Prog Artif Intell
https://doi.org/10.1007/s13748-017-0136-2

REGULAR PAPER

A self-adaptive evolutionary weighted extreme learning machine for binary imbalance learning

Xiaofen Tang[1,2] · **Li Chen**[1]

Received: 5 May 2017 / Accepted: 3 October 2017

Abstract It is known that the problem of imbalanced data sets widely exists in various application fields. The weighted extreme learning machine (WELM) was proposed. It solved the L_2-regularized weighted least squares problem in order to avoid the generation of an over-fitting model and to obtain better classification performances in imbalanced data sets when compared with extreme learning machine. While a WELM algorithm can address class imbalance issues, the random assignment of input parameters and the training sample weights generated according to class distribution of training data have been found to affect the performance of WELM. The aim of this study was to propose a self-adaptive differential evolutionary weighted extreme learning machine (SDE-WELM) which utilized a self-adaptive differential evolutionary to find the optimal input weights, hidden node parameters, and training sample weights of the WELM and exploit an appropriate criterion to be used as the fitness function for binary imbalance learning. The experimental results of the majority of the 40 data sets examined in this study indicated that the proposed method had the ability to achieve a better classification performance when compared with a weighted extreme learning machine (WELM), ensemble weighted extreme learning machine, evolutionary weighted extreme learning machine, and an artificial bee colony optimization-based weighted extreme learning machine and the four popular ensemble methods which combine data sampling and the Bagging or Boosting used in support vector machine as base classifier.

✉ Li Chen
chenli@nwu.edu.cn

Xiaofen Tang
xiaofentang@163.com

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

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

Keywords Weighted extreme learning machine · Imbalanced data classification · Single-hidden-layer feed-forward networks · The self-adaptive differential evolutionary algorithm

1 Introduction

Extreme learning machine (ELM) [1–4], is computationally powerful single-hidden-layer feed-forward neural network, which has been successfully and widely used in the classification of a variety of real-world problems, such as biofuel engine performance prediction [5], face recognition [6,7], speaker recognition [8], abnormal activity recognition [9], medical diagnoses [10], security assessment [11], bioinformatics [12], and vehicle classification applications [13]. Although standard ELM displays better generalization performance compared with many other machine learning methods, ELM is not well suited for imbalance dataset classification in its basic form [14]. This is due to the fact that the ELM has a natural tendency to favor the majority class by assuming balanced class distribution, or equal misclassification cost, when classifying data with complex class distribution [15]. Problems of imbalanced data sets exist in various areas, such as specific proteins detection [16], identification of genes [17] and drug discovery problem [18], medical diagnosis [19].

The majority of the standard learning algorithms usually fail due to their accuracy-oriented, which causes a bias toward the majority classes, and results in a lower sensitivity in the detection of minority classes. In fact, in some scenarios, the classification performance of the minority class can be more

Published online: 16 January 2018

important than that of the majority ones since the former often represents the main class of interest. As stated in the related references [20] and [21], the existing algorithms which deal with class imbalance problems can be distinguished into two groups as follows:

1. The use of ensemble approaches [20] which combine random sampling with bagging containing different classifiers have been applied to subsets of data. Also, combining random sampling with Boosting, which gives more attention to the samples misclassified in the previous iteration and adjusts the weights toward their correct classification, has been examined. For example, RUSboost [22], EsayEnsemble [23], OverBagging [20], and UnderBagging [24] are popular ensemble methods incorporated with the data sampling.
2. The modifications of standard classification algorithms to bias learning performance toward the minority classes have also been utilized [21,25,26].

In order to deal with the classification problems of imbalanced data sets, several improved variants of ELM have been proposed. Toh [27] and Deng et al. [28] proposed weighted regularized extreme learning machine (WRELM). However, their works did not target imbalance learning. Fuzzy extreme learning machine (FELM) assigns different misclassification cost to positive samples and negative samples according to the results of kernel-based possibilistic fuzzy c-Means clustering to handle the class imbalance problem in the presence of noise and outliers [29]. Weighted extreme learning machines (WELMs) assigns fixed weights for the samples. The fixed weights are proportional to their respective class size, in order to reflect their importance, and solve the L^2-regularized weighted least squares problem, resulting in better generalization performances [14]. Due to random assignment of input parameters, the WELM obtains extremely fast learning speed. However, the non-analytical determination of hidden layer parameters results in the WELM to having large impacts on classification performance [30,31]. In addition, the training sample weights generated according to the class distribution of training data have been observed to affect the performance of the WELM [32]. Therefore, a method for setting better training sample weights remains a problem worth addressing [33]. It has been determined that some meta-heuristic algorithms have the ability to optimize the weights of neural network [34,35]. Also, self-adaptive differential evolution algorithm (SDE) is a promising global optimization technique which has the ability to avoid local minima and has been successfully applied in the evolution of the synaptic weights, and also in the field of architecture [36–38].

This research study focused on determining the self-adaptive input weights and hidden layer parameters of WELM network, along with the setting of better and more effective training sample weights. This study presented a self-adaptive differential evolutionary weighted extreme learning machine (SDE-WELM) by combining a WELM with a self-adaptive differential evolutionary in order to improve the performance of the WELM. The SDE-WELM evolved classifiers in an imbalanced environment without relying on accuracy and focused on decreasing the norm of the output weights of the SLFN. In addition, when traditional evaluating criteria, such as the overall success or error rate, could potentially be influenced by a larger number of examples from the majority class, then the appropriate criterion was found to be crucial to assessing the performance of the imbalanced classification modeling, as well as the production of a non-biased solution in the evolutionary algorithms. In the empirical study, we first study the effectiveness of the fitness function and the variance of norm of output weights. In this study's empirical examination, the effectiveness of the fitness function and the variance of the norm of the output weights were first assessed. Then this study's method was compared with the WELM, ensemble WELM (EN-WELM) [30], and two methods which combined WELM with meta-heuristic algorithms for improving the performance of the WELM. These included a differential evolutionary weighted extreme learning machine (DE-WELM), which used a differential evolutionary algorithm to determine the optimal input weights, hidden node parameters, training sample weights of the WELM, and an artificial bee colony weighted extreme learning machine (ABC-WELM), which utilized an artificial bee colony algorithm to determine the optimal input weights, hidden node parameters, and training sample weights of the WELM. In addition, this study compared the proposed method with a weighted support vector machine (WSVM) and four popular ensemble learning algorithms which combined data sampling and the Bagging or Boosting and has shown the good performance [20]. These included the RUSboost, EsayEnsemble, OverBagging, and UnderBagging algorithms. This study considered an SVM as base classifier for above ensemble learning algorithms since it displayed the ability to provide relatively robust classification results when applied to imbalanced data sets [24,39]. Also, a nonparametric statistical comparison of the results was conducted, as suggested in the related literature [40,41].

The structure of this research study was as follows: First of all, the weighted extreme learning machine and the self-adaptive differential evolutionary algorithm are reviewed in Sect. 2. A detailed description of the proposed SDE-WELM algorithm is presented in Sect. 3. Next, a quantitative performance comparison of the proposed SDE-WELM with the WELM, DE-WELM, EN-WELM, ABC-WELM, WSVM, RUSboost, EsayEnsemble, OverBagging, and UnderBagging is detailed in Sect. 4. Finally, this study's summarization and conclusions are presented.

2 Preliminaries

This study presented a brief review of weighted the extreme learning machine (WELM) and the self-adaptive differential evolutionary algorithm (SDE) in this section.

2.1 Weighted regularized extreme learning machine

Given the imbalanced training set (x_i, t_i), where $x_i = \left[x_{i,1}, x_{i,2}, \ldots, x_{i,n}\right]^{\mathrm{T}} \in R^n$ and $t_i = \left[t_{i,1}, t_{i,2}, \ldots, t_{i,m}\right]^{\mathrm{T}} \in R^m$. $i = 1, 2, \ldots, N$, where N represents the number of samples, n represents the number of features, and m represents the number of classes. The weighted regularized extreme learning machine (WELM) assigns fixed weights for the samples proportional to their respective class sizes, in order to reflect their importance in the related literature [14]. The mathematical model of the weighted extreme learning machine with L hidden neurons, and generalized nonlinear activation function $g(x)$, could be expressed as:

$$\text{Minimize: } \frac{1}{2}\|\beta\|^2 + \frac{\gamma}{2} W \sum_{i=1}^{N} \|\varepsilon_i\|^2$$
$$\text{Subject to: } \beta_j g\left(a_j x_i + b_j\right) - t_i = \varepsilon_i, \quad j = 1, 2, \ldots, L \tag{1}$$

where $a_j = \left[a_{j1}, a_{j2}, \ldots, a_{jm}\right]$ is the weight vector between the jth hidden neuron and the input neurons; $\beta_j = \left[\beta_{j1}, \beta_{j2}, \ldots, \beta_{jc}\right]$ is the weight vector between the jth hidden layer and the output neurons; b_j is threshold of the ith hidden neuron; $\varepsilon_i = \left[\varepsilon_{i,1}, \ldots, \varepsilon_{i,m}\right]$ is the training error vector of the m output neurons with respect to the training sample x_i; γ is the ridge parameter which represents the trade-off between the minimization of the training errors and the maximization of the marginal distance. $W_{ii} = \frac{1}{\#(t_i)}$ 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 the extra weight assigned to each sample to strengthen the impact of the minority classes while weakening the relative impact of the majority classes, and $\#(t_i)$ is the number of samples belonging to class c, $c = 1, 2, \ldots, m$.

By substituting the constraints to the objective function yields, the Lagrangian for (1) can be written as follows:

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

where $\alpha_i \in R\ (i = 1, 2, \ldots, N)$ represents the Lagrangian multiplier with the equality constraints of (1). Furthermore, by making the partial derivatives with respect to the variables $(\beta, \varepsilon, \alpha)$ all equal to zero, the KKT corresponding optimality conditions can be obtained as follows:

$$\frac{\partial L}{\partial \beta} = 0 \rightarrow \beta = H^{\mathrm{T}} \alpha \tag{3}$$
$$\frac{\partial L}{\partial \varepsilon_i} = 0 \rightarrow \alpha_i = \gamma W \varepsilon_i \tag{4}$$
$$\frac{\partial L}{\partial \alpha_i} = 0 \rightarrow h(x_i)\beta - t_i + \varepsilon_i = 0, \quad i = 1, 2, \ldots, N \tag{5}$$

Then, by substituting (3) and (4) into (5), the closed-form solution to β can be obtained as follows:

$$\beta \begin{cases} = \left(\frac{I}{\gamma} + H^{\mathrm{T}} W H\right)^{-1} H^{\mathrm{T}} W T, & N \geq L \\ = H^{\mathrm{T}} \left(\frac{1}{\gamma} + W H H^{\mathrm{T}}\right) W T, & N < L \end{cases} \tag{6}$$

where I is the identity matrix; L is the number of hidden nodes; and N represents the number of input samples. For the binary classification problems, the WELM required only one output node, and the decision function was as follows:

$$\begin{aligned} f(x) &= \text{sign}\,(h(x)\beta) \\ &= \text{sign}\left(\begin{cases} h(x)\left(\frac{I}{\gamma} + H^{\mathrm{T}} W H\right)^{-1} H^{\mathrm{T}} W T \\ h(x) H^{\mathrm{T}} \left(1/\gamma + W H H^{\mathrm{T}}\right) W T \end{cases} \right) \end{aligned} \tag{7}$$

2.2 The self-adaptive differential evolutionary algorithm

Differential evolution (DE) algorithm was proposed by Storn and Price for numerical optimization problems [42] and was found to be an efficient global optimizer in continuous search domains. This algorithm has been successfully applied in the training of neural networks [43,44]. The DE initializes the populations which represent different solutions for an optimization problem, as well as search for the solutions of the optimization problems. Then, three operations can be performed, namely mutation, crossover, and selection operations. If D represents the dimension of the optimization parameters, then each individual x_i ($i = 1, 2, \ldots, \text{NP}$) is a D-dimensional vector among the SN individuals and is generated using a uniform distribution as follows:

$$x_{i,0}^{j} = x_{\min}^{j} + \text{rand}\ (0, 1) \cdot \left(x_{\max}^{j} - x_{\min}^{j}\right), \quad j = 1, 2, \ldots, D \tag{8}$$

where $x_{\min}^{j}$ and $x_{\max}^{j}$ are the bounds of x_i in jth dimension and rand $(0, 1)$ is a uniformly distributed random number in the range $[0, 1]$. A differential evolution algorithm employs mutation operations to produce a mutant vector $v_{i,G}$ for each individual vector $x_{i,G}$ at the Gth generation.

2.2.1 Mutation operation

With respect to each individual vector $x_{i,G}$ at generation G, the self-adaptive differential evolution algorithm perturbs an existing vector in order to generate a new mutant vector $v_{i,G}$ via a certain mutation strategy [42]. Several frequently used mutation strategies were listed as follows:

Strategy 1: DE/rand/1

$$v_{i,G} = x_{r_1^i,G} + F \cdot \left(x_{r_2^i,G} - x_{r_3^i,G}\right) \tag{9}$$

Strategy 2: DE/rand-to-best/2

$$\begin{aligned} v_{i,G} = x_{r_1^i,G} &+ F \cdot \left(x_{\text{best},G} - x_{r_1^i,G}\right) \\ &+ F \cdot \left(x_{r_2^i,G} - x_{r_3^i,G}\right) \\ &+ F \cdot \left(x_{r_4^i,G} - x_{r_5^i,G}\right) \end{aligned} \tag{10}$$

Strategy 3: DE/rand/2

$$\begin{aligned} v_{i,G} = x_{r_1^i,G} &+ F \cdot \left(x_{r_2^i,G} - x_{r_3^i,G}\right) \\ &+ F \cdot \left(x_{r_4^i,G} - x_{r_5^i,G}\right) \end{aligned} \tag{11}$$

Strategy 4: DE/current-to-rand/1

$$\begin{aligned} v_{i,G} = x_{i,G} &+ F \cdot \left(x_{r_1^i,G} - x_{i,G}\right) \\ &+ F \cdot \left(x_{r_2^i,G} - x_{r_3^i,G}\right) \end{aligned} \tag{12}$$

where the indices r_1, r_2, r_3, r_4, r_5 represent the random and mutually different integers generated in the range [1, NP], which should also be different from the index i; F is a factor in [0, 2] for the scaling of the differential vectors; $x_{\text{best},G}$ represents the individual vector with best fitness value in the population at generation G. Following the mutation phase, a crossover procedure was used to increase the diversities of the perturbed parameter vector.

2.2.2 Crossover operation

For each mutant vector $v_{i,G}$ and target vector $x_{i,G}$ at generation G, a trivial vector $u_{i,G}$ is created according to the following crossover equation:

$$u_{i,G}^j = \begin{cases} v_{i,G}^j & \text{if } \left(\text{rand}_j\,[0,1] \le \text{CR}\right) \text{ or } \left(j = j_{\text{rand}_j}\right) \\ x_{i,G}^j, & \text{Otherwise} \end{cases} \tag{13}$$

where CR is the crossover rate in [0, 1), which is the a user-specified constant utilized to control the fraction of parameter values copied from the mutant vector $v_{i,G}$; and rand_j is a randomly chosen integer in the range $[1, D]$.

