AUC) but also by the norm of the output weights.

From Table 2, the HABC-WELM shows the best performance in terms of G-mean over 15 datasets. the testing G-mean and SN of the DE-WELM are higher than those of the ABC-WELM over glass0, yeast1, pageblock1. However, the testing G-mean and SN of the HABC-WELM are higher than those of the ABC-WELM and DE-WELM over all data sets. It can be seen that the HABC-WELM offers very competitive results compared with the original WELM, DE-WELM and ABC-WELM in terms of producing higher testing AUC, G-mean and SN as well as lower STD over

Table 2 Performance of four methods on 15 imbalanced classification problem

Datasets	Methods	AUC	G-mean	SN	SP	γ	L
Glass0	WELM	83.18 ± 1.48	81.06 ± 1.68	88.78 ± 1.32	74.02 ± 2.33	2^3	270
	DE-WELM	85.27 ± 1.57	84.90 ± 1.52	93.78 ± 1.24	76.92 ± 2.17	2^3	270
	ABC-WELM	84.21 ± 1.40	83.37 ± 1.51	91.62 ± 1.15	76.92 ± 1.85	2^3	250
	HABC-WELM	86.06 ± 1.40	85.32 ± 1.53	94.33 ± 1.11	75.98 ± 1.97	2^3	260
Glass2	WELM	73.67 ± 1.44	79.61 ± 1.23	80.00 ± 3.56	79.35 ± 1.93	2^4	180
	DE-WELM	73.97 ± 1.11	79.88 ± 1.10	80.29 ± 3.32	79.77 ± 1.77	2^4	180
	ABC-WELM	74.04 ± 1.20	80.07 ± 1.15	80.43 ± 3.05	79.90 ± 1.64	2^4	160
	HABC-WELM	74.18 ± 1.15	80.42 ± 0.88	80.80 ± 2.41	79.97 ± 1.57	2^4	190
Glass6	WELM	98.23 ± 0.76	98.21 ± 0.75	99.87 ± 1.51	96.59 ± 2.10	2^{-2}	280
	DE-WELM	99.00 ± 0.69	98.99 ± 0.70	100 ± 0	97.98 ± 1.52	2^{-2}	260
	ABC-WELM	99.04 ± 0.65	99.01 ± 0.92	100 ± 1.35	98.02 ± 1.35	2^{-2}	250
	HABC-WELM	99.10 ± 0.59	99.36 ± 0.61	100 ± 0	98.90 ± 1.18	2^{-2}	230
Ionosphere	WELM	93.86 ± 1.64	88.32 ± 2.26	83.29 ± 3.62	93.72 ± 1.90	2^0	300
	DE-WELM	94.10 ± 1.47	88.62 ± 2.12	84.32 ± 3.25	93.11 ± 1.77	2^0	280
	ABC-WELM	94.21 ± 1.51	88.99 ± 2.03	84.72 ± 3.05	93.79 ± 1.39	2^0	260
	HABC-WELM	94.34 ± 1.37	89.27 ± 1.70	84.52 ± 3.00	93.96 ± 1.24	2^0	260
Ecoli0	WELM	$99.94 \pm 6.3075e-4$	97.19 ± 0.75	98.66 ± 1.51	$95.74 \pm 6.1093e-15$	2^2	180
	DE-WELM	$99.97 \pm 5.7455e-4$	97.56 ± 0.69	99.33 ± 1.30	$95.74 \pm 3.2363e-15$	2^2	160
	ABC-WELM	$99.98 \pm 5.3341e-4$	97.59 ± 0.60	99.37 ± 1.27	$95.47 \pm 3.2363e-16$	2^2	180
	HABC-WELM	$99.99 \pm 5.0402e-4$	97.78 ± 0.58	99.64 ± 1.18	$95.74 \pm 3.2363e-16$	2^2	160
Ecoli1	WELM	88.74 ± 0.99	88.72 ± 1.96	90.42 ± 2.03	87.06 ± 0.23	2^1	180
	DE-WELM	88.87 ± 0.90	89.21 ± 1.89	92.10 ± 1.85	87.12 ± 0.20	2^1	160
	ABC-WELM	88.92 ± 0.92	89.53 ± 1.87	91.91 ± 1.90	87.19 ± 0.25	2^1	180
	HABC-WELM	88.94 ± 0.87	89.77 ± 1.69	92.15 ± 1.88	87.08 ± 0.20	2^1	160
Ecoli3	WELM	85.71 ± 3.07	90.25 ± 2.50	92.82 ± 5.37	87.25 ± 2.03	2^5	150
	DE-WELM	86.69 ± 2.57	91.11 ± 2.27	95.60 ± 4.92	87.31 ± 1.70	2^5	120
	ABC-WELM	86.87 ± 2.62	91.24 ± 1.99	95.79 ± 5.13	86.98 ± 1.83	2^5	120
	HABC-WELM	86.94 ± 2.52	91.21 ± 2.14	96.03 ± 4.86	86.99 ± 1.78	2^5	230
Vehicle0	WELM	98.08 ± 0.54	98.06 ± 1.50	99.95 ± 1.25	96.17 ± 0.94	2^4	360
	DE-WELM	98.17 ± 0.41	98.43 ± 1.40	99.97 ± 1.31	96.30 ± 0.81	2^4	360
	ABC-WELM	98.29 ± 0.43	98.54 ± 1.35	99.98 ± 1.20	96.23 ± 0.74	2^4	340
	HABC-WELM	98.37 ± 0.37	98.71 ± 0.37	100 ± 0.26	97.26 ± 0.72	2^4	320

