Biomedical Signal Processing and Control 38 (2017) 400–410

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: www.elsevier.com/locate/bspc

Research paper

An adaptive kernel-based weighted extreme learning machine approach for effective detection of Parkinson's disease

Yang Wang[a,b,*], An-Na Wang[a], Qing Ai[a], Hai-Jing Sun[a]

[a] College of Information Science and Engineering, Northeastern University, Shenyang 110819, Liaoning, China
[b] School of Computer and Communication Engineering, Liaoning Shihua University, Fushun 113001, Liaoning, China

ARTICLE INFO

Article history:
Received 28 February 2017
Received in revised form 24 May 2017
Accepted 30 June 2017

Keywords:
Parkinson's disease
Imbalanced data
Extreme learning machine
Artificial bee colony
Feature selection

ABSTRACT

Imbalanced data appear in many real-world applications, from biomedical application to network intrusion or fraud detection, etc. Existing methods for Parkinson's disease (PD) diagnosis are usually more concerned with overall accuracy (ACC), but ignore the classification performance of the minority class. To alleviate the bias against performance caused by imbalanced data, in this paper, an effective method named AABC-KWELM has been proposed for PD detection. First, based on a fast classifier extreme learning machine (ELM), weighted strategy is used for dealing with imbalanced data and non-linear mapping of kernel function is used for improving the extent of linear separation. Furthermore, both binary version and continuous version of an adaptive artificial bee colony (AABC) algorithm are used for performing feature selection and parameters optimization, respectively. Finally, PD data set is used for evaluating rigorously the effectiveness of the proposed method in accordance with specificity, sensitivity, ACC, G-mean and F-measure. Experimental results demonstrate that the proposed AABC-KWELM remarkably outperforms other approaches in the literature and obtains better classification performance via 5-fold cross-validation (CV), with specificity of 100%, sensitivity of 98.62%, ACC of 98.97%, G-mean of 99.30%, and F-measure of 99.30%.

1. Introduction

After Alzheimer's disease, Parkinson's disease (PD) has become the second most common degenerative diseases of the central nervous system now. Because of the loss of dopamine-producing brain cell, patients with PD (PWP) are generally characterized by motor system disorders including bradykinesia, rigidity, tremor, and posture instability [1]. PD has influenced most of worldwide population, and the disease prevalence is increasing dramatically as people live longer. However, the cause of the disease has still unknown. It is reported that it is possible to alleviate symptoms remarkably in the early diagnosis of PD [2]. Therefore, the early diagnosis and treatment of PD is crucial. Research has shown that about 90% of PWP exhibit vocal impairment symptom known as dysphonia [3]. Dysphonia measurements have been a reliable diagnostic tool for PD.

Many researchers have handled PD diagnosis problem based on various machine learning techniques. Little et al. [4] has conducted a remarkable research on dysphonia measurements to discriminate healthy people from PWP. They employed kernel support vector machine (SVM) with feature selection method to detect PD. The classification accuracy of 91.4% was achieved using only four dysphonic features. Das [5] presented a comparative study of Decision Trees, Regression, DMneural and Neural Networks (NN) to detect PD. NN classifier yielded the best classification accuracy of 92.9%. Guo et al. [6] proposed the combination of genetic programming and expectation maximization algorithm for detecting PD. It improved data representation by creating feature functions and achieved the classification accuracy of 93.1%. Ozcift and Gulten [7] constructed 30 classifier ensembles based on rotation forest (RF) in combination with correlation based feature selection method for detecting PD, and it produced average classification accuracy of 87.13%. Aström and Koker [8] used a parallel feed-forward neural network structure to reduce the possibility of decision with error, and it achieved the classification accuracy of 91.20%. Luukka [9] introduced fuzzy entropy measures based feature selection method with similarity classifier for detecting PD. Mean classification accuracy of 85.03% was obtained using only two dysphonic features. Li et al. [10] used fuzzy-based transformation method to increase the information available with SVM for detecting PD, and it yielded the classification accuracy of 93.47%. Polat [11] introduced fuzzy

* Corresponding author at: School of Computer and Communication Engineering, Liaoning Shihua University, Fushun 113001, Liaoning, China.
E-mail address: wangyang0531@163.com (Y. Wang).