2.2.3 Selection operation

In order to maintain the population size over subsequent generations, the algorithm determined whether the trial vector $u_{i,G}$ or the individual vector $x_{i,G}$ survived to the next generation ($G + 1$ generation). Therefore, the fitness function value of each trial vector $f\left(u_{i,G}\right)$ was compared to that of its corresponding individual vector $f\left(x_{i,G}\right)$ in the current population. Then, the one with better fitness was kept as the population of the next generation. The selection operation could then be expressed as follows:

$$x_{i,G+1} = \begin{cases} u_{i,G}, & \text{if } f\left(u_{i,G}\right) \le f\left(x_{i,G}\right) \\ x_{i,G}, & \text{otherwise} \end{cases} \tag{14}$$

However, the performance of conventional differential evolution algorithm was found to be dependent on appropriate selection of the trial vector generation strategies, and their associated control parameter values, using trial-and-error procedure, which resulted in the expensive computational costs for the DE. The trial vector generation strategies, and their associated control parameter values of the self-adaptive differential evolution (SDE), were found to be gradually self-adaptively as they learned from their previous experiences in generating promising solutions. Similar to the DE, the SDE initialized a population representing different solutions for an optimization problem, as well as searching for solutions for the optimization problem [45]. Then, three operations, namely mutation, crossover, and selection operations, were repeated until the maximum iteration times were exceeded MaxGen, or the goal was met. Following the mutation phase, a trial vector generation strategy of the SDE was chosen from a candidate pool constructed by four strategies shown as Formula (15–18) according to the probability $p_{l,G}$ learned from its previous experience in generating promising solutions, where $p_{l,G}$ represented the probability that the strategy l ($l = 1, 2, 3, 4$) should be chosen at every generation.

DE/rand/1/bin:

$$u_{i,j} = \begin{cases} x_{r_1,j} + F \cdot \left(x_{r_2,j} - x_{r_3,j}\right), & \text{if rand}\,[0,1) < \text{CR} \text{ or } j = j_{\text{rand}} \\ x_{i,j}, & \text{otherwise} \end{cases}; \tag{15}$$

DE/rand-to-best/2/bin:

$$u_{i,j} = \begin{cases} x_{i,j} + F \cdot \left(x_{\text{best},j} - x_{i,j}\right) + F \cdot \left(x_{r_1,j} - x_{r_2,j}\right) + F \cdot \left(x_{r_3,j} - x_{r_4,j}\right), & \text{if rand}\,[0,1) < \text{CR or } j = j_{rand} \\ x_{i,j}, & \text{otherwise} \end{cases}; \tag{16}$$

DE/rand/2/bin:

$$u_{i,j} = \begin{cases} x_{r_1,j} + F \cdot \left(x_{r_2,j} - x_{r_3,j}\right) + F \cdot \left(x_{r_4,j} - x_{r_5,j}\right), & \text{if rand}\,[0,1) < \text{CR or } j = j_{\text{rand}} \\ x_{i,j}, & \text{otherwise} \end{cases}; \tag{17}$$

DE/current-to-rand/1:

$$U_{i,G} = X_{i,G} + K \cdot \left(F \cdot \left(X_{r_1,G} - X_{i,G}\right)\right) + F \cdot \left(X_{r_2,G} - X_{r_3,G}\right). \tag{18}$$

These probabilities were gradually adapted during the evolution in the following manner [45]: It was assumed that the probability of applying the lth strategy in the candidate pool to a target vector in the current population is p_l, $l = 1, 2, 3, 4$ (the probabilities with respect to each strategy were initialized as $1/4$). Then, following the evaluations of all the generated trial vectors, the number of trial vectors generated by the lth strategy which had the ability to successfully enter into the next generation and that are discarded in the next generation were recorded as $ns_{l,G}$ and $nf_{l,G}$, respectively. The learning period (LP) stored the number of successful and failure memories within a fixed number of the previous generations. The small constant value $\varepsilon = 0.01$ was used to avoid possible null success rates.

$$p_{l,G} = \frac{s_{l,G}}{\sum_{l=1}^{4} s_{l,G}} \tag{19}$$

where

$$s_{l,G} = \frac{\sum_{g=G-\text{LP}}^{G-1} ns_{l,G}}{\sum_{g=G-\text{LP}}^{G-1} ns_{l,G} + \sum_{g=G-\text{LP}}^{G-1} nf_{l,G}} + \varepsilon,$$
$$(l = 1, 2, 3, 4G > \text{LP})$$

3 Evolutionary extreme learning machine for binary imbalance learning

In previous studies, in order to achieve a more satisfactory classification performance when applying the WELM for a given data set, it was common to perform trial-and-error searches and fine-tune its associated control parameter values (for example, the values of the regularization parameter γ and the number of hidden nodes L). However, the classification performance of the WELM have been found to be unstable. Natural evolution and stochastic search are usually used to optimize the parameter of single-layer feed-forward neural networks [43,46,47]. The two representative methods which utilize evolutionary computation to optimize the ELM's parameter are evolutionary extreme learning machine (DE-ELM) [36] and self-adaptively evolutionary extreme learning machine(SDE-ELM) [37], respectively. The DE-ELM adopts differential evolution method to select the hidden node parameters and the ELM to calculate the output weights. However, different strategies and parameters have be manually adapted in order to achieve the best performance during the different stages of evolution, and those which have required high computational costs. It was found that the SDE-ELM could self-adaptively determine the suitable control parameters and generation strategies. However, it was determined that, similar to the existing neural network learning optimization processes, the DE-ELM and SDE-ELM rely on error to weigh the classifiers which results in finding the weights configuration associated with the minimum output error. Therefore, for balanced applications, these approaches have worked very well. However, when faced with imbalanced data, errors can tend to favor the evolution of solutions biased toward the majority classes. In fact, the classification performance of the minority class accurately can be more important than those of the majority classes accurately in some scenarios since the minority one often represent the main classes of interest. Moreover, while a WELM algorithm can address class imbalance issues, the random assignment of input parameters and the training sample weights generated according to class distribution of training data have been found to affect the performance of WELM. Therefore, motivated by these observations, this study developed a self-adaptively evolutionary weighted extreme learning machine (SDE-WELM) algorithm, in which the network hidden node parameters and training sample weights could be optimized by a self-adaptive differential evolutionary algorithm. Also, the classifiers could be evolved in an imbalanced environment, without relying on error. The core idea behind the proposed SDE-WELM algorithm was elucidated in this study as follows.

The SDE-WELM started out with a large population of network parameters and training sample weights. All the network input parameters and training sample weights W_{ii} were directly encoded into the individual vector of the population and then evolved together with the encoded solutions by self-adaptive rules which utilized the feedback from the fitness values to guide the updating of the parameters. The output weights of the network with respect to each population vector were computed according to Eq. (6). The parameter values involved in the individuals with better fitness values survived. The population of the first generation was composed of a set of NP vectors, where each one included all of the network input weights, hidden biases, and training sample weights as follows:

$$x_{k,G} = \left[a^T_{1,(k,G)}, a^T_{2,(k,G)}, \ldots, a^T_{L,(k,G)}, b_{1,(k,G)}, b_{2,(k,G)}, \ldots, b_{L,(k,G)}, w_{11}, w_{22}, w_{33}, \ldots, w_{NN}\right] \tag{20}$$

where a_j and b_j $(j = 1, \ldots, L)$ present the parameters randomly generated within the range of $[-1, 1]$ and $[0, 1]$ respectively; L represents the number of hidden neuron; G represents the number of generations and $k = 1, 2, \ldots, \text{NP}$. w_{ii} $(i = 1, \ldots, N)$ are the diagonal element of the weight matrix generated according to the class distribution of the training samples; N denotes the number of samples.

The SDE-WELM performed repeated mutation, crossover, and selection operations. During the evolution, with respect to each target vector in the current population, one strategy was chosen from the candidate pool according to a probability [shown as Formula (19)] which was learned from its previous experiences in generating promising solutions. At each generation, the strategy candidate pool which included several effective trial vector generation strategies [shown as formula (15–18)] was used. Consequently, both the trial vector generation strategies and their associated control parameter values could be gradually self-adapted according to their previous experiences of generating promising solutions. As previous stated in the related reference [42], the more successfully one strategy behaved in the previous generations when generating promising solutions, the higher the probability there was of it being chosen in the current generation to generate solutions. The input weights, hidden layer bias, and training samples weights of the WELM were optimized at each generation. Then, in order to evaluate and select the vector which survived to the next generation, it was required to select an appropriate evaluation metric as a fitness function. Another concern of this algorithm was the specific evaluation metric for each individual fitness value. In this study, the WELM was used as the fitness function and an evaluation metric was then selected as follow:

The evaluation metrics of imbalanced data sets classification performance remain an open problem. Some researchers have used different metrics as fitness functions. García and Herrera [48] used the area under the ROC curve (AUC), *G*-mean, AUC_EUS, and *G*-mean_EUS as fitness functions of evolutionary algorithms, in order to classify imbalanced data sets. The results of their study showed that the AUC_EUS and *G*-mean_EUS were appropriate evaluation metrics as fitness functions of evolutionary algorithms. Galar et al. proposed the enhancing ensembles for highly imbalanced data sets by evolutionary under-sampling (EUSboost) with *G*-mean_EUS [49]. Bhowan et al. used accuracy (ACC), AUC, and some new criterion as fitness functions of genetic programming, in order to evolve imbalanced classifiers. They empirically showed that the AUC evolved the classifiers with good performances in both the minority and majority classes [50]. In addition, some researchers have previously show that the AUC is a better metric by which to measure the performances of fitness functions when the class distribution is imbalanced [51,52] due to the symmetric nature of the distribution of the *G*-mean and F-measure over sensitivity and specificity. Therefore, this study examined the performance of the proposed SDE-WELM using the following four fitness functions and selected appropriate one.

$$\text{AUC} = \frac{1}{m \cdot n} \sum_{i=1}^{m} \sum_{j=1}^{n} I\left(x_i > y_j\right) \tag{21}$$

where $I(\cdot)$ is an indicator function satisfying I (true) $= 1$ and I (false) $= 0$. m, n refers to the number of positive examples and negative examples, respectively. x_i, y_j represent the outputs of the algorithm for the ith positive example and jth negative examples, respectively.

$$\text{G-mean} = \sqrt{\frac{\text{TP}}{\text{TP} + \text{FN}} \cdot \frac{\text{TN}}{\text{FP} + \text{TN}}} = \sqrt{\text{SN} \times \text{SP}} \tag{22}$$

where sensitivity (SN) $=$ TP/ (TP $+$ FN) is a true positive class accuracy. Specificity (SP) $=$ TN/(TN $+$ FP) is a true negative class accuracy.

$$\text{AUC\_EUS} = \begin{cases} \text{AUC} - \left|1 - \frac{n^+}{n^-}\right| \cdot P & \text{if } n^- > 0 \\ \text{AUC-P} & \text{if } n^- = 0 \end{cases} \tag{23}$$

$$G\text{-mean\_EUS} = \begin{cases} G\text{-mean} - \left|1 - \frac{n^+}{n^-}\right| \cdot P & \text{if } n^- > 0 \\ G\text{-mean} - P & \text{if } n^- = 0 \end{cases} \tag{24}$$

where n^+, n^- represent the number of original positive and negative samples within the training data set, respectively. P is the penalization factor.

With the fitness of all the individuals, the better individuals were selected based on the fitness of each individual. Based on the knowledge that networks end to have better generalization performance with small norm of output weights [36,53], the norm of the output weights, along with the fitness function on the validation set, was used as one more criterion for the trial vector selection. The present work mainly focused on decreasing the norm of the output weights of the SLFN, maximizing the fitness function, and constraining the input weights and hidden biases within a reasonable range to improve the convergence performance of the WELM. Therefore, the selection operation could be expressed as Eq. (25).

$$x_{k,G+1} = \begin{cases} u_{k,G+1}, & \text{if } \left(f\left(x_{k,G}\right) - f\left(u_{k,G}\right)\right) > \epsilon \cdot f\left(x_{k,G}\right), \\ u_{k,G+1}, & \text{if } \left|f\left(x_{k,G}\right) - f\left(u_{k,G}\right)\right| < \epsilon \cdot f\left(x_{k,G}\right) \\ & \text{ and } \|\beta_{u_{k,G+1}}\| < \|\beta_{x_{k,G}}\|, \\ x_{k,G}, & \text{otherwise.} \end{cases} \tag{25}$$

where $f\left(u_{k,G}\right)$ and $f\left(x_{k,G}\right)$ are the corresponding fitness values for trial vector $u_{k,G}$ and target vector $x_{k,G}$, respectively. $\|\beta_{x_{k,G}}\|$ and $\|\beta_{u_{k,G+1}}\|$ are the norm of the corresponding output weights obtained by the WELM when the input weights were set as the kth individual, and $\epsilon > 0$ is a tolerance rate.

Table 1 Specification of the imbalanced data sets to select the most appropriate evaluation metric

Data sets	#Ex	#Atts	Minority class	IR
Glass0	214	9	Building windows	2.06
Glass6	214	9	Headlamps	6.38
Ecoli3	336	7	imU	8.19
Wisconsin	683	9	Malignant	1.86
Pima	768	8	Class0	1.90
Pageblock1	5473	10	Horiz. line	15.6

Table 2 Test performance of the four different fitness function

Data sets	Algorithms	Fitness function	AUC	*G*-mean	SN	SP	γ	L
Glass0	WELM		83.1 ± 1.48	81.88 ± 1.68	90.39 ± 1.32	73.97 ± 2.33	2^2	70
	SDE-WELM	AUC	83.82 ± 0.98	$\mathbf{82.04} \pm 1.49$	$\mathbf{91.22 \pm 1.57}$	73.59 ± 0.10	2^2	50
		G-mean	83.78 ± 1.07	81.85 ± 1.65	$91.10 \pm .64$	73.55 ± 0.12	2^4	80
		Auc_EUS	83.58 ± 1.34	81.53 ± 1.62	91.29 ± 3.18	73.61 ± 0.28	2^2	80
		G-mean_EUS	83.66 ± 1.21	81.46 ± 2.03	90.82 ± 3.28	73.06 ± 0.34	2^2	80
Glass6	WELM		98.23 ± 0.76	98.21 ± 0.75	99.87 ± 1.51	$96.59 \pm 6.13\text{e}{-15}$	2^3	100
	SDE-WELM	AUC	99.95 ± 0.38	$\mathbf{99.12} \pm 0.30$	$\mathbf{100 \pm 0.64}$	97.85 ± 1.23	2^3	70
		G-mean	98.49 ± 0.37	98.89 ± 0.46	100 ± 0.82	97.40 ± 1.28	2^3	100
		AUC_EUS	97.51 ± 0.67	98.38 ± 1.18	98.84 ± 1.64	97.73 ± 1.05	2^3	100
		G-mean_EUS	97.81 ± 0.60	98.54 ± 1.18	99.00 ± 1.39	97.85 ± 1.59	2^3	100
Ecoli3	WELM		85.71 ± 3.07	90.25 ± 2.50	92.82 ± 5.37	87.25 ± 2.03	2^{-1}	20
	SDE-WELM	AUC	86.66 ± 2.27	91.24 ± 1.84	$\mathbf{95.89 \pm 4.52}$	88.29 ± 1.97	2^{-1}	20
		G-mean	86.66 ± 2.31	$\mathbf{91.28} \pm 1.95$	95.63 ± 4.33	87.62 ± 1.75	2^{-1}	20
		AUC_EUS	86.33 ± 2.47	91.02 ± 2.09	95.59 ± 4.46	86.47 ± 1.82	2^{-1}	20
		G-mean_EUS	86.88 ± 2.58	90.89 ± 1.78	96.03 ± 3.51	85.98 ± 2.07	2^{-1}	20
Wisconsin	WELM		97.81 ± 0.33	95.53 ± 0.85	93.54 ± 1.44	97.56 ± 0.46	2^6	170
	SDE-WELM	AUC	98.08 ± 0.15	$\mathbf{96.23 \pm 0.29}$	$\mathbf{94.50 \pm 0.56}$	97.76 ± 0.22	2^6	170
		G-mean	98.06 ± 0.31	96.04 ± 0.95	94.42 ± 0.38	97.71 ± 0.65	2^5	170
		AUC_EUS	98.11 ± 0.29	95.90 ± 0.69	94.22 ± 1.38	97.65 ± 0.38	2^5	160
		G-mean_EUS	98.08 ± 0.21	95.96 ± 0.53	94.30 ± 1.03	97.69 ± 0.33	2^5	180
Pima	WELM		74.70 ± 0.81	69.88 ± 1.16	57.10 ± 1.67	85.54 ± 1.01	2^3	140
	SDE-WELM	AUC	75.23 ± 0.66	$\mathbf{70.96} \pm 0.80$	60.28 ± 1.76	76.81 ± 1.88	2^3	130
		G-mean	75.09 ± 0.80	70.95 ± 1.00	$\mathbf{61.25 \pm 2.44}$	75.6 ± 2.83	2^2	130
		AUC_EUS	75.62 ± 0.58	71.00 ± 0.75	58.60 ± 1.81	79.10 ± 1.33	2^2	130
		G-mean_EUS	75.48 ± 0.70	71.07 ± 0.79	59.90 ± 2.23	77.59 ± 2.25	2^3	130
Abalone19	WELM		77.92 ± 2.90	63.38 ± 3.65	51.44 ± 6.27	$78..44 \pm 2.07$	2^4	60
	SDE-WELM	AUC	79.21 ± 3.47	$\mathbf{67.46} \pm 4.07$	$\mathbf{59.11 \pm 7.22}$	77.31 ± 1.42	2^4	60
		G-mean	78.22 ± 3.24	67.03 ± 3.82	55.56 ± 6.25	80.88 ± 2.03	2^3	60
		AUC_EUS	77.82 ± 2.45	66.70 ± 3.08	55.56 ± 6.94	80.08 ± 1.21	2^3	60
		G-mean _EUS	77.47 ± 2.97	66.41 ± 3.46	55.56 ± 6.04	79.38 ± 1.95	2^4	60