Table 2 continued

Datasets	Methods	AUC	G-mean	SN	SP	γ	L
Winsconsin	WELM	97.81 ± 0.33	95.53 ± 0.85	93.54 ± 1.50	97.56 ± 0.46	2^{-2}	170
	DE-WELM	97.91 ± 0.30	95.77 ± 0.80	93.69 ± 1.20	97.97 ± 0.46	2^{-2}	170
	ABC-WELM	97.96 ± 0.32	96.05 ± 0.83	94.15 ± 1.30	97.99 ± 0.40	2^{-2}	170
	HABC-WELM	97.99 ± 0.27	96.21 ± 0.80	94.15 ± 1.11	97.62 ± 0.40	2^{-2}	160
Yeast1	WELM	73.50 ± 0.44	75.02 ± 0.45	78.73 ± 0.63	71.49 ± 0.69	2^3	220
	DE-WELM	73.98 ± 0.36	75.62 ± 0.35	79.61 ± 0.54	71.96 ± 0.46	2^3	170
	ABC-WELM	73.69 ± 0.42	75.33 ± 0.40	78.98 ± 0.62	71.65 ± 0.60	2^3	170
	HABC-WELM	74.11 ± 0.37	75.41 ± 0.34	79.79 ± 0.54	71.15 ± 0.56	2^3	190
Yeast3	WELM	89.07 ± 0.74	89.47 ± 0.77	85.96 ± 1.48	93.18 ± 0.24	2^6	170
	DE-WELM	89.18 ± 0.65	89.59 ± 0.68	86.30 ± 1.27	93.18 ± 0.24	2^6	140
	ABC-WELM	89.29 ± 1.0	89.81 ± 0.62	86.49 ± 1.17	93.23 ± 0.24	2^6	140
	HABC-WELM	89.80 ± 0.60	89.94 ± 0.58	86.51 ± 1.12	93.10 ± 0.27	2^6	150
Pima	WELM	74.30 ± 0.81	69.88 ± 1.16	57.10 ± 1.67	85.54 ± 1.01	2^0	280
	DE-WELM	74.73 ± 0.69	70.25 ± 0.95	57.66 ± 1.25	85.29 ± 0.92	2^0	290
	ABC-WELM	74.80 ± 0.61	70.37 ± 0.79	57.69 ± 1.29	84.84 ± 0.91	2^0	240
	HABC-WELM	74.90 ± 0.60	70.58 ± 0.70	59.07 ± 1.31	85.44 ± 0.92	2^0	250
Abalone9vs18	WELM	88.54 ± 2.70	88.54 ± 2.82	87.20 ± 5.33	90.06 ± 0.54	2^{13}	150
	DE-WELM	88.61 ± 2.52	88.47 ± 2.51	87.64 ± 4.77	90.11 ± 0.50	2^{13}	130
	ABC-WELM	88.66 ± 2.61	88.57 ± 2.59	86.90 ± 4.69	90.32 ± 0.53	2^{13}	210
	HABC-WELM	88.91 ± 2.58	88.95 ± 2.50	87.80 ± 4.56	90.41 ± 0.50	2^{13}	230
Abalone19	WELM	77.92 ± 2.90	63.38 ± 3.65	51.44 ± 6.27	78.44 ± 2.07	2^1	150
	DE-WELM	78.14 ± 2.69	64.33 ± 3.40	51.65 ± 5.95	80.96 ± 1.39	2^1	150
	ABC-WELM	78.29 ± 2.70	65.71 ± 3.49	53.08 ± 6.00	81.29 ± 1.41	2^1	160
	HABC-WELM	78.80 ± 2.58	65.90 ± 3.41	54.22 ± 5.76	80.34 ± 1.12	2^1	180
Pageblock1	WELM	93.64 ± 0.54	93.97 ± 0.31	91.86 ± 0.58	96.13 ± 0.18	2^5	170
	DE-WELM	94.16 ± 0.43	94.59 ± 0.26	92.96 ± 0.50	96.50 ± 0.12	2^5	200
	ABC-WELM	93.93 ± 0.45	94.24 ± 0.30	92.51 ± 0.52	96.34 ± 0.15	2^5	160
	HABC-WELM	94.97 ± 0.30	95.25 ± 0.27	93.15 ± 0.50	96.08 ± 0.13	2^5	170

most datasets. Particularly, the HABC-WELM outperforms the ABC-WELM in all tests.

According to the experimental results on all data sets, there is the same value of γ and no much difference in the value of L when WELM, DE-WELM and HABC-WELM obtain the best results in term of G-mean. Therefore, one can fix γ as the best value and tune L when selecting the parameter of model, thus the time for parameter selection has been largely shortened.

Since this simulation is solely aimed to analyze the variance of norm and AUC of three approaches in which AUC is used as the fitness function, only one training, testing and validation datasets are randomly generated at each trial of simulations according to Table 1. Therefore, the value in Table 2 may not be a true reflection of performance. More thorough tests are done in the subsection simulation later.

4.2 Performance evaluation of proposed HABC-WELM

Hereafter, once we have shown and selected our better proposal, we analyze whether the classification performance of our proposed method is better than that of previous methods. To thoroughly evaluate the performance of HABC-WELM, four WELM-based methods, WSVM and four popular ensemble learning based on SVM base classifier, such as RUSBoost, EsayEnsemble, overBagging, underBagging are tested synchronously. We use 15 data sets in Table 1. For above nine methods, the evaluation procedure consists of two cycles in order to avoid over-fitting for final generated classifier. In the inner cycle, the five-fold cross-validation approach (80% samples are extracted into original training set and the rest ones are used for testing each time. The original training set is randomly divided into two groups: training set (2/3) and validation set (1/3)) is used. Each algorithm based