http://dx.doi.org/10.1016/j.bspc.2017.06.015
1746-8094/

c-means clustering-based feature weighting method to increase the distinguishing performance between classes. It combined *k*-nearest neighbor (KNN) classifier to detect PD and achieved the classification accuracy of 97.93%. Chen et al. [12] combined fuzzy *k*-nearest neighbor (FKNN) with the principle component analysis to construct the most discriminative features for detecting PD, and it obtained the classification accuracy of 96.07%. Zuo et al. [13] constructed an automatic diagnostic system to detect PD. Mean classification accuracy of 97.47% was obtained by using an adaptive FKNN approach based on particle swarm optimization (PSO). Gök [14] developed a discriminative model which applied RF ensemble KNN classifier algorithm, and it achieved the classification accuracy of 98.46%. From these works, existing methods has integrated feature reduction method with the efficient classifiers to further improve the performance of PD diagnosis. However, they are more concerned with overall accuracy (ACC) and designed based on the assumption that the size of each class is relatively balanced. Therefore, these methods ignore the minority class and tend to be biased against the majority class in dealing with imbalanced data [15,16]. In other words, they may achieve higher misclassification accuracy of the minority class than that of the majority class.

There are two methods dealing with imbalanced data i.e. resampling technique and algorithmic technique [17]. Resampling technique includes oversampling which duplicates some minority class samples randomly or creates new samples in the neighborhood of minority class samples and undersampling which removes some majority class samples randomly to balance the size of each class [18,19]. In the algorithmic technique, cost-sensitive learning method is widely used to cope with imbalanced data [20]. It assigns a different misclassification cost for each sample. Generally, minority class samples are assigned high misclassification cost, while majority class samples are assigned low misclassification cost to improve the classification performance. In this study, cost-sensitive learning method is of particular interest.

Very recently, extreme learning machine (ELM) [21,22] has achieved the excellent performance on the disease diagnosis problems, for instance, hepatitis disease diagnosis [23] and PD diagnosis [24]. In this study, kernel-based weighted extreme learning machine (KWELM) is presented to perform PD diagnosis. Weighted strategy is used to alleviate the bias against performance caused by imbalanced data. An extra weight is designed for each sample to strengthen the relative impact of the minority class. In addition, the kernelized version of ELM [25] is also applied, and its advantage is that only kernel parameter γ and penalty parameter C need to be adjusted.

Previous studies [4,9–14] for PD detection have proven that performing feature selection prior to classification can improve the classification accuracy. Artificial bee colony (ABC) [26] is one of the newest global optimization techniques. In view of its simplicity and robustness, ABC has successfully been used to handle various real-world optimization problems [27–29]. However, ABC is good at exploration but poor at exploitation of solutions researching [30]. Consequently, new dynamic search strategies to generate candidate solutions are proposed in order to balance exploitation and exploration. In this study, both binary version and continuous version of an adaptive artificial bee colony (AABC) are used for performing feature selection and parameters optimization, respectively. Binary AABC is used to identify the most informative features as a feature selection method. Then the reduced feature subsets are used as the input of the trained KWELM classifier whose kernel parameter γ and penalty parameter C are specified by continuous AABC. Finally, experimental results illustrate that the proposed method named AABC-KWELM is effective and robust on PD diagnosis problem.

The rest of this paper is organized as follows. In Section 2, relevant preliminary knowledge is reviewed briefly and new strategies are proposed. Section 3 describes the detailed implementations of the proposed method. In Section 4, the experimental design is described and many experiments are completed to demonstrate that the proposed method presents better performance than that achieved by some existing methods. Finally, conclusions are summarized in Section 5.

2. Preliminaries

In this section, ELM and ABC algorithms are reviewed briefly and new strategies based on them are presented.

2.1. Kernel-based weighted extreme learning machine (KWELM)

ELM is a kind of single-hidden layer feed-forward neural networks (SLFNs). Given a training data set consisting of N arbitrary samples (x_j, t_j), where $t_j = [t_{j1}, t_{j2}, \cdots, t_{jm}]^T \in R^m$ and $x_j = [x_{j1}, x_{j2}, \cdots, x_{jn}]^T \in R^n$. The jth sample t_j is an $m \times 1$ target vector, and x_j is an $n \times 1$ feature vector. Given hidden nodes $L \ll N$ and activation function g(x), then the standard mathematical model of SLFNs is as follows:

$$\sum_{i=1}^{L} \beta_i g(a_i \cdot x_j + b_i) = t_j, j = 1, 2, \cdots, N \tag{1}$$

where $\beta_i = [\beta_{i1}, \beta_{i2}, \cdots, \beta_{im}]^T$ is the output weight vector connecting the ith hidden node and the output nodes, $a_i = [a_{i1}, a_{i2}, \cdots, a_{in}]^T$ is the input weight vector connecting input nodes and the ith hidden node, $a_i \cdot x_j$ is the inner product of a_i and x_j, and b_i is the bias of the ith hidden node.