This process was repeated in each generation until the goal was reached or the maximum learning iterations were completed. The expectation was that the fitness of the population would be increased with each generation. Therefore, the WELM network with the optimal input weights, hidden biases, and the weights of training samples was obtained. Then the optimal WELM was applied to the testing data. The process of the proposed SDE-WELM is shown in the following:

Algorithm of the SDE-WELM:

Input: A training data set; number of hidden nodes L; number of generation Max_Gen.
Step 1: Set the generation counter $G = 0$, and randomly initialize a population of NP individuals $X_{k,G}, k = 1, 2, \ldots, \mathrm{NP}$. Then, initialize the crossover ratio CR, strategy probability $p_{l,G}$, and learning period (LP);
Step 2: Calculate the individuals' fitness values (AUC) and evaluate of the population. The output weights for each individual are preserved;
Step 3: While the stopping criterion is not satisfied, perform the following:

Step 3.1: Calculate the strategy probability $p_{l,G}$, and update the success and failure memory;
Step 3.2: Assign a trial vector generation strategy and parameters to each target vector
Step 3.3: Generate a new population where each trial vector is generated according to an associated trial vector generation strategy
Step 3.4: Select the individuals which have the highest fitness values from the existing population for the next generation.
Step 3.5: Increment the generation count G =G +1;

Step 4: END WHILE
Step 5: Output optimal solution

4 Experiments

In this section, the test performances of the four fitness functions in the SDE-WELM were discussed, and the most appropriate fitness function was selected. Then, the AUC and norm variance of output weights were observed during the training process of the SDE-WELM, and the completed comparison of the proposed method with the WELM, EN-WELM and two methods which combined WELM with meta-heuristic algorithms was presented and included a differential evolutionary weighted extreme learning machine (DE-WELM) and an artificial bee colony weighted extreme learning machine (ABC-WELM) for 15 unbalanced data sets. Then, in order to thoroughly evaluate the performance of the SDE-WELM, the details of the comparisons with the WELM, ensemble WELM (EN-WELM), DE-WELM, ABC-WELM, WSVM, RUSBoost, EsayEnsemble, OverBagging and UnderBagging for 40 unbalanced data sets were provided. All of the algorithms were implemented in MATLAB R2013b on a PC with an Intel Core i7 CPU 760 @ 2.80 GHz, and 8 G RAM.

4.1 Test performance of the four fitness function

In order to select the most appropriate evaluation metric as fitness function, this study used AUC, G-mean, AUC_EUS, and G-mean_EUS as fitness functions of the SDE-WELM algorithm and classified six imbalanced data sets, as shown

Table 3 Specification of the imbalanced data sets

Data sets	#Ex.	#Atts.	Minority class	IR	#Training	#Test	#Validating
Glass0	214	9	Building windows	2.06	111	58	55
Glass2	214	9	Vehicle windows	10.39	107	50	57
Glass6	214	9	Headlamps	6.38	103	54	57
Ionosphere	351	34	Bad	8.24	164	92	95
Ecoli0	336	7	cp	1.35	168	83	85
Ecoli1	336	7	im	3.36	166	81	89
Ecoli3	336	7	imU	8.19	170	86	80
Vehicle0	846	18	Van	3.25	448	179	219
Wisconsin	683	9	Malignant	1.86	328	170	185
Yeast1	1484	8	nuc	2.46	747	311	366
Yeast3	1484	8	ME3	8.11	741	370	373
Pima	768	8	Class0	1.90	379	189	200
Abalone9vs18	731	8	Class9	16.68	356	196	179
Abalone19	4174	8	Class19	128.87	2117	1013	1044
Pageblock1	5473	10	Horiz. line	15.6	2767	1337	1369

in Table 1. The characteristics of these data sets are detailed in Table 1. The value of penalization factor *P* in Eqs. (23), (24) is 0.2. The experiment was conducted using a fivefold cross-validation strategy. Since the SDE is a stochastic learning algorithm, each experiment was repeated 100 times and the average results of the different fitness functions on the test sets were recorded as detailed in Table 2. This table illustrates the reports of the best classification results in terms of

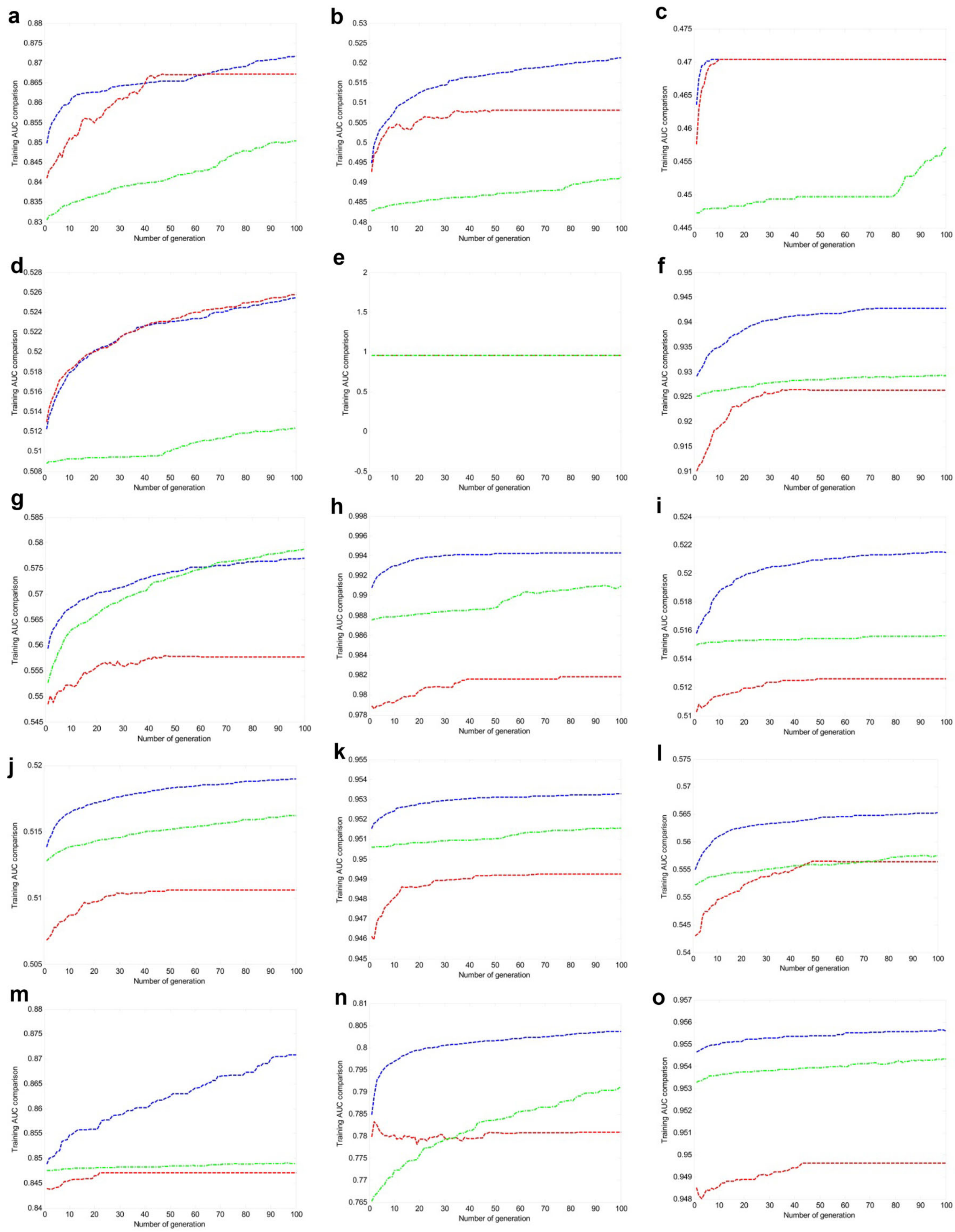

Fig. 1 Average AUC during the fittest evolved solution with the E-WELM (blue line), ABC-WELM (green line) and SDE-WELM (red line) over 100 independent runs. **a** Glass0, **b** Glass2, **c** Glass6, **d** Ionosphere, **e** Ecoli0, **f** Ecoli1, **g** Ecoli3, **h** Vehicle0, **i** Winsconsin, **j** Yeast1, **k** Yeast3, **l** Pima, **m** Abalone9vs18, **n** Abalone19, and **o** Pageblock1 (color figure online)

the average *G*-mean and corresponding AUC, SN, and SP, respective to the standard deviation.

The *G*-mean and SN values of the SDE-WELM over all data sets were found to be higher than that of the WELM (Table 2), which indicated that the SDE-WELM achieved a better classification performance for the minority classes, when compared with the WELM. The algorithm using AUC as fitness function achieved the highest SN over the five data

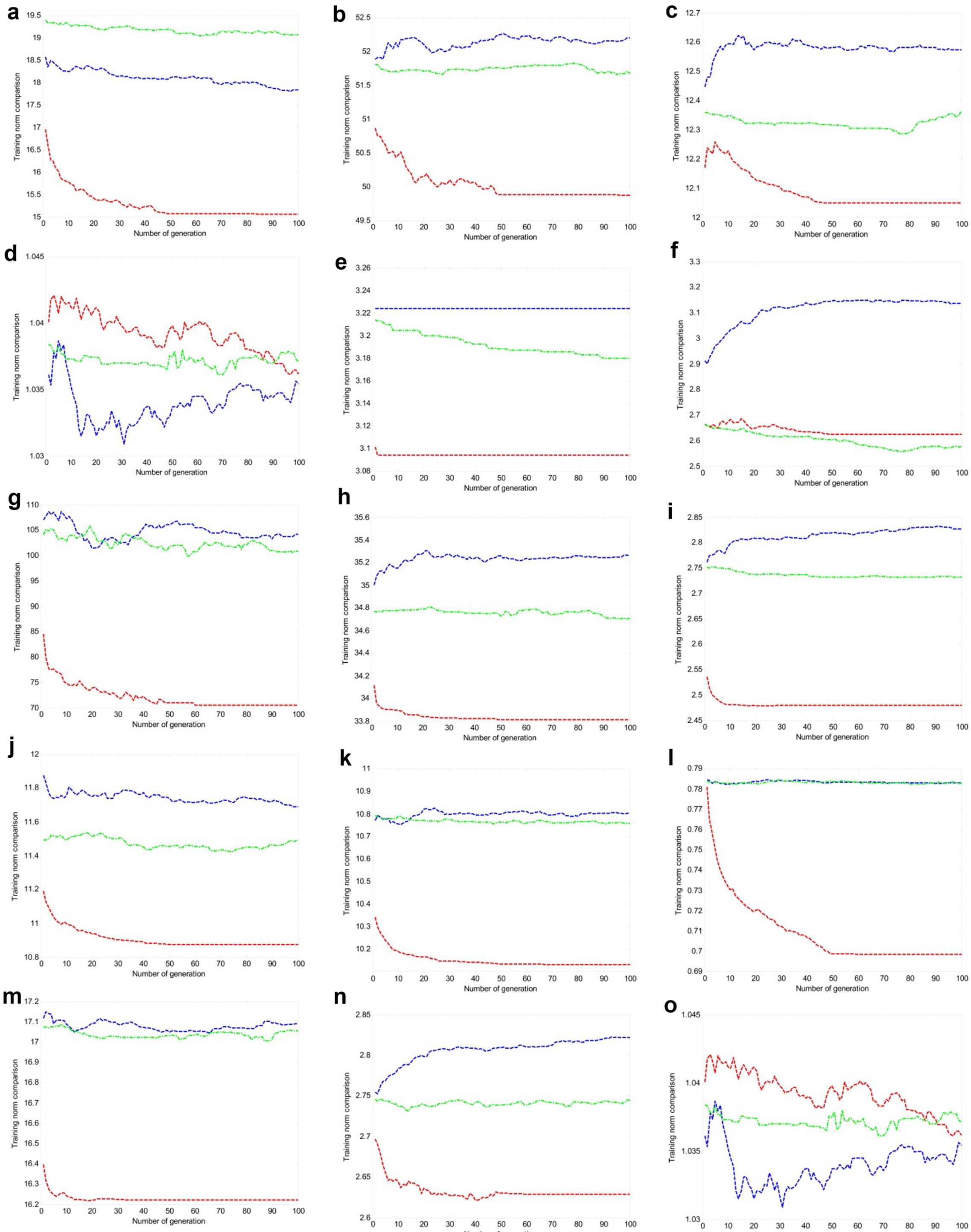

Fig. 2 Average norm of the output weights with the E-WELM (blue line), ABC-WELM (cyan line) and SDE-WELM (red line) over 100 independent runs. **a** Glass0, **b** Glass2, **c** Glass6, **d** Ionosphere, **e** Ecoli0, **f** Ecoli1, **g** Ecoli3, **h** Vehicle0, **i** Winsconsin, **j** Yeast1, **k** Yeast3, **l** Pima, **m** Abalone9vs18, **n** Abalone19, and **o** Pageblock1 (color figure online)

Table 4 Performances of the four algorithms on 15 imbalanced classification problems (one partition)