Table 3 Performance evaluation from fivefold cross validation over 10 runs

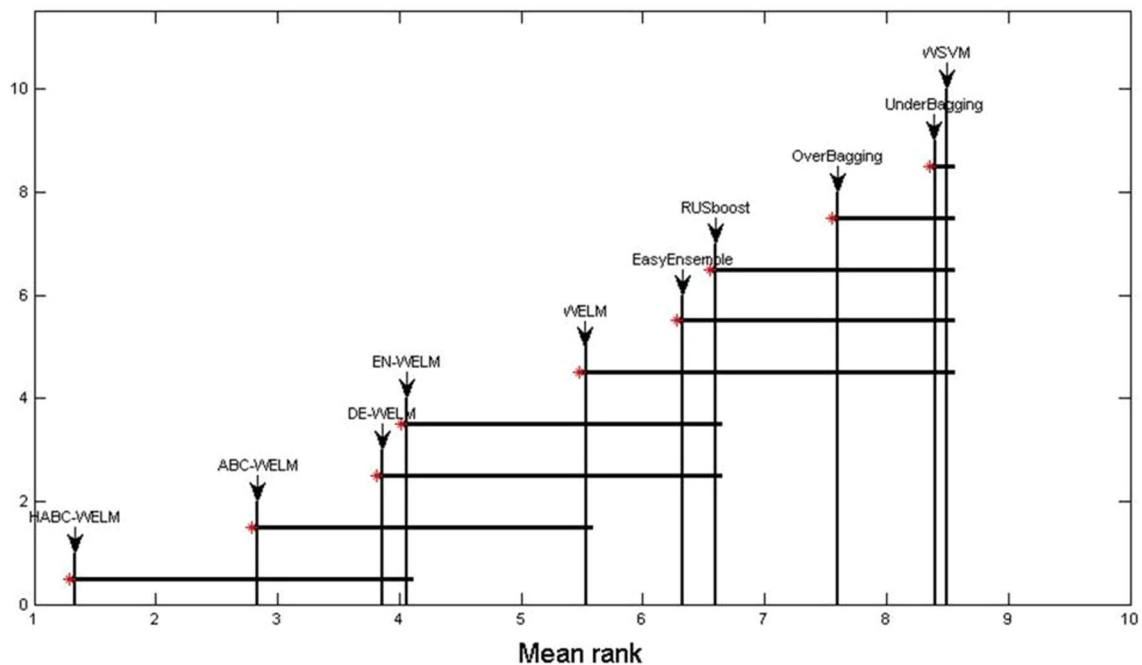
Dataset	Metric	WSVM	Bagging- based		Boost- based		WELM- based			DE-WELM	ABC-WELM	HABC-WELM
			OverBagging	UnderBagging	RUSBoost	EasyEnsemble	WELM	EN-WELM				
Glass0	AUC	98.15 ± 6.7e-16	97.09 ± 5.6e-16	97.99 ± 0.33	89.32 ± 2.15	84.28 ± 1.72	97.74 ± 0.36	98.28 ± 0.25	98.30 ± 0.25	98.33 ± 0.24	98.34 ± 0.22	
	G-mean	80.60 ± 4.5e-16	81.32 ± 2.2e-16	84.06 ± 1.59	80.56 ± 3.94	76.61 ± 3.56	82.14 ± 1.84	83.17 ± 1.52	83.29 ± 1.50	84.01 ± 1.32	84.69 ± 1.25	
	SN	77.81 ± 3.4e-16	88.11 ± 0	87.31 ± 1.14	92.54 ± 6.43	81.92 ± 4.73	90.81 ± 2.80	90.70 ± 2.12	90.77 ± 2.10	92.55 ± 1.51	92.88 ± 1.2	
	SP	85.64 ± 0	74.96 ± 3.3e-16	81.48 ± 2.69	70.47 ± 6.13	72.01 ± 5.27	74.57 ± 2.10	76.26 ± 1.64	76.76 ± 1.60	76.38 ± 1.82	76.37 ± 1.47	
	AUC	60.57 ± 2.8e-17	79.59 ± 0	81.02 ± 2.1	83.70 ± 6.13	74.26 ± 6.78	75.59 ± 1.50	75.72 ± 1.30	75.88 ± 1.71	75.90 ± 1.21	76.33 ± 1.19	
Glass2	G-mean	64.73 ± 0	75.63 ± 1.2e-16	77.58 ± 1.90	80.3 ± 11.97	68.74 ± 5.25	83.52 ± 1.39	83.61 ± 1.14	83.66 ± 1.14	83.98 ± 1.07	84.10 ± 0.98	
	SN	49.17 ± 0	80.36 ± 0	81.57 ± 3.	83.6 ± 25.20	88.50 ± 6.58	86.30 ± 0.85	86.49 ± 0.67	86.62 ± 0.61	87.43 ± 0.62	87.50 ± 0.59	
	SP	94.09 ± 6.7e-16	71.32 ± 0	73.97 ± 2.3	79.11 ± 4.82	54.33 ± 6.94	82.23 ± 1.89	82.17 ± 1.91	82.20 ± 1.83	82.61 ± 1.77	81.76 ± 1.90	
	AUC	60.96 ± 0	85.74 ± 0	86.08 ± 3.3e-16	97.02 ± 0.70	95.76 ± 0	89.39 ± 0	89.73 ± 0	89.75 ± 0	89.87 ± 0	89.91 ± 0	
	G-mean	73.73 ± 5.6e-16	92.20 ± 5.6e-16	92.57 ± 5.6e-16	94.25 ± 3.61	92.09 ± 1.33	94.27 ± 1.3e-16	94.62 ± 1.1e-16	94.71 ± 1.1e-16	94.77 ± 1.1e-16	94.89 ± 1e-16	
Ionosphere	SN	61.40 ± 0	87.62 ± 0	87.67 ± 3.3e-16	91.22 ± 6.34	90.66 ± 0	90.35 ± 5.7e-16	90.57 ± 0	90.60 ± 0	91.15 ± 0	91.26 ± 0	
	SP	99.46 ± 0	97.77 ± 0	98.11 ± 0	97.80 ± 3.02	94.05 ± 2.62	98.98 ± 0	98.92 ± 2.3e-17	99.07 ± 0	98.73 ± 0	98.99 ± 0	
	AUC	92.48 ± 0	92.65 ± 0	91.29 ± 5.6e-16	91.94 ± 1.76	94.50 ± 1.09	95.23 ± 1.89	95.54 ± 1.23	96.02 ± 0.61	95.70 ± 1.31	95.70 ± 1.02	
	G-mean	90.99 ± 3.4e-16	90.43 ± 5.6e-16	88.39 ± 4.5e-16	84.99 ± 3.50	86.00 ± 1.70	90.08 ± 2.97	90.73 ± 1.77	91.69 ± 1.88	91.14 ± 1.35	91.20 ± 1.47	