SLFNs can approximate the training samples with zero error if the number of hidden nodes L is equal to the number of training samples N. Eq. (1) can compactly be reformulated as

$$H\beta = T \tag{2}$$

$$H = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} g(a_1 \cdot x_1 + b_1) & \cdots & g(a_L \cdot x_1 + b_L) \\ \vdots & \cdots & \vdots \\ g(a_1 \cdot x_N + b_1) & \cdots & g(a_L \cdot x_N + b_L) \end{bmatrix}_{N \times L} \beta$$

$$= \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m}, \quad \text{and } T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m} \tag{3}$$

where H is the hidden layer output matrix, and the jth column of H represents the jth hidden node output vector on all the inputs. T is the output matrix, and β is the output weight matrix.

However, in most cases, it is $L \ll N$ and there may not exist a β that satisfies Eq. (2). The hidden layer biases and input weights need not be tuned at all and can be randomly generated, so the output weights can be determined by finding the Least Square solution $\hat{\beta} = H^+T$ of $H\beta = T$, where H^+ is the Moore- Penrose generalized inverse of matrix H. In short, ELM algorithm is summarized as follows.

1) Generate randomly input weights a_i and biases b_i, $i = 1, 2, \ldots, L$.
2) Calculate the hidden layer output matrix H.
3) Calculate the output weight $\beta = H^+T$.

According to Bartlett's theory [31], not only is ELM to minimize the training error but also the norm of the output weights. Meanwhile, an extra weight is designed for each sample to bet-

ter deal with imbalanced data, so the classification problem can be formulated as

$$min(\frac{1}{2}\|\beta\|^2 + CW\frac{1}{2}\sum_{j=1}^{N}\xi_j^2),$$

$$s.t.\sum_{j=1}^{N}\beta_i g(a_i \cdot x_j + b_i) = t_j - \xi_j \quad (4)$$

The equivalent dual optimization problem in regard to Eq. (4) based on KKT theorem is

$$L_{ELM} = \frac{1}{2}\|\beta\|^2 + CW\frac{1}{2}\sum_{j=1}^{N}\xi_j^2 - \sum_{j=1}^{N}\alpha_j(\beta_i g(a_i \cdot x_j + b_i) - t_j + \xi_j) \quad (5)$$

where ξ_j is the training error, C is penalty parameter, and α_j is Lagrange multiplier. W is diagonal matrix relevant to each training sample, namely $W = diag(w_{jj}), j = 1, 2, \cdots, N$. For instance, weighted strategy associated with the number of samples in the class can be assigned as follows [15]:

$$W1 : w_{jj} = \frac{1}{count(t_j)} \quad (6)$$

where $count(t_j)$ is the number of samples in class t_j. Then solution of Eq. (5) can be formulated as

$$\beta = H^T(\frac{I}{C} + WHH^T)^{-1}WT, \quad (7)$$

accordingly, the output function is

$$f(x) = h(x)\beta = h(x)H^T(\frac{I}{C} + WHH^T)^{-1}WT \quad (8)$$

Furthermore, in this study, a new weighted strategy is presented as

$$W2 : w_{jj} = 1 - \frac{count(t_j)}{N}. \quad (9)$$

W2-based weighted strategy has enormous influence on the classification performance. The reason is as follows:

$$\Delta W2 = \left(1 - \frac{count\ (minority)}{N}\right) - \left(1 - \frac{count\ (majority)}{N}\right)$$

$$= \frac{count\,(majority) - count\ (minority)}{N}$$

$$= \frac{count\ (majority) - count\ (minority)}{count\,(majority) + count\ (minority)}$$

$$\Delta W1 = \left(\frac{1}{count\,(minority)}\right) - \left(\frac{1}{count\,(majority)}\right)$$

$$= \frac{count\,(majority) - count\ (minority)}{count\,(majority) \cdot count\ (minority)}$$

where

$$count\ (majority) \cdot count\ (minority) - (count\ (majority) + count\ (minority)) = (count\,(majority) - 1) \cdot (count\,(minority) - 1) - 1$$

For $count(majority) > 2$ and $count(minority) > 2$, so $count\ (majority) - 1 > 1$ and $count(minority) - 1 > 1$, namely $(count\ (majority) - 1) \cdot (count(minority) - 1) > 1$. Therefore, $\Delta W2 > \Delta W1$ and *W2*-based weighted strategy has better performance than *W1*-based weighted strategy.