Data sets	Algorithms	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	85.15 ± 1.62	84.71 ± 1.57	93.56 ± 2.24	76.74 ± 2.17	2^3	270
	ABC-WELM	84.18 ± 1.45	82.82 ± 1.57	89.89 ± 2.41	76.47 ± 1.98	2^3	250
	SDE-WELM	85.97 ± 1.08	**85.76** ± 1.55	**94.52** ± 2.13	74.91 ± 0.24	2^3	260
Glass2	WELM	73.67 ± 1.44	79.61 ± 1.23	80.00 ± 1.562e−15	79.35 ± 1.93	2^4	180
	DE-WELM	73.44 ± 1.08	79.48 ± 1.01	79.83 ± 3.26	79.13 ± 1.99	2^4	180
	ABC-WELM	73.94 ± 1.62	79.81 ± 1.24	80.20 ± 2.86	79.26 ± 1.46	2^4	160
	SDE-WELM	74.19 ± 0.95	**80.36** ± 0.86	**80.61** ± 1.562e−15	79.94 ± 1.04	2^4	160
Glass6	WELM	98.23 ± 0.76	98.21 ± 0.75	99.87 ± 1.51	96.59 ± 6.103e−15	2^{-2}	280
	DE-WELM	98.98 ± 0.68	98.97 ± 0.69	**100** ± 0	97.96 ± 1.36	2^{-2}	260
	ABC-WELM	99.13 ± 0.68	98.92 ± 1.08	99.83 ± 1.67	98.02 ± 1.27	2^{-2}	250
	SDE-WELM	98.93 ± 0.58	**99.52** ± 0.59	**100** ± 0	97.85 ± 1.16	2^{-2}	240
Ionosphere	WELM	93.86 ± 1.64	88.32 ± 2.26	83.29 ± 3.62	93.72 ± 1.90	2^0	300
	DE-WELM	94.04 ± 1.32	88.56 ± 2.03	84.29 ± 3.14	93.08 ± 1.79	2^0	280
	ABC-WELM	94.22 ± 1.43	89.04 ± 2.24	84.61 ± 3.18	93.75 ± 1.33	2^0	260
	SDE-WELM	94.28 ± 1.26	**89.16** ± 1.53	84.32 ± 2.65	94.17 ± 2.16	2^0	260
Ecoli0	WELM	99.94 ± 6.3075e−4	97.19 ± 0.75	98.66 ± 1.51	95.74 ± 6.1093e−15	2^2	180
	DE-WELM	99.97 ± 5.7455e−4	97.42 ± 0.67	99.28 ± 1.35	95.74 ± 4.4633e−15	2^2	160
	ABC-WELM	99.97 ± 5.0402e−4	97.47 ± 0.62	99.22 ± 1.25	95.47 ± 4.4633e−16	2^2	180
	SDE-WELM	99.94 ± 4.9994e−4	**98.53** ± 0.52	**99.86** ± 1.17	95.74 ± 4.4633e−15	2^2	200
Ecoli1	WELM	88.74 ± 0.99	88.72 ± 0.96	90.42 ± 2.03	87.06 ± 0.23	2^1	200
	DE-WELM	88.80 ± 0.93	89.16 ± 1.93	92.00 ± 1.08	87.02 ± 0.41	2^1	230
	ABC-WELM	88.87 ± 0.93	89.45 ± 1.32	91.74 ± 1.66	87.05 ± 0.38	2^1	240
	SDE-WELM	89.04 ± 0.93	**89.81** ± 1.25	**92.37** ± 1.75	87.32 ± 0.16	2^1	240
Ecoli3	WELM	85.71 ± 3.07	90.25 ± 2.50	92.82 ± 5.37	87.25 ± 2.03	2^5	150
	DE-WELM	86.64 ± 2.57	91.03 ± 2.10	95.09 ± 4.91	87.22 ± 1.53	2^5	120
	ABC-WELM	86.84 ± 2.33	91.16 ± 1.93	95.73 ± 4.93	86.88 ± 1.71	2^5	120
	SDE-WELM	87.24 ± 2.36	**91.65** ± 1.91	**96.36** ± 4.48	87.29 ± 1.82	2^5	130
Vehicle0	WELM	98.08 ± 0.54	98.06 ± 0.40	99.95 ± 0.25	96.17 ± 0.74	2^4	360
	DE-WELM	98.12 ± 0.39	98.10 ± 0.40	99.95 ± 0.31	96.30 ± 0.71	2^4	360
	ABC-WELM	98.21 ± 0.48	98.18 ± 0.35	99.96 ± 0.30	96.23 ± 0.66	2^4	340
	SDE-WELM	98.48 ± 0.38	**98.96** ± 0.30	**100** ± 0	97.36 ± 0.75	2^4	320
Winsconsin	WELM	97.81 ± 0.33	95.53 ± 0.85	93.54 ± 1.44	97.56 ± 0.46	2^{-2}	170
	DE-WELM	97.77 ± 0.35	95.46 ± 0.85	93.48 ± 1.63	97.49 ± 0.56	2^{-2}	170
	ABC-WELM	97.80 ± 0.32	95.79 ± 0.83	94.02 ± 1.79	97.60 ± 0.45	2^{-2}	170
	SDE-WELM	98.15 ± 0.16	**96.39** ± 0.44	**94.66** ± 0.85	97.75 ± 0.24	2^{-2}	150
Yeast1	WELM	73.50 ± 0.44	75.02 ± 0.45	78.73 ± 0.63	71.49 ± 0.69	2^3	220
	DE-WELM	73.63 ± 0.37	75.19 ± 0.35	79.08 ± 0.54	71.76 ± 0.46	2^3	170
	ABC-WELM	73.69 ± 0.36	75.01 ± 0.34	78.79 ± 0.54	71.15 ± 0.56	2^3	180
	SDE-WELM	73.74 ± 0.42	**75.53** ± 0.30	**79.94** ± 0.54	71.60 ± 0.53	2^3	180
Yeast3	WELM	89.57 ± 0.74	89.47 ± 0.77	85.96 ± 1.48	93.18 ± 0.24	2^6	170
	DE-WELM	89.59 ± 0.65	89.52 ± 0.68	86.10 ± 1.27	93.08 ± 0.24	2^6	140
	ABC-WELM	89.78 ± 1.0	89.70 ± 0.62	86.44 ± 1.17	93.10 ± 0.24	2^6	120
	SDE-WELM	90.08 ± 0.24	**90.03** ± 0.25	**87.10** ± 0.44	93.06 ± 0.23	2^6	120

Table 4 continued

Data sets	Algorithms	AUC	*G*-mean	SN	SP	γ	*L*
Pima	WELM	74.70 ± 0.81	69.88 ± 1.16	57.10 ± 1.67	85.54 ± 1.01	2^0	280
	DE-WELM	74.73 ± 0.67	70.05 ± 0.98	57.55 ± 1.27	85.29 ± 0.90	2^0	290
	ABC-WELM	74.80 ± 0.63	70.12 ± 0.83	57.55 ± 1.31	84.84 ± 0.89	2^0	240
	SDE-WELM	75.40 ± 0.67	$\mathbf{71.00} \pm 0.75$	$\mathbf{60.30} \pm 1.46$	83.59 ± 1.69	2^0	240
Abalone9vs18	WELM	88.54 ± 2.70	88.54 ± 2.82	87.20 ± 5.33	90.06 ± 0.54	2^3	150
	DE-WELM	88.61 ± 4.27	88.77 ± 2.46	87.54 ± 4.69	90.09 ± 0.53	2^3	160
	ABC-WELM	88.46 ± 3.26	88.47 ± 2.68	86.70 ± 4.56	90.32 ± 0.56	2^3	160
	SDE-WELM	90.21 ± 1.51	$\mathbf{88.99} \pm 1.56$	$\mathbf{89.00} \pm 3.02$	90.22 ± 0.51	2^3	160
Abalone19	WELM	77.92 ± 2.90	63.38 ± 3.65	51.44 ± 6.27	78.44 ± 2.07	2^1	60
	DE-WELM	78.09 ± 2.87	64.26 ± 3.80	50.56 ± 5.99	81.96 ± 0.64	2^1	60
	ABC-WELM	78.25 ± 2.70	65.66 ± 3.34	53.89 ± 5.27	80.29 ± 1.31	2^1	60
	SDE-WELM	79.71 ± 3.47	$\mathbf{66.46} \pm 3.07$	$\mathbf{54.11} \pm 5.22$	77.31 ± 1.42	2^1	60
Pageblock1	WELM	93.64 ± 0.54	93.97 ± 0.31	91.86 ± 0.58	96.13 ± 0.18	2^5	170
	DE-WELM	94.02 ± 0.28	94.30 ± 0.28	92.52 ± 0.54	96.13 ± 0.14	2^5	170
	ABC-WELM	93.79 ± 0.36	94.12 ± 0.32	92.19 ± 0.59	96.07 ± 0.17	2^5	160
	SDE-WELM	93.96 ± 0.40	95.19 ± 0.25	94.22 ± 0.47	97.20 ± 0.19	2^5	180

sets, and also the best *G*-mean over the five data sets. Therefore, the AUC was determined to be the most appropriate evaluation metric of fitness function.

4.2 AUC and norm of the output weights during the training process

In order to evaluate this study's proposed SDE-WELM, a comparison of classification performances of the SDE-WELM with those of the WELM, EN-WELM and the two methods combining the WELM with meta-heuristic algorithms was completed. The two methods included a differential evolutionary weighted extreme learning machine (DE-WELM), and an artificial bee colony weighted extreme learning machine (ABC-WELM) for 15 unbalanced data sets [14,54,55]. The characteristics of the data sets are detailed in Table 3. The imbalance ratios (IR, defined as the number of negative class examples divided by the number of positive class examples) of the data sets were found to vary between from 1.35 to 128.87. All of the WELM-based approaches were implemented with a sigmoid activation function. For the DE-WELM and SDE-WELM, the *F* (step-size) and CR (crossover probability) were set as 1 and 0.8, respectively. The size of population was 50, and the maximum number of generation was 100. For the ABC-WELM, the SN (number of food sources) was 20, and the limit was 100, with the maximum number of cycles is set as 100. The performance of the proposed SDE-WELM was evaluated in detail using a comparison with the performances of the WELM, ABC-WELM, and DE-WELM. In order to eliminate the performance fluctuations caused by different data partitions and to observe the variances of the AUC and norm of the output weights during the training phase of the WELM-based approach, the training, testing, and validation data sets remained unchanged. Due to the fact that the differential evolution algorithm, self-adaptive differential evolution algorithm, and artificial bee colony algorithm are stochastic learning algorithms, each experiment was repeated for 30 runs, using a different random seed. Figures 1 and 2 detail the average AUC and norm of the fittest solution in the population over 100 generations. Table 4 displays the best results in terms of the *G*-mean and corresponding AUC, SN and SP, respective of the STD (the best results on testing set are shown in bold letters) of 15 data sets over 30 independent runs. The number of hidden nodes and ridge parameters are also shown in Table 3.

The *G*-mean was used as the measure to compare the classifiers, due to its usage in previous studies regarding WELM [14]. In addition, the *G*-mean is the trade-off between sensitivity and specificity, whereas the minority classes often represent the main class of interest in imbalanced domains. In order to measure the effectiveness of the competing algorithms, the sensitivity and specificity are also given. The best classification results in terms of the *G*-mean and corresponding AUC, SN, and SP are shown in Table 4.

The following are observed in Figs. 1 and 2:

1. Figure 1 shows the smooth increase in fitness value (AUC) of the three approaches, with the except of the

Ecoli0. This indicated that the fitness function (AUC) was effective for these tasks during the training phase. As can be seen in Fig. 1 and Table 4, although there was no variance observed for the AUC value of the Ecoli0 in the SDE-WELM, ABC-WELM, and E-WELM, the AUC, *G*-mean, and SN of these approaches were found to be higher than those of the WELM. These findings indicated that the fitness function (AUC) evolved the classifiers with best solution at the first generation.

2. Although there were some fluctuations, a trend could still be observed from the curves in Fig. 2, in which the norm of the output weights of the SLFN was found to be decreasing in the ABC-WELM, and SDE-WELM for all of the tasks, and for seven tasks in the E-WELM

Table 5 Specification of the imbalanced data sets

Data sets	#Ex	#Atts	Minority class	IR
Glass0	214	9	Float building windows	2.06
Glass1	214	9	Non-float	1.82
Glass2	214	9	Float vehicle windows	10.39
Glass4	214	9	Containers	15.46
Glass5	214	9	Tableware	22.78
Glass6	214	9	Headlamps	6.38
Glass0123vs456	214	9	Containers; tableware; headlamps	3.19
Glass016vs2	192	9	Float vehicle windows	10.29
Glass016vs5	184	9	Tableware	19.44
Ionosphere	351	34	Bad	8.24
Ecoli0	336	7	cp	1.35
Ecoli1	336	7	im	3.36
Ecoli2	336	7	PP	5.46
Ecoli3	336	7	imU	8.19
Ecoli4	336	7	OM	15.8
Ecoli0vs1	220	7	im	1.86
Ecoli0137vs26	311	7	PP; imL	4.76
Vehicle0	846	18	Van	3.25
Vehicle1	846	18	saab	2.90
Vehicle2	846	18	bus	2.88
Vehicle3	846	18	opel	2.99
Segment1	2310	19	brickface	6.00
Wisconsin	683	9	Malignant	1.86
Yeast1	1484	8	nuc	2.46
Yeast3	1484	8	ME3	8.11
Yeast4	1484	8	ME2	28.10
Yeast5	1484	8	ME1	32.73
Yeast6	148	8	EXC	39.10
Yeast1vs7	459	8	VAC	14.3
Yeast2vs4	295	8	ME2	4.78
Yeast2vs8	264	8	POX	12.2
Yeast05679vs4	628	8	ME2	11.31
Yeast1289vs7	728	8	VAC	23.26
Yeast1458vs7	574	8	VAC	18.13
Pima	768	8	Class0	1.90
Abalone9vs18	731	8	Class9	16.68
Abalone19	4174	8	Class19	128.87
Pageblock1	5473	10	Horiz. line	15.6
Pageblock13vs2	472	10	Graphic	15.85
Wine1	178	13	Class 1	2.02