	SN	96.93 ± 3.4e-16	98.79 ± 0	98.67 ± 4.5e-16	79.65 ± 0.68	80.23 ± 1.81	82.79 ± 5.16	83.10 ± 4.72	85.43 ± 3.45	83.74 ± 2.97	83.69 ± 3.05	
Ecoli0	SP	85.58 ± 0	82.83 ± 0	79.30 ± 3.3 e-16	91.18 ± 5.89	92.85 ± 2.71	98.30 ± 0.90	99.13 ± 0.62	98.74 ± 0	98.99 ± 0.1.00	99.22 ± 0.80	
	AUC	99.86 ± 0	99.95 ± 0	99.95 ± 0.50	98.59 ± 0.55	97.64 ± 0.37	99.88 ± 0	99.95 ± 0	99.96 ± 0	99.98 ± 0	99.99 ± 0	
	G-mean	96.21 ± 0	97.41 ± 4.5e-16	97.42 ± 0.63	95.79 ± 2.43	95.62 ± 0.45	97.43 ± 0	97.74 ± 0	97.79 ± 0	97.96 ± 0	98.05 ± 0	
	SN	96.44 ± 0	98.02 ± 0	98.13 ± 1.27	97.61 ± 2.06	96.33 ± 0.87	97.82 ± 0	98.19 ± 0	98.21 ± 0	98.29 ± 0	98.40 ± 0	
	SP	95.99 ± 0	96.86 ± 4.5e-16	96.74 ± 4.5e-16	94.09 ± 3.86	94.97 ± 0	97.07 ± 0	97.40 ± 0	97.62 ± 0	97.71 ± 0	97.87 ± 0	
Ecoli1	AUC	84.18 ± 2.2e-16	90.69 ± 0.27	87.60 ± 1.0	94.86 ± 0.99	95.62 ± 0.58	93.32 ± 4.5e-16	93.40 ± 4.5e-16	93.69 ± 4.5e-16	93.85 ± 3.6e-16	93.92 ± 3.6e-16	
	G-mean	88.15 ± 3.4e-16	89.23 ± 0.45	87.07 ± 0.99	88.59 ± 2.16	89.23 ± 2.22	88.24 ± 4.5e-16	88.27 ± 4.5e-16	88.31 ± 5.1e-16	88.75 ± 4.9e-16	88.79 ± 4.9e-16	
	SN	75.37 ± 4.5e-16	85.28 ± 0	79.30 ± 1.69	92.70 ± 4.06	91.30 ± 3.81	92.07 ± 0	92.17 ± 3.7e-16	92.25 ± 3.7e-16	92.99 ± 3.0e-16	93.17 ± 2.3e-16	
	SP	94.48 ± 3.4e-16	93.80 ± 1.03	95.74 ± 0.78	84.81 ± 2.78	87.30 ± 1.76	85.12 ± 2.5e-16	84.75 ± 1.7e-16	85.21 ± 0	84.86 ± 0	84.90 ± 0	
	AUC	75.84 ± 4.5e-16	73.89 ± 1.1e-16	66.64 ± 3.58	93.53 ± 2.09	93.40 ± 1.52	84.05 ± 0.64	84.23 ± 0.34	84.57 ± 0.52	84.60 ± 0.41	84.65 ± 0.37	
Ecoli3	G-mean	81.60 ± 5.6e-16	81.00 ± 1.1e-16	73.96 ± 3.99	89.78 ± 3.10	86.60 ± 2.02	90.15 ± 0.46	90.32 ± 0.39	90.51 ± 0.35	90.70 ± 0.36	90.95 ± 0.37	
	SN	70.16 ± 3.4e-16	70.15 ± 0	56.74 ± 5.38	92.21 ± 4.10	86.11 ± 0	91.90 ± 0.23	92.39 ± 0.7	92.02 ± 0.21	92.63 ± 0.18	92.83 ± 0.13	
	SP	95.82 ± 4.5e-16	95.01 ± 0	97.67 ± 4.5e-16	88.78 ± 2.38	87.52 ± 3.66	88.56 ± 0.86	89.13 ± 0.17	88.80 ± 0.65	88.41 ± 0.48	88.21 ± 0.53	
	AUC	92.78 ± 0	94.69 ± 0.24	94.62 ± 3.3 e-16	94.85 ± 0.37	99.26 ± 0.30	98.24 ± 0.51	98.80 ± 0.50	98.83 ± 0.42	98.85 ± 0.47	99.07 ± 0.41	