If $h(x)$ is unknown to users [25], a kernel matrix can be defined as

$$\Omega_{ELM} = HH^T : \Omega_{ELMi,j} = h(x_i) \cdot h(x_j) = K(x_i, x_j)$$

where $h(x)$ maps the data from the input space to the hidden layer feature space H [30]. Eq. (8) can compactly be reformulated as

$$f(x) = h(x)H^T(\frac{I}{C} + WHH^T)^{-1}WT = \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_N) \end{bmatrix}^T (\frac{I}{C} + W\Omega_{ELM})^{-1}WT \quad (10)$$

The $h(x)$ need not be known to users, but its corresponding kernel need be given to users. Meanwhile, the number of hidden nodes need not be given at all either. In this study, the Gaussian radial basis kernel function $K(x_1, x_2) = exp(-\gamma\|x_1 - x_2\|)^2$ is applied. Therefore, kernel parameter γ and penalty parameter C are crucial in model construction.

2.2. An adaptive artificial bee colony (AABC)

ABC is the global optimization technique which simulates the intelligent foraging behavior of honeybee swarms. In ABC, the colony of artificial bees consists of employed bees, onlookers, and scouts [32]. Employed bees take charge of searching available food sources and gathering required information, and then they share their food information with onlookers. The onlookers choose good food sources from those found by employed bees. When the quality of the food source is not improved over the predefined number of cycles, the food source is abandoned by its employed bee, then the corresponding employed bee becomes a scout and starts to randomly search for a new food source [33].

The position of each food source corresponds to a possible solution. In the initialization phase, ABC generates an initial population of *SN* solutions, and *SN* is the number of employed bees. Each initial solution $U_i = \{u_{i,1}, u_{i,2}, \cdots, u_{i,D}\}$ is produced randomly within the range of the boundaries of the parameters as follows:

$$u_{i,j} = u_{min,j} + rand(0, 1)\quad (u_{max,j} - u_{min,j}) \quad j = 1, 2, \cdots, D \qquad i = 1, 2, \cdots, SN \quad (11)$$

where D is the number of optimization parameters, then $u_{max,j}$ and $u_{min,j}$ are upper and lower bound for the jth parameter, respectively. Motivated by discrete PSO [34], binary ABC is used as a feature selection method. In binary ABC, the solution is transformed using a sigmoid function from continuous space to probability space,

$$sigmoid(u_{i,j}) = \frac{1}{1 + exp(-u_{i,j})} \quad (12)$$

$$u'_{i,j} = \begin{cases} 1, & if \quad rand(0, 1) < sigmoid(u_{i,j}) \\ 0, & otherwise \end{cases} \quad (13)$$

After the initialization, the population repeats cycles with the search processes of employed bees, onlookers, and scouts. In the employed bees phase, each employed bee performs a local search around each food source to generate a candidate position V_i:

$$v_{i,j} = u_{i,j} + \varphi_{i,j}\quad (u_{i,j} - u_{k,j}) \quad j \in \{1, 2, \cdots, D\} \qquad k \in \{1, 2, \cdots, SN\} \quad (14)$$

where j and k are randomly selected indices, and $k \neq i$. $\varphi_{i,j}$ is a random real number in [−1,1]. Then better food source is retained using the greedy selection between V_i and U_i. In binary ABC, the