Table 6 Average AUC of the 40 imbalanced classification problems

Data sets	WSVM	OverBagging	UnderBagging	RUSBoost	EasyEnsemble	WELM	EN-WELM	DE-WELM	ABC-WELM	SDE-WELM
Glass0	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	98.11 ± 0.25	98.31 ± 0.24	**98.35 ± 0.24**
Glass1	75.28 ± 2.48e−16	76.73 ± 1.37	77.01 ± 2.37	84.18 ± 4.75	**87.42 ± 2.91**	74.90 ± 3.25	76.41 ± 2.76	76.98 ± 2.65	77.29 ± 2.03	79.60 ± 1.59
Glass2	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	75.9 ± 1.21	76.76 ± 1.07
Glass4	78.13 ± 6.77e−17	**98.24 ± 1.35e−17**	78.78 ± 0	97.45 ± 0	98.04 ± 0	89.84 ± 0.29	91.36 ± 0.25	91.97 ± 0.20	92.58 ± 0.19	93.10 ± 0.14
Glass5	74.89 ± 2.25e−16	**97.49 ± 0**	66.32 ± 0	86.30 ± 16.79	95.52 ± 3.57	86.92 ± 2.69	87.38 ± 1.45	88.10 ± 1.23	88.56 ± 1.30	89.30 ± 1.09
Glass6	60.96 ± 0	85.74 ± 0	86.08 ± 3.3e−16	**97.02 ± 0.70**	95.76 ± 0	89.39 ± 0	89.73 ± 0	89.26 ± 0	89.85 ± 0	90.24 ± 0
Glass0123vs456	86.50 ± 3.38e−16	76.06 ± 2.25e−16	93.21 ± 2.82	95.33 ± 1.94	**97.00 ± 1.56**	84.65 ± 1.46	86.29 ± 1.36	87.97 ± 0.99	88.56 ± 0.63	89.32 ± 0.39
Glass16vs2	66.70 ± 2.03e−16	77.64 ± 6.38	84.20 ± 0	84.40 ± 6.85	85.44 ± 6.3	84.71 ± 2.94	84.80 ± 2.35	85.20 ± 1.50	85.99 ± 1.41	**86.67 ± 1.20**
Glass16vs5	76.88 ± 6.77e−17	98.69 ± 2.93e−16	47.65 ± 2.37	92.09 ± 5.36	94.27 ± 5.68	97.86 ± 2.50	97.98 ± 0.50	98.58 ± 0.39	98.95 ± 0.35	**99.47 ± 0.30**
Ionosphere	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**	95.7 ± 1.20	95.89 ± 0.89
Ecoli0	99.86 ± 0	99.95 ± 0	**99.95 ± 0.50**	98.59 ± 0.55	97.64 ± 0.37	99.88 ± 0	99.95 ± 0	99.91 ± 0	99.96 ± 0	99.89 ± 0
Ecoli1	84.18 ± 2.2e−16	90.69 ± 0.27	87.60 ± 1.0	94.86 ± 0.99	**95.42 ± 0.58**	93.32 ± 4.5e−16	93.40 ± 4.5e−16	93.69 ± 5.6e−16	93.85 ± 4.3e−16	93.41 ± 4.5e−16
Ecoli2	87.61 ± 2.71e−16	92.10 ± 2.82e−17	87.75 ± 2.83	95.04 ± 1.61	97.48 ± 0.91	98.24 ± 0.30	98.58 ± 0.22	98.61 ± 0.21	98.65 ± 0.20	**98.77 ± 0.20**
Ecoli3	75.84 ± 4.5e−16	73.89 ± 1.1e−16	66.64 ± 3.58	**95.53 ± 2.09**	93.40 ± 1.52	84.05 ± 0.64	84.23 ± 0.34	84.57 ± 0.68	84.65 ± 0.37	84.80 ± 0.24
Ecoli4	99.66 ± 3.83e−17	99.86 ± 2.48e−4	91.94 ± 4.51e−17	95.15 ± 4.53	98.64 ± 3.25	99.94 ± 0.30	99.97 ± 4.96e−17	97.98 ± 2.32e−17	97.98 ± 1.01e−17	**100 ± 0**
Ecoli0vs1	99.58 ± 2.71e−16	96.73 ± 1.35e−17	99.92 ± 0.57	99.67 ± 0.88	96.45 ± 3.52	99.43 ± 0.12	99.75 ± 0.32	99.95 ± 0.30	99.95 ± 0.29	**99.97 ± 0.30**
Ecoli0137vs26	86.50 ± 3.38e−16	96.15 ± 1.50e−4	99.51 ± 1.30e−4	96.74 ± 2.12	95.81 ± 1.00	99.81 ± 0.72	99.77 ± 0.13	99.87 ± 0.20	99.91 ± 0.20	**99.95 ± 0.17**
Vehicle0	92.78 ± 0	94.62 ± 3.3e−16	94.62 ± 3.3e−16	94.85 ± 0.37	99.26 ± 0.30	98.24 ± 0.51	98.80 ± 0.50	98.43 ± 0.40	98.80 ± 0.50	**99.41 ± 0.39**
Vehicle1	74.51 ± 3.16e−16	**87.54 ± 0.94**	65.08 ± 0.59	85.7 ± 1.45	86.78 ± 0.83	85.20 ± 1.57	87.07 ± 1.50	87.12 ± 1.29	87.24 ± 1.20	87.47 ± 1.17
Vehicle2	96.68 ± 3.38e−16	96.91 ± 5.22e−4	98.40 ± 0.12	83.74 ± 1.52	98.47 ± 0.21	98.91 ± 0.74	99.70 ± 0.45	99.81 ± 0.20	99.89 ± 0.32	**99.97 ± 0.21**
Vehicle3	73.22 ± 1.80e−16	66.37 ± 0.56	66.31 ± 0.92	81.82 ± 2.56	84.58 ± 0.96	86.05 ± 1.69	86.38 ± 1.32	87.14 ± 1.00	87.63 ± 1.06	**88.21 ± 0.77**
Segment1	99.77 ± 1.16e−16	99.99 ± 0.18	99.59 ± 4.74e−16	99.93 ± 9.31e−4	**100± 7.84e−5**	99.40 ± 0.16	99.55 ± 0.14	99.90 ± 0.10	99.95 ± 0.12	99.97 ± 0.10
Wisconsin	97.23 ± 0	97.05 ± 0.22	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	**99.32 ± 0.15**
Yeast1	65.97 ± 3.6e−16	70.62 ± 0.36	70.62 ± 0.36	74.99 ± 2.25	**78.78 ± 0.63**	70.90 ± 0.62	72.17 ± 0.51	71.07 ± 0.67	72.23 ± 0.50	72.39 ± 0.47
Yesat3	78.93 ± 1.1e−16	58.00 ± 6.8e−4	58.00 ± 6.8e−4	95.29 ± 0.42	**96.55 ± 0.23**	91.82 ± 0.73	92.10 ± 0.59	92.00 ± 0.17	91.87 ± 0.57	92.21 ± 0.60
Yesat4	63.95 ± 2.14e−16	72.08 ± 1.65e−4	56.94 ± 1.35e−17	79.57 ± 2.84	91.02 ± 1.47	84.92 ± 1.09	87.43 ± 0.28	89.12 ± 0.17	90.51 ± 0.15	**91.14 ± 0.15**
Yesat5	78.05 ± 2.93e−16	98.60 ± 3.22	44.24 ± 0.85	93.99 ± 1.60	99.11 ± 0.43	95.77 ± 0.12	95.90 ± 3.56e−4	97.63 ± 1.02e−4	98.98 ± 2.31e−5	**99.20 ± 1.15e−5**
Yesat6	74.54 ± 2.25e−16	90.77 ± 0.51	58.92 ± 2.25e−16	91.82 ± 2.56	**94.82 ± 6.33**	90.57 ± 0.30	90.74 ± 6.31e−4	91.36 ± 0	91.58 ± 0	92.69 ± 0
Yesat1vs7	45.15 ± 7.90e−17	84.93 ± 5.03	34.32 ± 0.73	78.31 ± 4.22	83.50 ± 2.54	91.33 ± 0.64	93.45 ± 0	95.21 ± 0	95.71 ± 0	**96.47 ± 0**
Yesat2vs4	64.17 ± 225e−16	93.56 ± 2.20	68.87 ± 5.10	94.12 ± 2.20	93.71 ± 1.14	92.22 ± 0.29	92.64 ± 2.48e−17	93.18 ± 1.05e−17	93.65 ± 1.05e−17	**94.68 ± 0**
Yesat2vs8	62.90 ± 9.03e−17	80.45 ± 1.35e−16	44.82 ± 4.58	**94.02 ± 6.02**	86.45 ± 2.83	84.76 ± 6.68	86.17 ± 0.12	88.45 ± 0.32	88.96 ± 0.21	89.59 ± 0.19
Yeast05679vs4	62.71 ± 1.01e−16	91.43 ± 0.99	19.43 ± 2.58	91.94 ± 3.48	**92.18 ± 2.29**	67.08 ± 2.32	68.71 ± 1.82	74.52 ± 0.50	75.69 ± 0.36	76.06 ± 0.43
Yesat1289vs7	64.60 ± 2.58e−17	81.09 ± 1.64	12.48 ± 3.38e−17	74.99 ± 4.54	81.51 ± 2.85	82.37 ± 1.65	84.63 ± 3.44	85.48 ± 2.89	86.54. ± 3.12	**87.46 ± 3.15**
Yesat1458vs7	14.45 ± 0	75.85±	3.33 ± 3.11	82.60 ± 5.07	76.39 ± 4.18	88.31 ± 3.42	89.09 ± 2.63	90.34 ± 1.68	91.68 ± 1.87	**91.84 ± 2.31**
Pima	71.76 ± 0	68.76 ± 0.95	68.76 ± 0.95	76.79 ± 1.53	**83.12 ± 0.96**	76.61 ± 1.14	76.83 ± 0.99	76.89 ± 0.86	80.11 ± 0.86	80.39 ± “0.73

Table 6 continued

Data sets	WSVM	OverBagging	UnderBagging	RUSBoost	EasyEnsemble	WELM	EN-WELM	DE-WELM	ABC-WELM	SDE-WELM
Abalone9vs18	80.21 ± 3.4e−16	6.41 ± 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	74.58 ± 6.80
Abalone19	75.34 ± 3.9e−16	74.25 ± 0.36	74.25 ± 0.36	77.56 ± 3.04	81.53 ± 1.65	78.36 ± 1.69	79.45 ± 1.59	79.81 ± 4.8e−4	80.79 ± 1.40	**81.71 ± 1.39**
Pageblock1	90.94 ± 2.2e−16	74.00 ± 2.2e−16	74.00 ± 2.2e−16	**98.70 ± 0.21**	98.49 ± 0.56	95.34 ± 0.37	95.77 ± 0.40	95.43 ± 3.9e−4	95.63 ± 0.36	95.62 ± 0.41
Pageblock13vs2	93.33 ± 6.77e−16	98.55 ± 4.51e−17	83.67 ± 1.30e−4	99.04 ± 5.35	99.92 ± 0.76	99.43 ± 0.51	99.67 ± 0.39	99.84 ± 0.32	99.91 ± 0.51	**99.97 ± 0.29**
Wine1	99.00 ± 0	98.81 ± 3.81e−4	**99.98 ± 0.18**	98.61 ± 0.50	99.28 ± 0.24	96.64 ± 0.30	96.98 ± 3.91e−4	97.35 ± 0.21	97.64 ± 0.19	98.20 ± 0.28

during the training process. This was determined to be consistent with the knowledge that the neural networks tended to have better generalization performances with smaller norms of output weights. Also, though the norms of the output weights of the SLFN were observed to be increased for eight data sets in the DE-WELM, the testing AUC, G-mean, and SN were found to be slightly less than those of the WELM in only the Glass2 and Wisconsin tasks, as illustrated in Fig. 2 and Table 4. These results suggested that there was an optimization bias of the network input weights and hidden nodes, according to not only the area under curve (AUC) on validation set, but also the norm of the output weights.

3. There were almost no variances observed for the AUC and norms after 50 generations in the SDE-WELM for the majority of data sets. These results indicated that the fitness function (AUC) had evolved the classifiers with the best solutions at 50 generations, as illustrated in Figs. 1 and 2.

The following are observed from the results detailed in Table 4:

1. The G-mean and AUC value of the DE-WELM,ABC-WELM and SDE-WELM were higher than those of the WELM throughout the majority of the data sets, which indicated that using a stochastic search algorithm to find the optimal input weights, hidden nodes parameters, and training sample weights of the WELM could obtain a better performance than the WELM.
2. The best performance of the proposed method was found to be on 14 of the 15 data sets for the G-mean; 13 of the 15 for the SN; and 13 of the 15 for the AUC.
3. The proposed SDE-WELM was determined to perform better than the original WELM, DE-WELM, and ABC-WELM, in terms of producing higher testing AUC, G-mean, SN, and lower STD. These results supported the argument that inappropriate choices of strategies and control parameters involved in the differential evolutionary algorithm may result in premature convergence or stagnation. Also, the self-adaptive determination of the suitable trial vector generation strategies and associated control parameters for different applications during the training phase of the SLFN may also lead to different network generalization performances [39].

According to the experimental results of all the data sets, the same value of γ was observed, and only minimum differences in the values of the hidden nodes were evident, when the WELM, DE-WELM ABC-WELM, and SDE-WELM was used to obtain the best results in terms of the G-mean. Therefore, it was determined γ could be fixed as the best value, and tune L when selecting the parameter of the model. As a