Table 3 continued

Dataset	Metric	WSVM	Bagging- based		Boost- based		WELM- based			ABC-WELM	HABC-WELM
			OverBagging	UnderBagging	RUSBoost	EasyEnsemble	WELM	EN-WELM	DE-WELM		
Wisconsin	G-mean	92.13 ± 5.6e-16	94.74 ± 0.18	93.98 ± 2.2 e-16	94.90 ± 1.30	95.92 ± 0.50	98.46 ± 0.33	98.50 ± 0.29	98.50 ± 0.23	98.64 ± 0.25	98.79 ± 0.20
	SN	86.73 ± 0	91.65 ± 1.1e-16	89.96 ± 3.3 e-16	97.31 ± 2.38	97.11 ± 0	99.31 ± 0	99.32 ± 0	99.35 ± 0	99.59 ± 0	99.81 ± 0
	SP	97.94 ± 0	98.00 ± 0.36	98.23 ± 6.7 e-16	92.58 ± 1.35	94.80 ± 0.96	97.63 ± 0.65	97.68 ± 0.50	97.57 ± 0.40	97.67 ± 0.41	97.89 ± 0.40
	AUC	97.23 ± 0	96.85 ± 2.2e-16	97.05 ± 0.22	98.72 ± 0.58	99.08 ± 0.35	97.62 ± 0.20	97.81 ± 0.19	97.87 ± 0.29	97.81 ± 0.20	97.91 ± 0.18
Yeast1	G-mean	97.11 ± 3.4e-16	96.07 ± 3.3e-16	96.86 ± 0.11	97.42 ± 0.53	97.12 ± 1.11	97.14 ± 0.38	97.34 ± 0.32	97.37 ± 0.32	97.35 ± 0.30	97.47 ± 0.27
	SN	98.70 ± 4.5e-16	97.60 ± 0	98.37 ± 0	98.72 ± 0	97.21 ± 2.00	97.3 ± 0.62	98.09 ± 0.45	98.10 ± 0.48	97.90 ± 0.35	97.99 ± 0.33
	SP	95.56 ± 3.4e-16	94.60 ± 4.5e-16	95.39 ± 0.22	96.15 ± 1.02	97.07 ± 1.33	96.98 ± 0.23	97.81 ± 0.21	97.15 ± 0.16	96.95 ± 0.28	96.95 ± 0.21
	AUC	65.97 ± 3.4e-16	70.23 ± 0.42	70.62 ± 0.36	74.99 ± 2.25	78.78 ± 0.63	70.90 ± 0.62	72.17 ± 0.51	71.19 ± 0.60	72.25 ± 0.41	72.25 ± 0.40
Yeast3	G-mean	59.94 ± 1.1e-16	68.85 ± 0.48	68.89 ± 0.40	65.67 ± 2.12	71.32 ± 0.97	71.55 ± 0.66	71.88 ± 0.56	71.89 ± 0.61	71.95 ± 0.43	72.98 ± 0.41
	SN	38.67 ± 0	85.74 ± 0.85	85.63 ± 0.79	86.09 ± 7.15	71.98 ± 3.12	72.62 ± 1.06	73.26 ± 0.82	72.29 ± 1.0	73.47 ± 1.10	73.42 ± 0.97
	SP	93.71 ± 3.4e-16	55.34 ± 0.96	55.57 ± 0.94	50.88 ± 6.84	70.71 ± 2.03	70.53 ± 0.55	70.60 ± 0.64	70.63 ± 0.52	70.50 ± 0.46	70.62 ± 0.64
	AUC	78.93 ± 1.1e-16	85.87 ± 0.11	58.00 ± 6.8e-4	95.29 ± 0.42	96.55 ± 0.23	91.82 ± 0.73	92.10 ± 0.59	92.15 ± 0.10	91.91 ± 0.50	92.18 ± 0.56
Pima	G-mean	81.62 ± 5.6e-16	87.74 ± 9.8e-4	52.07 ± 4.3e-4	90.41 ± 1.48	91.47 ± 1.17	92.84 ± 0.63	93.17 ± 0.60	93.23 ± 9.7e-4	93.09 ± 0.47	93.71 ± 0.49
	SN	68.43 ± 3.8e-16	79.36 ± 3.3e-16	27.49 ± 0	92.18 ± 2.73	92.81 ± 2.42	92.89 ± 1.12	93.19 ± 0.96	93.16 ± 3.4e-16	93.25 ± 0.80	93.33 ± 0.35
	SP	97.48 ± 5.4e-16	97.14 ± 0.22	99.58 ± 0.16	88.83 ± 1.62	90.25 ± 0.78	92.73 ± 0.29	93.25 ± 0.23	92.88 ± 0	93.07 ± 0.16	93.30 ± 0.20
	AUC	71.76 ± 0	74.38 ± 0.97	68.76 ± 0.95	76.79 ± 1.53	83.12 ± 0.96	76.61 ± 1.14	76.83 ± 0.99	76.87 ± 0.86	80.15 ± 0.86	80.12 ± 0.80
Abalone9vs18	G-mean	73.30 ± 1.1e-16	73.27 ± 0.85	60.80 ± 1.11	72.37 ± 1.71	75.34 ± 1.51	75.56 ± 0.81	75.79 ± 0.76	75.81 ± 0.71	75.99 ± 0.64	76.07 ± 0.67
	SN	51.08 ± 1.1e-16	74.55 ± 1.68	40.99 ± 1.61	85.83 ± 1.94	74.55 ± 2.42	73.76 ± 1.28	73.94 ± 1.19	74.04 ± 1.04	74.60 ± 0.91	74.60 ± 0.85
	SP	91.69 ± 1.1e-16	72.19 ± 0.84	91.19 ± 0.70	61.57 ± 2.72	76.58 ± 1.67	77.62 ± 0.91	77.70 ± 1.05	77.50 ± 0.85	77.35 ± 0.73	77.35 ± 0.74
	AUC	80.21 ± 3.4e-16	12.23 ± 0	6.41 ± 0	86.98 ± 3.79	92.01 ± 2.49	73.77 ± 4.95	73.90 ± 8.09	73.94 ± 6.11	74.21 ± 6.14	74.31 ± 6.44
Abalone19	G-mean	75.66 ± 4.5e-16	30.61 ± 0	18.20 ± 0	85.15 ± 3.91	84.27 ± 3.21	85.44 ± 3.11	85.65 ± 4.93	85.71 ± 3.72	85.9 ± 3.79	85.97 ± 3.95
	SN	78.47 ± 0	12.23 ± 0	100 ± 0	86.74 ± 4.17	82.70 ± 5.38	80.93 ± 5.65	81.13 ± 8.90	81.40 ± 6.85	81.84 ± 6.95	81.94 ± 7.29
	SP	73.01 ± 6.7e-16	100 ± 0	6.41 ± 3.3e-16	84.24 ± 2.69	86.37 ± 2.47	91.17 ± 0.75	91.03 ± 0.78	90.79 ± 0.53	90.66 ± 0.27	90.69 ± 0.13
	AUC	75.34 ± 3.9e-16	80.66 ± 0.95	74.25 ± 0.36	77.56 ± 3.04	81.53 ± 1.65	78.36 ± 1.69	79.45 ± 1.59	79.51 ± 4.2e-4	80.62 ± 1.00	80.78 ± 1.02
Pageblock1	G-mean	64.89 ± 4.7e-16	71.08 ± 1.20	60.93 ± 0.59	67.93 ± 3.30	76.70 ± 2.13	73.84 ± 2.97	75.06 ± 2.61	75.56 ± 5.3e-4	76.49 ± 1.40	76.84 ± 1.38
	SN	75.27 ± 0	73.51 ± 2.08	68.15 ± 154	56.16 ± 4.36	82.57 ± 4.52	73.28 ± 5.07	73.49 ± 4.97	73.67 ± 4.1e-4	75.42 ± 2.00	76.09 ± 2.31
	SP	55.94 ± 3.4e-16	68.83 ± 1.99	54.49 ± 0.91	82.77 ± 2.15	72.03 ± 3.33	75.13 ± 3.25	76.85 ± 3.50	76.74 ± 0	77.79 ± 2.00	77.71 ± 0.78
	AUC	90.94 ± 2.2e-16	89.85 ± 0.48	74.00 ± 2.2e-16	98.70 ± 0.21	98.49 ± 0.56	95.34 ± 0.37	95.77 ± 0.40	95.79 ± 0	95.64 ± 0.20	95.67 ± 0.10
	G-mean	90.25 ± 4.5e-16	90.12 ± 0.45	65.86 ± 3.3e-16	94.99 ± 0.99	94.76 ± 0.91	95.27 ± 0.19	95.72 ± 0.20	95.80 ± 0	95.67 ± 0.08	95.91 ± 0.10
	SN	81.79 ± 4.5e-16	82.27 ± 0.84	44.00 ± 2.8e-16	95.47 ± 2.22	95.33 ± 1.82	95.09 ± 0.41	95.55 ± 0.49	95.61 ± 0	95.41 ± 0.10	95.41 ± 0.27
	SP	99.62 ± 4.5e-16	97.97 ± 0.12	99.94 ± 3.4e-16	94.53 ± 1.13	94.20 ± 0.47	95.46 ± 0.11	96.09 ± 0.17	95.58 ± 0	95.90 ± 0.10	95.90 ± 0