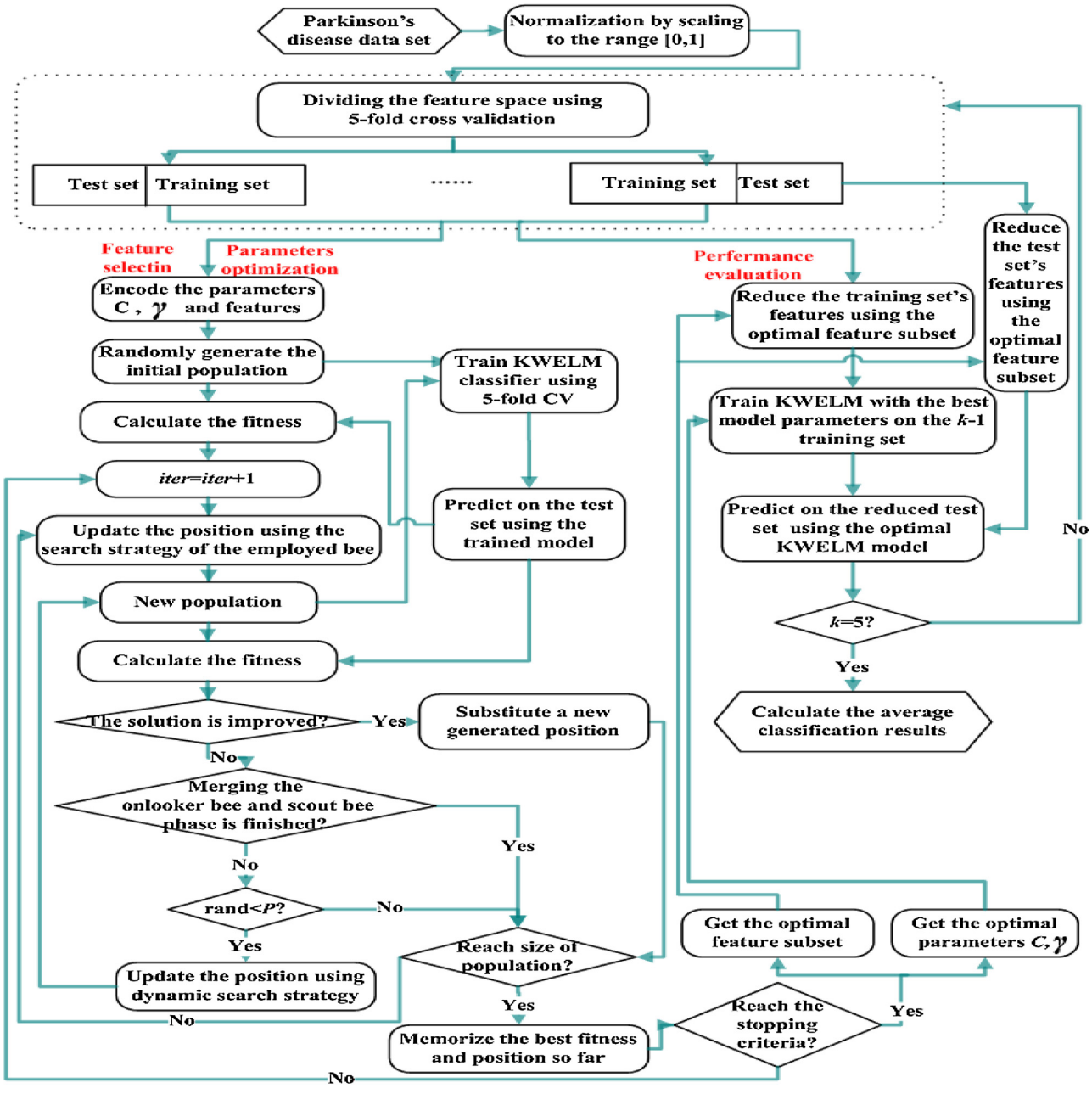

Fig. 1. The flowchart of the AABC-KWELM algorithm.

solution is transformed using a sigmoid function from continuous space to probability space,

$$sigmoid(v_{i,j}) = \frac{1}{1 + exp(-v_{i,j})} \tag{15}$$

$$v'_{i,j} = \begin{cases} 1, & if \quad rand(0,1) < sigmoid(v_{i,j}) \\ 0, & otherwise \end{cases} \tag{16}$$

then better food source is retained using the greedy selection between V'_i and U'_i.

In the onlookers phase, an onlooker selects a food source relying on p_i relevant to that food source,

$$p_i = \frac{f_i}{\sum_{j=1}^{SN} f_j} \tag{17}$$

where f_i denotes the fitness of solution U_i. As a matter of fact, food sources with better fitness are more possible to be selected and updated. A new food source position V_i is generated by Eq. (14) once the onlooker bee selects the food source, then the greedy selection is applied again. In binary ABC, a new food source position V'_i is generated by Eqs. (14)–(16), then the greedy selection is applied again.