Table 7 Average *G*-mean of the 40 imbalanced classification problems

Data sets	WSVM	OverBagging	UnderBagging	RUSBoost	EasyEnsemble	WELM	EN-WELM	DE-WELM	ABC-WELM	SDE-WELM
Glass0	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	83.27 ± 1.71	83.92 ± 1.34	**84.18 ± 1.29**
Glass1	74.80 ± 2.03e−17	70.14 ± 1.55	76.27 ± 2.53	74.34 ± 5.77	**79.43 ± 3.68**	74.87 ± 2.57	74.91 ± 2.77	75.32 ± 1.30	75.79 ± 0.97	76.98 ± 0.50
Glass2	64.73 ± 0	75.63 ± 1.2e−16	77.58 ± 1.90	80.30 ± 11.97	68.74 ± 5.25	83.52 ± 1.39	83.61 ± 1.14	83.72 ± 1.23	83.89 ± 1.07	**84.37 ± 0.86**
Glass4	84.18 ± 2.71e−17	**95.82 ± 9.03e−16**	85.66 ± 0.55	92.96 ± 0	95.34 ± 0	90.30 ± 0.29	90.94 ± 0.25	91.35 ± 0.20	91.69 ± 0.20	92.58 ± 0.17
Glass5	73.62 ± 2.25e−17	92.32 ± 0	70.26 ± 0	82.11 ± 16.01	**93.60 ± 6.41**	85.91 ± 3.68	86.26 ± 2.94	86.43 ± 1.30	86.30 ± 1.65	86.92 ± 0.20
Glass6	69.73 ± 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−17	94.20 ± 1.1e−16	94.75 ± 1.3e−17	**94.97 ± 2.9e−17**
Glass0123vs456	92.86 ± 9.03e−17	72.67 ± 2.48e−16	95.42 ± 1.54	95.25 ± 2.52	**95.87 ± 2.80**	91.32 ± 1.42	92.85 ± 0.44	93.32 ± 0.32	93.56 ± 0.29	94.26 ± 0.24
Glass16vs2	71.15 ± 564e−17	71.42 ± 1.72	74.81 ± 0	74.18 ± 1.73	**77.54 ± 7.34**	75.72 ± 5.67	76.19 ± 5.7	76.19 ± 3.20	76.89 ± 3.11	76.52 ± 1.26
Glass16vs5	77.17 ± 9.03e−17	95.89 ± 0	60.54 ± 2.77	87.11 ± 7.34	93.17 ± 6.24	98.81 ± 1.89	98.97 ± 0.30	99.45 ± 0.32	99.63 ± 0.30	**99.98 ± 0.36**
Ionosphere	90.99 ± 3.4e−16	90.43 ± 5.6e−15	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**	90.99 ± 1.45	91.41 ± 1.47
Ecoli0	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	97.59 ± 0	97.89 ± 0	**98.21 ± 0**
Ecoli1	84.15 ± 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	88.58 ± 5.6e−16	88.72 ± 4.3e−16	89.06 ± 4.5e−16
Ecoli2	93.58 ± 3.83e−17	88.54 ± 2.03e−16	93.53 ± 1.65	92.35 ± 1.77	90.79 ± 2.31	94.36 ± 0.17	95.11 ± 0.11	95.39 ± 0.11	95.87 ± 0.09	**96.35 ± 0.10**
Ecoli3	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 ± 0.40	90.70 ± 0.37	**90.75 ± 0.34**
Ecoli4	92.19 ± 3.61e−17	97.34 ± 3.05e−4	94.42 ± 3.90e−17	92.41 ± 6.89	96.32 ± 4.89	96.83 ± 0.22	97.91 ± 2.48e−17	98.06 ± 2.01e−17	98.51 ± 1.39e−17	**99.62 ± 1.13e−17**
Ecoli0vs1	98.25 ± 1.58e−16	97.25 ± 1.58e−16	98.4 ± 0.29	96.63 ± 1.67	97.44 ± 4.03	99.21 ± 70	99.37 ± 6.48e−4	99.49 ± 5.61e−4	99.70 ± 0	**99.84 ± 3.19e−4**
Ecoli0137vs26	92.86 ± 9.03e−17	93.36 ± 0.34	90.10 ± 0.55	92.03 ± 1.91	90.67 ± 1.69	94.07 ± 0.65	94.29 ± 0.68	94.32 ± 0.69	94.56 ± 0.61	**95.13 ± 0.60**
Vehicle0	92.13 ± 5.6e−16	94.74 ± 0.18	93.98 ± 2.2e−16	94.90 ± 1.30	95.92 ± 0.50	98.46 ± 0.33	98.50 ± 0.29	98.52 ± 0.22	98.51 ± 0.28	**98.91 ± 0.22**
Vehicle1	73.64 ± 2.84e−17	**88.94 ± 0.87**	62.75 ± 0.70	77.97 ± 2.05	79.88 ± 1.41	85.07 ± 1.35	87.12 ± 1.46	87.31 ± 1.35	87.56 ± 1.32	88.34 ± 1.30
Vehicle2	95.66 ± 4.74e−17	96.91 ± 8.56e−4	97.61 ± 0.15	77.39 ± 2.09	97.94 ± 0.80	98.89 ± 0.37	99.41 ± 0.71	99.68 ± 0.50	99.76 ± 0	**99.92 ± 0**
Vehicle3	71.01 ± 2.25e−17	79.98 ± 0.99	61.12 ± 0.26	83.14 ± 3.49	78.62 ± 1.78	85.29 ± 1.59	85.53 ± 1.34	85.92 ± 0.50	86.32 ± 0.44	**87.17 ± 0.30**
Segment1	99.51 ± 1.35e−16	**99.89 ± 0.15**	99.10 ± 1.58e−16	99.52 ± 0.25	**99.72 ± 0.14**	99.36 ± 0.15	99.49 ± 0.10	99.57 ± 0	99.59 ± 0	99.69 ± 0
Wisconsin	97.11 ± 3.4e−16	96.07 ± 3.3e−16	96.86 ± 0.11	94.72 ± 0.53	97.12 ± 1.11	97.14 ± 0.38	97.34 ± 0.32	97.37 ± 0.32	97.35 ± 0.30	**97.69 ± 0.27**
Yeast1	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	71.57 ± 0.67	71.93 ± 0.51	**72.36 ± 0.48**
Yesat3	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	93.00 ± 9.7e−4	93.04 ± 0.57	**93.25 ± 0.57**
Yesat4	53.52 ± 2.25e−16	84.35 ± 8.22e−5	6.70 ± 0	83.13 ± 364	**84.90 ± 2.31**	83.18 ± 2.42	83.26 ± 1.78	83.21 ± 1.12	83.47 ± 1.03	84.54 ± 0.97
Yesat5	82.00 ± 3.38e−17	97.40 ± 2.89	29.67 ± 0.79	94.78 ± 3.66	97.51 ± 1.65	96.60 ± 8.32e−4	96.65 ± 6.21e−4	97.01 ± 3.97e−5	97.65 ± 2.69e−5	**98.38 ± 2.69e−5**
Yesat6	64.76 ± 2.59e−17	88.24 ± 1.58	21.32 ± 1.58e−16	84.78 ± 3.49	87.70 ± 7.34	88.38 ± 0.15	88.82 ± 3.61e−4	88.97 ± 1.12e−4	89.30 ± 8.91e−5	**91.63 ± 8.91e−5**
Yesat1vs7	66.85 ± 7.68e−17	78.76 ± 2.39	57.12 ± 0.59	72.89 ± 6.51	74.08 ± 4.77	77.77 ± 0.40	78.46 ± 2.48e−17	79.34 ± 8.50e−18	79.98 ± 3.50e−19	**80.39 ± 2.97e−19**
Yesat2vs4	79.53 ± 3.04e−17	87.83 ± 4.38	82.86 ± 3.23	**90.73 ± 4.38**	89.65 ± 2.73	88.32 ± 0.17	89.35 ± 0	89.39 ± 0	89.74 ± 0	90.13 ± 0
Yesat2vs8	50.69 ± 1.69e−17	73.79 ± 1.12e−16	58.17 ± 3.55	**85.89 ± 6.70**	79.05 ± 4.47	72.26 ± 4.60	72.96 ± 5.57e−4	75.39 ± 4.87e−4	77.91 ± 2.36e−4	77.98 ± 8.59e−5
Yeast05679vs4	48.91 ± 2.48e−17	84.67 ± 0.83	38.75 ± 2.73	82.83 ± 3.12	**86.13 ± 3.84**	81.56 ± 1.50	82.58 ± 2.74	83.69 ± 1.84	84.17 ± 1.65	84.75 ± 1.13

Table 7 continued

Data sets	WSVM	OverBagging	UnderBagging	RUSBoost	EasyEnsemble	WELM	EN-WELM	DE-WELM	ABC-WELM	SDE-WELM
Yesat1289vs7	49.41 ± 3.37e−17	76.77 ± 3.95	17.04 ± 0	74.07 ± 6.37	74.01 ± 3.38	77.24 ± 2.39	77.59 ± 1.75	79.36 ± 1.52	79.82 ± 1.98	**81.49 ± 1.27**
Yesat1458vs7	29.24 ± 0	71.24 ± 4.02	9.88 ± 5.38	74.45 ± 7.38	70.76 ± 6.6	73.55 ± 2.21	76.49 ± 1.78	76.71 ± 0.89	77.03 ± 0.75	**78.19 ± 0.42**
Pima	69.30 ± 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	75.83 ± 0.71	75.96 ± 0.64	**76.28± ‘0.64**
Abalone9vs18	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	85.90 ± 3.79	**86.14 ± 4.18**
Abalone19	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	75.52 ± 7.2e−4	76.34 ± 2.40	**76.89 ± 2.20**
Pageblock1	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	95.35 ± 3.5e−4	95.59 ± 0.15	**96.07 ± 0.17**
Pageblock13vs2	96.51 ± 1.58e−17	97.32 ± 2.25e−17	90.96 ± 0.55	99.10 ± 4.77	98.84 ± 0.92	99.21 ± 0.26	99.32 ± 0.21	99.27 ± 0.30	99.39 ± 0.13	**99.94 ± 0.12**
Wine1	98.06 ± 3.61e−17	97.9 ± 0.77	98.24 ± 0.97	98.02 ± 1.66	**98.65 ± 1.77**	96.50 ± 1.53	96.80 ± 1.11	96.57 ± 0.58	97.21 ± 0.060	98.39 ± 0.49

result, the time consumption for parameter selection will be largely reduced.

The aim of this subsection was solely to analyze the variances of the norm and AUC in the three approaches, in which the AUC was used as the fitness function. This study randomly crossed the data sets, as shown in Table 3. Therefore, the values displayed in Table 4 may not be a true reflection of performances of the three methods. A more thorough testing was thereby detailed in a later subsection later.

4.3 Performance evaluation of the proposed SDE-WELM

In this study, in order to thoroughly evaluate the performance of the proposed SDE-WELM, a comparison was conducted of the WELM, ensemble WELM (EN-WELM), DE-WELM, ABC-WELM, WSVM, RUSBoost, EsayEnsemble, OverBagging, and UnderBagging for 40 unbalanced data sets, as detailed in Table 5. The SVM was kernel-based learning similar to ELM. Furthermore, the SVM was able to provide relatively robust classification results when applied to the imbalanced data sets. Therefore, the SVM was selected to implement the base classifier of the RUSBoost, EsayEnsemble, OverBagging, and UnderBagging and was implemented in LIBSVM. All of the parameters for the SVM were selected from the set $\left\{10^{-10}, \ldots, 10^{10}\right\}$. The resampling rates of the RUSboost and EsayEnsemble were set as 5%. The number of iteration was set as 100 in the OverBagging, UnderBagging, RUSboost, and EsayEnsemble. Then, in order to compare different algorithms and to determine whether or not significant differences existed among them, this study used nonparametric tests suggested in [40,41] to present the comparison as a statistical support. A Friedman test was used for the multiple comparisons, in order to detect the statistical differences among a group of results. Then, a Nemenyi post hoc test was used to verify if the proposed algorithm was significantly better than others.

Since the comparison was only meaningful for the practical structures, the number of hidden neurons L, and the regularization parameter γ in the WELM were chosen using a fivefold cross-validation for the purpose of ensuring the use of the optimal models. The entire cross-validation process was repeated for ten runs, and the final results were the averages of those ten runs. In this way, the performances variance of all the tests which were caused by the different initial parameters and random partitions of the data sets could be both effectively analyzed. For the EN-WELM, the number of base classifiers was 10. The average classification results, and the corresponding standard deviations which were denoted by the AUC, *G*-mean, SN, and SP are, respectively, detailed in Tables 6, 7, 8, and 9.

The following are observed from the results detailed in Tables 6, 7, 8, and 9:

Table 8 Average SN of the 40 imbalanced classification problems

Data sets	WSVM	OverBagging	UnderBagging	RUSBoost	EasyEnsemble	WELM	EN-WELM	E-WELM	ABC-WELM	SDE-WELM
Glass0	77.81 ± 3.4e−16	88.11 ± 0	87.31 ± 1.14	92.54 ± 6.43	81.92 ± 4.73	90.81 ± 2.80	90.91 ± 2.65	90.70 ± 2.12	92.45 ± 1.53	**92.71 ± 1.41**
Glass1	65.49 ± 1.91e−17	75.99 ± 2.21	74.74 ± 3.77	**92.04 ± 8.39**	81.06 ± 6.22	73.91 ± 3.09	73.69 ± 4.73	75.36 ± 2.98	76.21 ± 2.63	77.05 ± 1.96
Glass2	49.17 ± 0	80.36 ± 0	81.57 ± 3.06	83.60 ± 2.52	**88.50 ± 6.58**	86.30 ± 0.85	86.62 ± 0.61	86.49 ± 0.67	87.43 ± 0.62	87.59 ± 0.43
Glass4	73.33 ± 2.71e−17	**100 ± 0**	76.67 ± 0.87	92.22 ± 0	100 ± 0	93.38 ± 2.34e−17	95.63 ± 0	93.69 ± 0	94.21 ± 0	93.02 ± 2.34e−19
Glass5	70.00 ± 0	100 ± 0	63.33 ± 0	76.67 ± 15.69	97.33 ± 5.48	100 ± 4.25	100 ± 4.47	100 ± 3.51	100 ± 3.21	**100 ± 1.69**
Glass6	61.40 ± 0	87.62 ± 0	87.67 ± 3.3e−16	91.22 ± 6.34	90.66 ± 0	90.35 ± 5.7e−16	90.13 ± 0	90.57 ± 0	91.11 ± 0	**91.34 ± 0**
Glass0123vs456	90.10 ± 2.25e−17	79.46 ± 4.74e−16	95.13 ± 2.68	98.29 ± 3.08	**98.67 ± 4.11**	87.17 ± 1.50	88.79 ± 2.31	88.36 ± 1.26	89.12 ± 1.02	90.51 ± 0.98
Glass16vs2	66.7 ± 0	82.67 ± 1.31	70.71±	**88.86 ± 1.31**	85.00 ± 1.47	70.00 ± 2.98	78.46 ± 1.98	79.23 ± 1.99	79.89 ± 1.69	81.56 ± 1.30
Glass16vs5	70.78 ± 0	100 ± 0	48.00 ± 4.40	80.00 ± 1.47	100 ± 1.29	100 ± 2.74	100 ± 0	100 ± 0	100 ± 0	**100 ± 0**
Ionosphere	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	85.43 ± 3.45	83.10 ± 4.72	83.65 ± 3.31	84.32 ± 3.01
Ecoli0	96.44 ± 0	98.02 ± 0	98.13 ± 1.27	97.61 ± 2.06	96.33 ± 0.87	97.82 ± 0	98.01 ± 0	98.29 ± 0	98.25 ± 0	**98.52 ± 0**
Ecoli1	75.37 ± 4.5e−16	85.28 ± 0	79.30 ± 1.69	92.70 ± 4.06	91.30 ± 3.81	92.07 ± 0	92.70 ± 4.5e−16	92.17 ± 3.7e−16	**92.97 ± 2.3e-16**	92.34 ± 4.3e−16
Ecoli2	90.14 ± 1.80e−17	87.13 ± 1.90e−17	89.42 ± 2.48	92.48 ± 1.83	93.29 ± 2.54	94.44 ± 4.68e−17	94.67 ± 2.48e−17	94.99 ± 1.120e−17	95.84 ± 1.01e−17	**97.12 ± 3.68e-19**
Ecoli3	70.16 ± 3.4e−16	70.15 ± 0	56.74 ± 5.38	92.21 ± 4.10	86.11 ± 0	91.90 ± 023	92.02 ± 0.22	92.39 ± 0.7	92.63 ± 0.12	**92.69 ± 0.15**
Ecoli4	87.02 ± .93e−16	100 ± 2.88e−17	89.90 ± 0	88.22 ± 7.73	97.78 ± 5.25	**100 ± 0**	**100 ± 0.12**	**100 ± 0**	**100 ± 0**	**100 ± 0**
Ecoli0vs1	96.58 ± 1.35e−17	97.27 ± 1.12e−16	96.92 ± 0.42	97.51 ± 0.77	97.70 ± 0.65	99.98 ± 0	**100 ± 0.24**	**100 ± 0.22**	**100 ± 0.20**	**100 ± 0.20**
Ecoli0137vs26	90.10 ± 2.25e−17	92.25 ± 0.68	85.43 ± 0.87	90.74 ± 1.00	91.14 ± 2.07	91.72 ± 0.69	91.79 ± 0.70	92.39 ± 0.59	92.61 ± 0.55	**93.49 ± 0.54**
Vehicle0	86.73 ± 0	91.65 ± 1.1e−16	89.96 ± 3.3e−16	97.31 ± 2.38	97.11 ± 0	99.31 ± 0	99.49 ± 0	99.32 ± 0	99.56 ± 0	**100 ± 0**
Vehicle1	61.40 ± 1.12e−17	89.16 ± 1.43	42.46 ± 0.88	88.87 ± 3.68	83.49 ± 2.81	86.74 ± 2.59	89.46 ± 2.39	89.52 ± 1.00	89.74 ± 1.10	**90.31 ± 0.65**
Vehicle2	92.47 ± 1.80e−17	97.54 ± 0.17	95.71 ± 0.28	87.36 ± 3.92	97.90 ± 0.80	99.29 ± 0.32	99.37 ± 1.27	99.51 ± 0.68	99.68 ± 0.20	**99.82 ± 0.21**
Vehicle3	57.11 ± 1.91e−17	75.97 ± 1.52	40.66 ± 0.81	78.58 ± 5.66	82.44 ± 3.48	85.28 ± 3.10	85.73 ± 2.47	86.12 ± 1.20	86.91 ± 1.00	**88.63 ± 0.63**
Segment1	99.07 ± 6.77e−17	100 ± 0.25	98.26 ± 2.93e−17	99.43 ± 0.19	**99.71 ± 0.12**	99.05 ± 0.13	99.12 ± 0.14	99.31 ± 0	99.39 ± 0	99.61 ± 0
Wisconsin	98.70 ± 4.5e−16	97.60 ± 0	98.37 ± 0	**98.72 ± 0**	97.21 ± 2.00	97.30 ± 0.62	97.60 ± 0.48	98.09 ± 0.45	97.75 ± 0.35	98.24 ± 0.35
Yeast1	38.67 ± 0	85.74 ± 0.85	85.63 ± 0.79	**86.09 ± 7.15**	71.98 ± 3.12	72.62 ± 1.06	72.40 ± 1.0	73.26 ± 0.82	73.42 ± 1.10	73.55 ± 0.84
Yesat3	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	93.16 ± 3.4e−16	93.19 ± 0.96	93.21 ± 0.90	**93.31 ± 0.99**
Yesat4	29.72 ± 6.77e−17	84.73 ± 2.25e−17	1.43 ± 0	78.93 ± 6.89	**91.02 ± 3.63**	80.74 ± 4.41	82.50 ± 1.10	83.69 ± 0.56	83.92 ± 0.37	85.65 ± 0.21
Yesat5	96.48 ± 2.25e−17	100 ± 4.45	11.21 ± 1.02	93.02 ± 6.75	100 ± 2.97	100 ± 0	100 ± 0	100 ± 0	100 ± 0	**100 ± 0**
Yesat6	42.60 ± 2.25e−17	90.46 ± 2.03	7.58 ± 2.54e−17	78.58 ± 5.66	**91.92 ± 1.44**	85.95 ± 0.90	86.79 ± 0	87.98 ± 0	88.41 ± 0	89.84 ± 0
Yesat1vs7	45.25 ± 5.64e−17	**81.67 ± 1.20**	34.32 ± 0.73	71.75 ± 1.21	80.86 ± 7.83	76.87 ± 9.36e−17	80.17 ± 0	80.21 ± 0	80.69 ± 0	81.02 ± 0
Yesat2vs4	69.08 ± 5.0e−17	88.32 ± 8.24	73.06 ± 5.34	**94.82 ± 8.24**	94.56 ± 5.22	88.35 ± 1.20	90.20 ± 0.36	90.20 ± 0.12	90.96 ± 0.20	91.49 ± 2.34e−17
Yesat2vs8	26.1 ± 3.95e−17	75.00 ± 3.38e−17	45.00 ± 4.60	80.29 ± 5.48	**87.14 ± 4.41**	57.33 ± 4.75	60.78 ± 3.68	61.68 ± 1.58	62.87 ± 1.29	61.27 ± 2.92e−17