Table 4 Average ranks of the comparison between HABC-WELM and the other methods

p-value	WSVM	OverBagging	UnderBagging	RUSBoost	EasyEnsemble	WELM	EN-WELM	E-WELM	ABC-WELM	HABC-WELM
1.78e-14	8.50	7.60	8.40	6.60	6.33	5.53	3.86	4.06	2.83	1.33

**Fig. 3** Results of the pairwise comparisons of the ten methods using Nemenyi post hoc test with $\alpha = 0.05$

on WELM run 10 times considering that the sampling of subsets and input parameters of hidden nodes introduces randomness. The result is averaged over the five-fold. The outer cycle consists of 10 runs and the final result is averaged over 10 runs. In this way, the performance variance of all tests caused by different initial parameters and random partition of datasets can be both analyzed and the number of hidden neurons L and the trade-off constant γ in the WELM-based are chosen to ensure the use of optimal models. When gradually increasing the number of hidden nodes in preset region, the one with the highest test G-mean is selected as the suitable number of hidden nodes. The number of base classifier of EN-WELM is 10. SVM is implemented in LIBSVM. Resampling rate is set 5% in RUSBoost and EasyEnsemble. The number of iteration is set 10 in overBagging, underBagging, RUSBoost and EasyEnsemble. All other parameters for SVM is selected from the set $\{10^{-10}, \dots, 10^{10}\}$. The best average classification results in terms of G-mean and corresponding AUC, SN, SP are given in Table 3. For each evaluation metric the standard deviation is also given after the \pm sign. In this table, the best results on each data set are emphasized in bold.

From Table 3, some conclusion can be drawn as follows:

Firstly, the results show that the performance of the proposed method is the best on 13 of 15 datasets for G-mean,

and 8 of 15 datasets for SN. Secondly, compared with other WELM-based, the proposed method obtains the highest testing G-mean and SN except for EN-WELM over ionosphere. Apart from Glass2, Ecoli0, Ecoli3, Wisconsin and Yeast1, the SP of HABC-WELM increases significantly compared to WELM, which indicates that the proposed method can achieve better classification performance of both minority class and majority class on most datasets. Besides the fitness function (AUC) evolves classifiers with good performance on both the minority and majority class compared with WELM. In comparison with WELM, the STD of G-mean and SN caused by the random generation of input parameters in the proposed method has been reduced, which indicates that the proposed method is more stable than WELM. It can be seen that the HABC-WELM outperforms the original ABC-WELM. As what has been pointed out by many researchers, the solution search of ABC algorithm is good at exploration but poor at exploitation, which may result in premature convergence or stagnation. Therefore, incorporating the information of global best solution of ABC into the solution search equation to guide the search of new candidate solutions and exploiting the diversities of the perturbed parameter vectors of differential evolution may lead to better network generalization performances. Thirdly,

compared with overBagging, underBagging, RUSBoost and EsayEnsemble, the proposed method achieves the best G-mean over 14 datasets and the best SN over 10 datasets.

We analyze whether the proposed method performs better than other methods. To do so, we apply Friedman test that ranks the algorithm in term of testing G-mean for each data set separately. The significance level is taken as 0.05. The null hypothesis is that all the algorithms are equivalent and so their average ranks should be equal. P-value is the probability of observing Friedman test statistic as extreme as, or more extreme than, the observed value under the null hypothesis. Small p-values casts doubt on the validity of the null hypothesis. Average ranks computed for Friedman test is shown in Table 4. The test outputs a p-value of $1.31\text{e-}14$, which indicates that the hypothesis of equivalence can be rejected with high confidence. Hence, we continue a post-hoc test using Nemenyi test to identify which method significantly perform best, as suggested by Demasar [35,55].

Figure 3 plots the ten methods against average performance ranks, where all methods are sorted according to their ranks. The ‘*’ denotes the respective average rank of each method and the line segment to the right of each method represents its critical difference, which means the methods whose ‘*’ on the right end of the line are outperformed significantly. The ‘↓’ with vertical dotted line highlight the critical difference. The left vertical line and right vertical line is respectively associated with the best and the worst method. All methods right to this line perform significantly worse than this method and all methods left to this line perform significantly better than it. From Fig. 3, the proposed method is significantly better than another.

5 Conclusions

In this paper, a heuristic weighted extreme learning machine method named as HABC-WELM for imbalanced data classification was proposed, in which a hybrid ABC was used to optimize the input weights, hidden biases of WELM and the weight assigned to training sample. In the process of selecting the input weights and hidden biases, the metric applied to evaluate all populations is designed to AUC rather than the error between the network approximate output and the expected output. By 15 datasets, it has been demonstrated that the AUC metric is effective, and that the proposed method has better classification performance than the WELM, DE-WELM and EN-WELM, ABC-WELM, WSVM, RUSBoost, EsayEnsemble, overBagging and underBagging methods over most datasets. The future research works will include how to apply the proposed method to real-world imbalanced data mining applications such as identification of protein functional sites. Moreover, it is also interesting to extend

current HABC-WELM to multi-class datasets and some real world applications.