Table 8 continued

Data sets	WSVM	OverBagging	UnderBagging	RUSBoost	EasyEnsemble	WELM	EN-WELM	E-WELM	ABC-WELM	SDE-WELM
Yeast05679vs4	24.82 ± 0	85.50 ± 0.90	19.60 ± 2.58	83.60 ± 5.32	**92.04 ± 7.61**	73.98 ± 2.28	77.52 ± 1.49	78.50 ± 0.67	80.36 ± 0.50	81.18 ± 0.44
Yesat1289vs7	26.45 ± 5.64e−18	81.19 ± 1.65	12.50 ± 0	69.79 ± 1.13	**81.74 ± 7.13**	75.83 ± 4.41	78.23 ± 3.32	80.36 ± 2.69	81.69 ± 2.12	82.16 ± 1.06
Yesat1458vs7	14.67 ± 0	78.40 ± 1.63	4.00 ± 3.11	75.50 ± 9.28	82.38 ± 7.70	76.95 ± 3.51	82.06 ± 2.90	82.07±	83.19±	**84.21 ± 0.36**
Pima	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	73.92 ± 1.04	73.94 ± 1.19	74.60 ± 0.91	75.32± “0.88
Abalone9vs18	78.47 ± 0	12.23 ± 0	**100 ± 0**	86.74 ± 7.17	82.70 ± 5.38	70.36 ± 0	71.00 ± 0	70.77 ± 0	70.69 ± 0	70.99 ± 0
Abalone19	75.27 ± 0	73.51 ± 2.08	68.15 ± 1.54	56.16 ± 4.36	**82.57 ± 4.52**	73.28 ± 5.07	74.67 ± 3.4e−4	73.49 ± 4.97	75.09 ± 4.43	75.35 ± 3.67
Pageblock1	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	95.13 ± 3.4e−4	**95.55 ± 0.49**	75.41 ± 0.39	95.27 ± 0.45
Pageblock13vs2	93.33 ± 6.77e−17	100 ± 4.51e−17	83.67 ± 0.87	99.05 ± 1.65	100 ± 0	99.82 ± 0	99.98 ± 0	99.95 ± 0.12	99.99 ± 0	**100 ± 0**
Wine1	100 ± 4.51e−17	100 ± 0.92	99.43 ± 1.63	99.73 ± 0.2.42	**100 ± 1.67**	99.47 ± 0.95	96.00 ± 0.69	97.61 ± 0.57	98.32 ± 0.31	98.75 ± 0.28

1. When compared with all other all methods, the proposed method was determined to have achieved the best *G*-mean over 27 data sets; the best AUC over 20 data sets; and the best SN over 17 data sets;
2. When compared with the other WELM-based methods, the proposed method was found to achieve the best *G*-mean over 37 data sets; the best AUC over 36 data sets; and the best SN over 36 data sets. Apart from Glass2, Glass6, Ecoli1, Pima, Glass5, Ecoli2, Vehicle3, Yeast6, and Abalone9vs18, the SP of the proposed method was found to be increased significantly when compared to the WELM. These results indicated that the SDE-WELM was able to achieve a better classification performance for the minority classes and majority classes when compared with the performances of the WELM, DE-WELM, EN-WELM, and ABC-WELM for the majority of the data sets. Also, when compared with the WELM, the STD of *G*-mean and SN caused by the random generation of input parameters in the proposed method had been reduced, which indicated that the proposed method was more stable than WELM. In addition, the *G*-mean and AUC values of the DE-WELM, ABC-WELM and SDE-WELM were determined to be higher than those of the WELM over the majority of data sets, which indicated that using a stochastic search algorithm to find the optimal input weights, hidden nodes parameters, and training sample weights of the WELM achieved a better performance than WELM. Therefore, it was found that the SDE-WELM performed better than the original WELM, DE-WELM, and ABC-WELM in terms of producing higher testing AUC, *G*-mean, SN, and lower STD. These results supported the argument that, when chosen randomly, the input weights, and hidden node parameters of the WELM impact the classification performances, and the results also showed that using a SDE method to optimize the networks' input parameters will reduce the negative impacts of un-optimal parameters, and ensures the network has a better generalization performance;
3. When compared with ensemble methods which combine data sampling and Bagging or Boosting, the proposed method was found to have achieved the best *G*-mean over 27 data sets and the best SN over 17 data sets.

In this study, for the purpose of confirming whether or not there were significant differences between the proposed method and the other imbalanced learning methods, a Friedman test (recommend by the previous related reference [40]), along with corresponding post hoc tests, were used for a comparison of the many classifiers over the multiple data sets. The Friedman test [56,57] is a nonparametric test. It separately ranks algorithm in terms of their classification performances for each data set, with the best performing algorithm being ranked as 1, the second best being ranked 2, and so on. A null

Table 9 Average SP of the 40 imbalanced classification problems

Data sets	WSVM	OverBagging	UnderBagging	RUSBoost	EasyEnsemble	WELM	EN-WELM	DE-WELM	ABC-WELM	SDE-WELM
Glass0	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 ± 2.12	76.37 ± 1.80	76.43 ± 1.62
Glass1	85.83 ± 2.48e−16	65.67 ± 1.89	79.59 ± 2.42	61.77 ± 8.73	78.51 ± 5.56	76.79 ± 3.26	76.44 ± 3.50	76.68 ± 2.50	76.29 ± 2.03	76.79 ± 1.25
Glass2	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	82.21 ± 1.83	82.12 ± 1.89	81.26 ± 1.90
Glass4	98.96 ± 0	91.84 ± 4.51e−16	98.94 ± 0.10	94.40 ± 1.91	90.95 ± 4.38	88.31 ± 0.57	88.10 ± 0.39	88.10 ± 0.32	87.96 ± 0.29	87.03 ± 0.18
Glass5	98.54 ± 2.12e−16	85.41 ± 0	98.93 ± 0	96.90 ± 2.23	85.53 ± 5.57	74.63 ± 3.69	74.59 ± 1.57	74.15 ± 1.21	74.15 ± 1.09	73.96 ± 0.69
Glass6	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	98.74 ± 4.1e−17
Glass0123vs456	95.94 ± 2.03e−16	66.54 ± 3.83e−16	95.8 ± 0.51	92.33 ± 2.65	93.24 ± 1.88	96.60 ± 1.40e−16	95.14 ± 0.28	94.23 ± 0.15	95.12 ± 0.18	94.07 ± 0.15
Glass16vs2	98.46 ± 3.83s-16	59.41 ± 6.24	80.20 ± 0	62.75 ± 6.24	68.14 ± 7.89	76.15 ± 4.84	83.42 ± 4.84	85.36 ± 4.52	83.42 ± 4.20	82.96 ± 4.20
Glass16vs5	84.23 ± 2.03e−16	92.05 ± 2.58e−17	98.76 ± 2.91	97.72 ± 1.51	87.00 ± 6.24	97.86 ± 0.59	97.90 ± 0.42	97.23 ± 0.36	97.29 ± 0.41	96.18 ± 0.35
Ionosphere	85.58 ± 0	82.83 ± 0	79.30 ± 3.3e−16	91.18 ± 5.89	92.85 ± 2.71	98.30 ± 0.90	99.13 ± 0.62	98.74 ± 0	98.99 ± 0.89	99.29 ± 0.54
Ecoli0	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	97.37 ± 0	97.53 ± 0	97.95 ± 0
Ecoli1	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 ± 2.2e−16	84.86 ± 0	84.60 ± 0
Ecoli2	97.15 ± 4.06e−16	91.29 ± 3.16e−16	98.13 ± 4.55	90.63 ± 2.87	88.59 ± 2.47	94.37 ± 0.35	95.73 ± 0.22	93.25 ± 0.20	93.12 ± 0.50	93.97 ± 0.11
Ecoli3	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	89.14 ± 0.74	88.21 ± 0.53	89.04 ± 0.71
Ecoli4	97.68 ± 3.61e−16	94.78 ± 7.46e−4	99.76 ± 0	97.44 ± 4.88	94.93 ± 5.30	95.89 ± 0.43	93.78 ± 2.83e−16	93.75 ± 2.83e−16	94.21 ± 1.26e−16	93.78 ± 0
Ecoli0vs1	100 ± 1.35e−16	97.30 ± 1.12e−16	100 ± 0.15	95.81 ± 3.31	97.21 ± 4.90	98.46 ± 0.51	98.75 ± 0.24	97.89 ± 0.31	98.75 ± 0.11	9.98 ± 0.15
Ecoli0137vs26	95.94 ± 2.03e−16	94.95 ± 0.14	97.61 ± 0.11	93.28 ± 3.31	87.67 ± 2.07	96.78 ± 0.99	96.45 ± 1.24	96.45 ± 1.16	95.56 ± 1.03	95.56 ± 0.95
Vehicle0	97.94 ± 0	98.00 ± 0.36	98.23 ± 6.7e−16	92.58 ± 1.35	94.80 ± 0.96	97.63 ± 0.65	97.68 ± 0.50	97.57 ± 0.44	97.67 ± 0.43	97.99 ± 0.45
Vehicle1	88.98 ± 4.6e−16	77.14 ± 0.50	93.00 ± 0.34	68.55 ± 2.70	76.59 ± 1.38	83.63 ± 1.33	84.91 ± 1.04	84.69 ± 0.23	84.51 ± 0.21	84.47 ± 0
Vehicle2	99.07 ± 3.38e−16	96.31 ± 2.48e−4	99.55 ± 1.58e−16	68.78 ± 2.72	98.00 ± 0.90	98.50 ± 0.52	99.45 ± 0.22	99.87 ± 6.39e−4	99.68 ± 3.52e−4	99.87 ± 1.10e−4
Vehicle3	90.31 ± 1.80e−16	75.97 ± 0.50	93.47 ± 7.92e−4	91.87 ± 1.55	75.01 ± 1.75	85.40 ± 1.28	85.40 ± 0.96	85.40 ± 0	84.92 ± 0	84.92 ± 0
Segment1	99.95 ± 1.35e−16	99.70 ± 5.15e16	99.95 ± 0	99.61 ± 0.34	99.73 ± 0.18	99.67 ± 0.21	99.87 ± 0.17	99.77 ± 0	99.69 ± 0	99.67 ± 0
Wisconsin	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	96.96 ± 0.28	97.30 ± 0.16
Yeast1	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	70.99 ± 0.69	70.47 ± 0.65	70.66 ± 0.57
Yesat3	97.48 ± 5.4e−16	97.14 ± 0.22	99.58±	88.83 ± 1.62	90.25 ± 0.78	92.73 ± 0.29	93.25 ± 0.23	92.88 ± 0.20	93.07 ± 0.23	93.29 ± 0.20
Yesat4	99.52 ± 2.25e−16	84.06 ± 2.83e−4	100 ± 0	88.53 ± 1.71	79.41 ± 1.75	85.43 ± 1.12	85.88 ± 0.84	86.54 ± 0.23	85.88 ± 0.12	86.54 ± 0.20
Yesat5	99.16 ± 3.16e−16	94.22 ± 5.66e−4	99.93 ± 2.07e−4	96.86 ± 0.73	95.09 ± 0.88	93.33 ± 0.13	93.41 ± 0.16	93.72 ± 0.12	93.41 ± 016	93.13 ± 0.12
Yesat6	99.39 ± 1.15e−16	8639 ± 4.34e−4	99.93 ± 0	91.87 ± 1.55	84.13 ± 7.89	51.28 ± 0.31	91.44 ± 0.15	90.36 ± 0.58	90.18 ± 0.27	90.28 ± 0.14

Table 9 continued

Data sets	WSVM	OverBagging	UnderBagging	RUSBoost	EasyEnsemble	WELM	EN-WELM	DE-WELM	ABC-WELM	SDE-WELM
Yesat1vs7	99.77 ± 1.12e−16	76.06 ± 2.08	100 ± 9.03e−17	75.74 ± 4.18	68.91 ± 3.79	79.47 ± 0.80	79.44 ± 0.68	77.65 ± 0.54	79.44 ± 0.29	78.17 ± 0.37
Yesat2vs4	92.05 ± 1.80e−16	87.94 ± 1.40	94.77 ± 0.67	86.93 ± 1.40	85.25 ± 2.34	88.58 ± 0.69	88.96 ± 0.33	88.32 ± 9.7e−16	89.47 ± 3.45e−16	89.29 ± 2.48e−17
Yesat2vs8	99.94 ± 1.58e−16	75.86 ± 6.7e−17	95.11 ± 1.50e−16	9229 ± 4.19	73.09 ± 5.06	92.22 ± 5.24	98.82 ± 4.18	97.36 ± 0.56	98.15 ± 0.48	97.36 ± 0.12
Yeast05679vs4	99.08 ± 4.29-16	84.02 ± 0.38	99.42 ± 0	85.24 ± 1.35	80.77 ± 3.91	88.30 ± 1.30	88.69 ± 1.30	90.58 ± 0.75	88.69 ± 0.71	91.63±
Yesat1289vs7	99.45 ± 3.83e−16	73.30 ± 6.51e−4	99.84 ± 2.48e−16	79.76 ± 3.48	67.53 ± 3.20	78.50 ± 1.02	79.49 ± 0.44	79.63 ± 0.23	78.49 ± 0.21	79.68 ± 0.163
Yesat1458vs7	98.56 ± 3.38e−16	66.97 ± 5.70e−4	100 ± 0	75.0 ± 3.86	62.17 ± 4.29	72.66 ± 1.41	75.04 ± 0.97	79.36 ± 0.78	78.15 ± 0.28	76.390.36
Pima	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	77.78 ± 0.85	77.35 ± 0.73	77.24± “0.77
Abalone9vs18	73.01 ± 6.7e−16	100 ± 0	6.41 ± 3.3e−16	84.24 ± 2.69	86.37 ± 2.47	91.17 ± 0.75	91.08 ± 0.78	90.79 ± 0.53	9066 ± 0.27	90.71 ± 3.8e−16
Abalone19	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 ± 3.4e−4	77.71 ± 2.75	78.94 ± 2.15
Pageblock1	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 ± 6.9e−4	95.80 ± 0.15	95.87 ± 0.11
Pageblock13vs2	100 ± 0	94.72 ± 0	100 ± 0.11	99.18 ± 5.66	97.70 ± 1.76	98.46 ± 0.51	98.69 ± 0.4	99.31 ± 0.50	98.69 ± 0.23	98.13 ± 0.21
Wine1	96.19 ± 5.04e−16	95.95 ± 0.62	97.08 ± 1.22	96.38 ± 2.63	97.35 ± 2.33	93.75 ± 2.17	94.04 ± 2.14	93.87 ± 1.58	94.12 ± 1.27	94.04 ± 0.99

Table 10 Average ranks of the comparison in terms of *G*-mean among all of the methods

p value	WSVM	OverBagging	UnderBagging	RUSBoost	EasyEnsemble	WELM	EN-WELM	DE-WELM	ABC-WELM	SDE-WELM
8.9504e−39	8.6125	6.4875	8.25	6.975	5.4625	6.1	4.7625	3.9825	2.8625	1.6

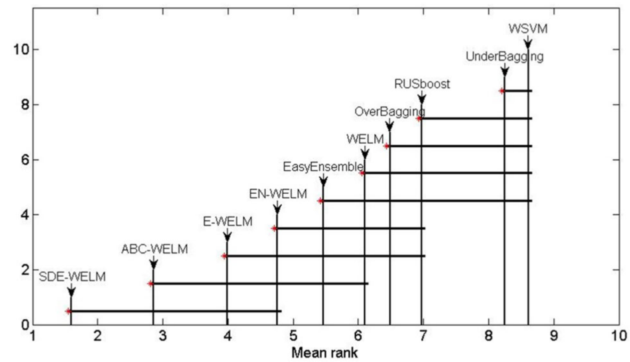

Fig. 3 Visualization of the pair-wise comparisons of the ten methods using a Nemenyi post hoc test in terms of *G*-mean values

hypothesis indicates that all the algorithms are equivalent, and therefore, their average ranks should all be equal. If a null hypothesis is rejected, a post hoc test named Nemenyi test performed to identify which method significantly performed the best, as suggested by reference [40]. A Nemenyi test [58] is used when all the classifiers are compared with each other. The performance of two algorithms will be considered to be significantly different if the corresponding average ranks of the two algorithms are significantly different. In this study, a Friedman test was used in terms of the *G*-mean value. The significance level (α) of the Friedman test was taken as 0.05. For each data set, the best method in terms of *G*-mean was ranked as 1, the second best as 2, and so on. The mean rank of each method was calculated over the 40 data sets. The null hypothesis, in which all the methods were considered to be equivalent, and there were no differences among their mean ranks, was rejected in this study if the p value was lower than the α. As can be seen in Table 10, the p value obtained *G*-mean value was 8.9504e−39, which meant that the Friedman test rejected the null hypothesis with a p value of 8.9504e−39, and significant difference existed among these imbalanced learning methods (The average rankings are shown in Table 10). Therefore, this study executed a post hoc test Nemenyi test in order to further analyze the differences among the 10 methods. The post hoc procedures assisted to in the determination of how different several of the algorithms were. Figure 3 visualizes the ten methods' mean ranks in terms of the *G*-mean using the Nemenyi test at $p = 0.05$. In the figure, the mean ranks over the 40 data sets are shown in ascending rank order on the *x*-axis. The respective mean rank of each method is denoted with a red '*'. The critical differences of each method are given by the line segment to its right. As shown in Fig. 3, the methods with '*' on the left end of the line segment were considered to have significantly outperformed, and the performance of the proposed method was confirmed to be better than the performances of the other methods.

Table 11 Average ranks of the comparison between all the methods in terms of the AUC values

p value	WSVM	OverBagging	UnderBagging	RUSBoost	EasyEnsemble	WELM	EN-WELM	DE-WELM	ABC-WELM	SDE-WELM
7.2357e−60	8.45	6.3	8.175	5.25	3.875	6.8	5.125	4.925	3.525	2.3

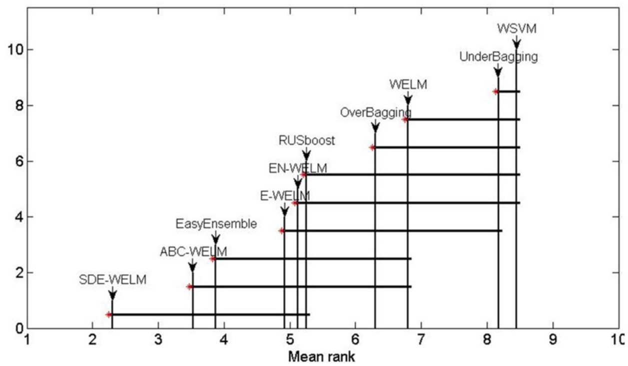

Fig. 4 Visualization of the pair-wise comparisons of the ten methods using a Nemenyi post hoc test in terms of the AUC values

In addition, the results of the Friedman test in terms of AUC values are also provided in Table 11. Figure 4 visualizes the ten methods' mean ranks in terms of the AUC values using a Nemenyi test at $p = 0.05$. As can be seen in Table 11, the p value obtained AUC value was 7.2357e−60. Therefore, the null hypothesis was rejected, which meant that the Friedman test rejected the null hypothesis with a p value of 7.2357e−60, and significant differences existed among the methods. Then, this study executed a post hoc Nemenyi test in order to further analyze the differences among the 10 methods. As detailed in Tables 10, 11 and Figs. 3 and 4, it was concluded that significant differences existed between the proposed method and the other imbalanced learning methods.

5 Conclusions

In this research study, a the self-adaptive differential evolutionary weighted extreme learning machine (SDE-WELM) for dealing with the binary-class imbalance problems was presented, in which a self-adaptive differential evolution was used to optimize the input weights, hidden biases, and training sample weights of a WELM. An important point of the SDE-WELM was selection of the fitness function, which was then applied to evaluate all of the populations when an error between the network approximated output and the expected output caused the classification performance to be biased toward the majority classes. Then, the most appropriate metric was selected as the fitness function, which was then applied to evaluate all the populations in the process of selecting the input weights, hidden biases, and training sample weights. A total of 40 experiments were conducted in this study to demonstrate that the fitness functions (AUC metric) were effective for these tasks. It was also confirmed that the proposed algorithm, which was found to more stable, had a better classification performance of minority classes when compared with the WELM, DE-WELM, ABC-WELM, and EN-WELM algorithms. This study also expects that the proposed SDE-WELM can be effectively applied in the future to real-world imbalanced data mining applications, such as identification of protein functional sites in future.

Acknowledgements This study was funded by the National Nature Science Foundation of China (Nos. 61379010; 11561054)

Compliance with ethical standards

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

Ethical standard This article does not contain any studies involving human participants or animals which were performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in this research study.

References

1. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: theory and applications. Neurocomputing **70**(1), 489–501 (2006)
2. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: a new learning scheme of feedforward neural networks. In: Proceedings. IEEE International Joint Conference on Neural Networks, 2004, vol. 2, pp. 985–990. IEEE, Washington (2004)
3. Huang, G.B., Zhou, H., Ding, X., Zhang, R.: Extreme learning machine for regression and multiclass classification. IEEE Trans. Syst. Man Cybern. Part B Cybern. **42**(2), 513–529 (2012)
4. Huang, G., Huang, G.B., Song, S., You, K.: Trends in extreme learning machines: a review. Neural Netw. **61**, 32–48 (2015)
5. Wong, K.I., Vong, C.M., Wong, P.K., Luo, J.: Sparse Bayesian extreme learning machine and its application to biofuel engine performance prediction. Neurocomputing **149**, 397–404 (2015)
6. Peng, Y., Wang, S., Long, X., Lu, B.L.: Discriminative graph regularized extreme learning machine and its application to face recognition. Neurocomputing **149**, 340–353 (2015)
7. Choi, K., Toh, K.A., Byun, H.: Incremental face recognition for large-scale social network services. Pattern Recogn. **45**(8), 2868–2883 (2012)
8. Lan, Y., Hu, Z., Soh, Y.C., Huang, G.B.: An extreme learning machine approach for speaker recognition. Neural Comput. Appl. **22**(3–4), 417–425 (2013)
9. 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)
10. Termenon, M., Graña, M., Barrós-Loscertales, A., Ávila, C.: Extreme learning machines for feature selection and classification of cocaine dependent patients on structural MRI data. Neural Process. Lett. **38**(3), 375–387 (2013)
11. Xu, Y., Dong, Z.Y., Zhao, J.H., Zhang, P., Wong, K.P.: A reliable intelligent system for real-time dynamic security assessment of power systems. IEEE Trans. Power Syst. **27**(3), 1253–1263 (2012)
12. Song, Y., Crowcroft, J., Zhang, J.: Automatic epileptic seizure detection in EEGs based on optimized sample entropy and extreme learning machine. J. Neurosci. Methods **210**(2), 132–146 (2012)
13. Alexandre, E., Cuadra, L., Salcedo-Sanz, S., Casanova-Mateo, C.: Hybridizing extreme learning machines and genetic algorithms to select acoustic features in vehicle classification applications. Neurocomputing **152**(C), 58–68 (2015)
14. Zong, W., Huang, G.B., Chen, Y.: Weighted extreme learning machine for imbalance learning. Neurocomputing **101**, 229–242 (2013)

15. 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)
16. Fan, Y.X., Shen, H.B.: Predicting pupylation sites in prokaryotic proteins using pseudo-amino acid composition and extreme learning machine. Neurocomputing **128**(5), 267–272 (2014)
17. Meyer, I.M.: A practical guide to the art of rna gene prediction. Brief. Bioinform. **8**(6), 396 (2007)
18. Czarnecki, W.M.: Weighted tanimoto extreme learning machine with case study in drug discovery. IEEE Comput. Intell. Mag. **10**(3), 19–29 (2015)
19. Saraswathi, S., Sundaram, S., Sundararajan, N., Zimmermann, M., Nilsenhamilton, 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. Bioinform. **8**(2), 452–463 (2011)
20. 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)
21. He, H., García, E.A.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. **21**(9), 1263–1284 (2009)
22. 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 Syst. Hum. **40**(1), 185–197 (2010)
23. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory undersampling for class-imbalance learning. IEEE Trans. Syst. Man Cybern. Part B Cybern. **39**(2), 539–550 (2009)
24. Barandela, R., Valdovinos, R.M., Sánchez, J.S.: New applications of ensembles of classifiers. Pattern Anal. Appl. **6**(3), 245–256 (2003)
25. Barandela, R., Sánchez, J.S., Garcıa, V., Rangel, E.: Strategies for learning in class imbalance problems. Pattern Recognit. **36**(3), 849–851 (2003)
26. Xu, L., Chow, M.Y., Taylor, L.S.: Power distribution fault cause identification with imbalanced data using the data mining-based fuzzy classification e-algorithm. IEEE Trans. Power Syst. **22**(1), 164–171 (2007)
27. Toh, K.A.: Deterministic neural classification. Neural Comput. **20**(6), 1565–1595 (2008)
28. Deng, W., Zheng, Q., Chen, L.: Regularized extreme learning machine. In: IEEE Symposium on Computational Intelligence and Data Mining, 2009. CIDM ’09, pp. 389–395. IEEE, Washington (2009)
29. Xia, S.X., Meng, F.R., Liu, B., Zhou, Y.: A kernel clustering-based possibilistic fuzzy extreme learning machine for class imbalance learning. Cogn. Comput. **7**(1), 74–85 (2015)
30. Liu, N., Wang, H.: Ensemble based extreme learning machine. IEEE Signal Process. Lett. **17**(8), 754–757 (2010)
31. Cao, J., Lin, Z., Huang, G.B., Liu, N.: Voting based extreme learning machine. Inf. Sci. **185**(1), 66–77 (2012)
32. Sharma, R., Bist, A. S.: Genetic algorithm based weighted extreme learning machine for binary imbalance learning. In: International Conference on Cognitive Computing and Information Processing, pp. 1–6. IEEE, Washington (2015)
33. Li, K., Kong, X., Lu, Z., Liu, W., Yin, J.: Boosting weighted elm for imbalanced learning. Neurocomputing **128**(5), 15–21 (2014)
34. Garro, B.A., Sossa, H., Vázquez, R.A.: Artificial neural network synthesis by means of artificial bee colony (abc) algorithm. In: 2011 IEEE Congress of Evolutionary Computation (CEC), pp. 331–338. IEEE, Washington (2011)
35. Yao, X.: Evolving artificial neural networks. Proc. IEEE **87**(9), 1423–1447 (1999)
36. Zhu, Q.Y., Qin, A.K., Suganthan, P.N., Huang, G.B.: Evolutionary extreme learning machine. Pattern Recogn. **38**(10), 1759–1763 (2005)
37. Cao, J., Lin, Z., Huang, G.B.: Self-adaptive evolutionary extreme learning machine. Neural Process. Lett. **36**(3), 285–305 (2012)
38. Garro, B.A., Sossa, H., Vazquez, R.A.: Design of artificial neural networks using a modified particle swarm optimization algorithm. In: 2009 International Joint Conference on Neural Networks, pp. 938–945. IEEE (2009)
39. Japkowicz, N., Stephen, S.: The Class Imbalance Problem: A Systematic Study. IOS Press, Amsterdam (2002)
40. Ar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**(1), 1–30 (2006)
41. 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)
42. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341–359 (1997)
43. Subudhi, B., Jena, D.: Differential evolution and Levenberg Marquardt trained neural network scheme for nonlinear system identification. Neural Process. Lett. **27**(3), 285–296 (2008)
44. Ilonen, J., Kamarainen, J.K., Lampinen, J.: Differential evolution training algorithm for feed-forward neural networks. Neural Process. Lett. **17**(1), 93–105 (2003)
45. Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans. Evol. Comput. **13**(2), 398–417 (2009)
46. Karaboga, D., Akay, B., Ozturk, C.: Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks. In: International Conference on Modeling Decisions for Artificial Intelligence, pp. 318–329. Springer, Berlin (2007)
47. Xu, Y., Shu, Y.: Evolutionary extreme learning machine-based on particle swarm optimization. In: International Symposium on Neural Networks, pp. 644–652. Springer, Berlin (2006)
48. García, S., Herrera, F.: Evolutionary undersampling for classification with imbalanced datasets: proposals and taxonomy. Evol. Comput. **17**(3), 275–306 (2009)
49. Galar, M., Fernández, A., Barrenechea, E., Herrera, F.: EUSBoost: enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. Pattern Recognit. **46**(12), 3460–3471 (2013)
50. 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 Cybern. **42**(2), 406–421 (2012)
51. Lin, W.J., Chen, J.J.: Class-imbalanced classifiers for high-dimensional data. Brief. Bioinform. **14**(1), 13–26 (2013)
52. Lusa, L.: Class prediction for high-dimensional class-imbalanced data. BMC Bioinform. **11**(1), 523 (2010)
53. 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)
54. Asuncion, A., Newman, D.J.: UCI machine learning repository, Irvine, CA. Availiable online: http://www.ics.uci.edu/~mlearn/MLRepository.html
55. 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)
56. Friedman, M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J. Am. Stat. Assoc. **34**(205), 109–109 (1939)
57. Friedman, M.: A comparison of alternative tests of significance for the problem of *m* rankings. Ann. Math. Stat. **11**(1), 86–92 (1940)
58. Nemenyi, P.B.: Distribution-free multiple comparisons. PhD Thesis, Princeton University (1963)