Acknowledgements This study was funded by National Key Technology Science and Technique Support Program (No. 2013BAH49f03).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Research Involving Human Participants or Animals This article does not contain any studies with human participants or animals performed by any of the authors.

Informed Consent Informed consent was obtained from all individual participants included in the study.

References

- Huang, G.B., Zhu, Q.Y., & Siew, C.K.: Extreme learning machine: a new learning scheme of feedforward neural networks. In: Proceedings of 2004 IEEE International Joint Conference on Neural Networks, Vol. 2, pp. 985–990. IEEE (2004)
- Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: theory and applications. *Neurocomputing* **70**(1), 489–501 (2006)
- Huang, G.B., Zhou, H., Ding, X., Zhang, R.: Extreme learning machine for regression and multiclass classification. *IEEE Trans. Syst. Man Cybern. Part B* **42**(2), 513–529 (2012)
- Huang, G., Huang, G.B., Song, S., You, K.: Trends in extreme learning machines: a review. *Neural Netw.* **61**, 32–48 (2015)
- Savojardo, C., Fariselli, P., Casadio, R.: BETAWARE: a machine-learning tool to detect and predict transmembrane beta-barrel proteins in prokaryotes. *Bioinformatics* **29**(4), 504–505 (2013)
- Savojardo, C., Fariselli, P., Casadio, R.: Improving the detection of transmembrane β -barrel chains with N-to-1 extreme learning machines. *Bioinformatics* **27**(22), 3123–3128 (2011)
- Fan, Y.X., Shen, H.B.: Predicting pupylation sites in prokaryotic proteins using pseudo-amino acid composition and extreme learning machine. *Neurocomputing* **128**, 267–272 (2014)
- Lan, Y., Soh, Y.C., & Huang, G.B.: Extreme learning machine based bacterial protein subcellular localization prediction. In: IJCNN 2008, IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pp. 1859–1863. IEEE (2008)
- Czarnecki, W.M.: Weighted tanimoto extreme learning machine with case study in drug discovery. *IEEE Comput. Intell. Mag.* **10**(3), 19–29 (2015)
- Vong, C.M., Ip, W.F., Wong, P.K., Chiu, C.C.: Predicting minority class for suspended particulate matters level by extreme learning machine. *Neurocomputing* **128**, 136–144 (2014)
- Saraswathi, S., Sundaram, S., Sundararajan, N., Zimmermann, M., Nilsen-Hamilton, M.: ICGA-PSO-ELM approach for accurate multiclass cancer classification resulting in reduced gene sets in which genes encoding secreted proteins are highly represented. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **8**(2), 452–463 (2011)
- Li, L.N., Ouyang, J.H., Chen, H.L., Liu, D.Y.: A computer aided diagnosis system for thyroid disease using extreme learning machine. *J. Med. Syst.* **36**(5), 3327–3337 (2012)
- Lin, S.J., Chang, C., Hsu, M.F.: Multiple extreme learning machines for a two-class imbalance corporate life cycle prediction. *Knowl.-Based Syst.* **39**, 214–223 (2013)

14. Baradarani, A., Wu, Q.J., Ahmadi, M.: An efficient illumination invariant face recognition framework via illumination enhancement and DD-DTCWT filtering. *Pattern Recognit.* **46**(1), 57–72 (2013)
15. Mohammed, A.A., Minhas, R., Wu, Q.J., Sid-Ahmed, M.A.: Human face recognition based on multidimensional PCA and extreme learning machine. *Pattern Recognit.* **44**(10), 2588–2597 (2011)
16. Gao, X., Chen, Z., Tang, S., Zhang, Y., Li, J.: Adaptive weighted imbalance learning with application to abnormal activity recognition. *Neurocomputing* **173**, 1927–1935 (2016)
17. Mirza, B., Lin, Z., Liu, N.: Ensemble of subset online sequential extreme learning machine for class imbalance and concept drift. *Neurocomputing* **149**, 316–329 (2015)
18. Xia, S.X., Meng, F.R., Liu, B., Zhou, Y.: A kernel clustering-based possibilistic fuzzy extreme learning machine for class imbalance learning. *Cognit. Comput.* **7**(1), 74–85 (2015)
19. Zong, W., Huang, G.B., Chen, Y.: Weighted extreme learning machine for imbalance learning. *Neurocomputing* **101**, 229–242 (2013)
20. Liu, N., Wang, H.: Ensemble based extreme learning machine. *IEEE Signal Process. Lett.* **17**(8), 754–757 (2010)
21. Cao, J., Lin, Z., Huang, G.B., Liu, N.: Voting based extreme learning machine. *Inf. Sci.* **185**(1), 66–77 (2012)
22. Sharma, R., Bist, A.S.: Genetic algorithm based weighted extreme learning machine for binary imbalance learning. In: 2015 International Conference on Cognitive Computing and Information Processing (CCIP), pp. 1–6. IEEE (2015)
23. Dudek, G.: Extreme learning machine for function approximation-interval problem of input weights and biases. In: 2015 IEEE 2nd International Conference on Cybernetics (CYBCONF), pp. 62–67. IEEE (2015)
24. Zhang, N., Qu, Y., Deng, A.: Evolutionary extreme learning machine based weighted nearest-neighbor equality classification. In: 2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), vol. 2, pp. 274–279. IEEE (2015)
25. Hu, K., Zhou, Z., Weng, L., Liu, J., Wang, L., Su, Y., Yang, Y.: An optimization strategy for weighted extreme learning machine based on PSO. *Int. J. Pattern Recognit. Artif. Intell.* **31**(01), 1751001 (2017)
26. Zhu, Q.Y., Qin, A.K., Suganthan, P.N., Huang, G.B.: Evolutionary extreme learning machine. *Pattern Recognit.* **38**(10), 1759–1763 (2005)
27. Batista, G.E., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. *ACM Sigkdd Explor. Newsl.* **6**(1), 20–29 (2004)
28. Estabrooks, A., Jo, T., Japkowicz, N.: A multiple resampling method for learning from imbalanced data sets. *Comput. Intell.* **20**(1), 18–36 (2004)
29. Mazurowski, M.A., Habas, P.A., Zurada, J.M., Lo, J.Y., Baker, J.A., Tourassi, G.D.: Training neural network classifiers for medical decision making: the effects of imbalanced datasets on classification performance. *Neural Netw.* **21**(2), 427–436 (2008)
30. Seiffert, C., Khoshgoftaar, T.M., Van Hulse, J., Napolitano, A.: RUSBoost: a hybrid approach to alleviating class imbalance. *IEEE Trans. Syst. Man Cybern. Part A* **40**(1), 185–197 (2010)
31. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory undersampling for class-imbalance learning. *IEEE Trans. Syst. Man Cybern. Part B (Cybernetics)* **39**(2), 539–550 (2009)
32. Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., Herrera, F.: A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **42**(4), 463–484 (2012)
33. Barandela, R., Valdovinos, R.M., Sánchez, J.S.: New applications of ensembles of classifiers. *Pattern Anal. Appl.* **6**(3), 245–256 (2003)
34. Walsh, I., Pollastri, G., Tosatto, S.C.: Correct machine learning on protein sequences: a peer-reviewing perspective. *Briefings Bioinf.* **17**(5), 831–840 (2015)
35. Arunkumar, N., Ram Kumar, K., Venkataraman, V.: Automatic detection of epileptic seizures using permutation entropy, Tsallis entropy and Kolmogorov complexity. *J. Med. Imaging Health Inf.* **6**(2), 526–531 (2016)
36. Basturk, B., Karaboga, D.: An artificial bee colony (ABC) algorithm for numeric function optimization. *IEEE Swarm Intell. Symp.* **8**(1), 687–697 (2006)
37. Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N.: A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artif. Intell. Rev.* **42**(1), 21–57 (2014)
38. Jadon, S.S., Bansal, J.C., Tiwari, R., Sharma, H.: Accelerating artificial bee colony algorithm with adaptive local search. *Memet. Comput.* **7**(3), 215–230 (2015)
39. Ozturk, C., Karaboga, D.: Hybrid artificial bee colony algorithm for neural network training. In: 2011 IEEE Congress on Evolutionary Computation (CEC), pp. 84–88. IEEE (2011) [34] Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces
40. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **11**(4), 341–359 (1997)
41. Stephygraph, L.R., Arunkumar, N.: Brain-actuated wireless mobile robot control through an adaptive human-machine interface. *Adv. Intell. Syst. Comput.* **397**, 537–549 (2016)
42. Fernandes, S.L., Gurupur, V.P., Sunder, N.R., Arunkumar, N., Kadry, S.: A novel nonintrusive decision support approach for heart rate measurement. *Pattern Recognit. Lett.* <https://doi.org/10.1016/j.patrec.2017.07.002> (2017)
43. Yu, T., Jan, T., Simoff, S., Debenham, J.: A hierarchical VQSVM for imbalanced data sets. In: 2007 International Joint Conference on Neural Networks, pp. 518–523. IEEE (2007)
44. Zhang, Z., Ou, J., Li, D., Zhang, S.: Optimization design of coupling beam metal damper in shear wall structures. *Appl. Sci.* **7**(2), 137 (2017)
45. Pan, W., Chen, S., Feng, Z.: Automatic clustering of social tag using community detection. *Appl. Math. Inf. Sci.* **7**(2), 675–681 (2013)
46. Liu, C., Li, Y., Zhang, Y., Yang, C., Hongbin, W., Qin, J., Cao, Y.: Solution-processed, undoped, deep-blue organic light-emitting diodes based on starburst oligofluorenes with a planar triphenylamine core. *Chem. A Eur. J.* **18**(22), 6928–6934 (2012)
47. Pacifico, L.D., Ludermit, T.B.: Evolutionary extreme learning machine based on particle swarm optimization and clustering strategies. In: The 2013 International Joint Conference on Neural Networks (IJCNN), pp. 1–6. IEEE (2013)
48. López, V., Fernández, A., García, S., Palade, V., Herrera, F.: An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics. *Inf. Sci.* **250**, 113–141 (2013)
49. Bhowan, U., Johnston, M., Zhang, M.: Developing new fitness functions in genetic programming for classification with unbalanced data. *IEEE Trans. Syst. Man Cybern. Part B* **42**(2), 406–421 (2012)
50. Bartlett, P.L.: The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE Trans. Inf. Theory* **44**(2), 525–536 (1998)
51. Barandela, R., Sánchez, J.S., García, V., Rangel, E.: Strategies for learning in class imbalance problems. *Pattern Recognit.* **36**(3), 849–851 (2003)
52. Shao, Y.H., Chen, W.J., Zhang, J.J., Wang, Z., Deng, N.Y.: An efficient weighted Lagrangian twin support vector machine for

imbalanced data classification. *Pattern Recognit.* **47**(9), 3158–3167 (2014)

53. Frank, A., Asuncion, A.: UCI Machine Learning Repository <http://archive.ics.uci.edu/ml>. School of Information and Computer Science, p. 213. University of California, Irvine, CA (2010)
54. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **7**, 1–30 (2006)
55. Sun, Z., Song, Q., Zhu, X., Sun, H., Xu, B., Zhou, Y.: A novel ensemble method for classifying imbalanced data. *Pattern Recognit.* **48**(5), 1623–1637 (2015)



Xiaofen Tang received her M.S. degree in computer science from Ningxia University in Ningxia, China. She is currently an associate professor in Ningxia University. Her research interests is mainly in the areas of neural network and bioinformatics. She published several research papers in scholarly journals in the above research areas.



Li Chen received her Ph.D. degree in computer science from XiDian University in Xi'an, China. She is currently a Professor in Northwest University. Her research interests include intelligent information processing, data mining and pattern recognition. She published more than hundred papers in journals and conferences and holds several patents.