In the scouts phase, if a food source is not improved over the predefined number of cycles, then the scout bee initializes a new food source randomly by Eq. (11). In binary ABC, the scout bee ini-

tializes a new food source randomly by Eqs. (11)–(13). In short, ABC algorithm is summarized as follows.

1) Initialization phase:

1.1) Set the maximum iteration *MCN*, the current iteration *iter* = 1, and generate randomly *SN* individual in the search space to form initial population.

1.2) Evaluate fitness of population.

2) While *iter* <= *MCN*, do

2.1) The employed bees phase.

2.2) The onlookers phase.

2.3) The scouts phase.

2.4) Memorize the best solution achieved so far, and update *iter* = *iter* + 1.

3) Output the best solution achieved.

However, ABC is good at exploration but poor at exploitation which results in poor convergence [30]. In this study, it is very necessary to search for a well improved optimization method named AABC. Inspired by differential evolution (DE) [35], a new search equation of employed bees is presented as follows:

$$v_{i,j} = u_{best,j} + \varphi_{i,j}(u_{best,j} - u_{k,j}) \tag{18}$$

where $u_{best,j}$ is the jth parameter of the current best solution. It is more likely to reach quickly the global optimum and improve further the exploitation capability with the guidance of Eq. (18). In binary AABC, in order to overcome the shortcomings of binary ABC, which is easy to cause new solutions the same as old solutions by single-dimensional updating strategy, multi-dimensional updating strategy is used,

$$v_{i,j} = \begin{cases} u_{best,j} + \varphi_{i,j}(u_{best,j} - u_{k,j}) & if \quad rand(0,1) < R \quad or \quad j = j_{rand} \\ u_{i,j} & otherwise \end{cases} \tag{19}$$

where $j = j_{rand}$ is to ensure at least one dimension is updated, R is the updating probability in the range [0,1]. Then the solution is transformed by Eqs. (15) and (16).

Meanwhile, the onlookers and scouts phase will be merged and a new dynamic search equation is presented as follows:

$$v_{i,j} = u_{i,j} + \varphi_{i,j} \times F \times u_{i,j}$$
$$F = 1 - \frac{iter - 1}{MCN} \tag{20}$$

where *iter* and *MCN* represent the current and maximum number of iteration, respectively. The dynamic factor F is applied to control the convergence speed. The search equation begins with a wide range of local search, and the search range becomes smaller with the increase of iteration time. The search strategy can effectively avoid plunging into local optimum. In binary AABC, multi-dimensional updating strategy is also used,

$$v_{i,j} = \begin{cases} u_{i,j} \\ u_{i,j} \end{cases} + \varphi_{i,j} \times F \times u_{i,j}, \quad F = 1 - \begin{cases} \frac{iter - 1}{MCN} \\ otherwise \end{cases} \quad if \quad rand(0,1) < R$$
$$or\ j = j_{rand} \tag{21}$$

then the solution is also transformed by Eqs. (15) and (16).

3. Proposed AABC-KWELM model

The proposed AABC-KWELM method is described in this section. Fig. 1 shows the flowchart of the algorithm. Construction of the optimal model comprises three main procedures: feature selection, parameters optimization and performance evaluation.

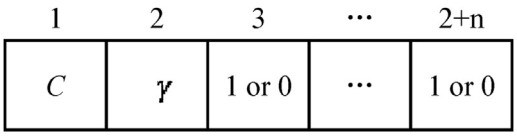

Fig. 2. Parameters representation.

Table 1
Description of PD data set.

Feature	Description
F1:MDVP:Fo (Hz)	Average vocal fundamental frequency
F2:MDVP:Fhi (Hz)	Maximum vocal fundamental frequency
F3:MDVP:Flo (Hz)	Minimum vocal fundamental frequency
F4:MDVP:Jitter (%)	
F5:MDVP:Jitter (Abs)	
F6:MDVP:RAP	Measures of variation in fundamental frequency
F7:MDVP:PPQ	
F8:Jitter:DDP	
F9:MDVP:Shimmer	
F10:MDVP:Shimmer(dB)	
F11:Shimmer:APQ3	Measures of variation in amplitude
F12:Shimmer:APQ5	
F13:MDVP:APQ	
F14:Shimmer:DDA	
F15:NHR	Measures of ratio of noise to tonal components in the voice
F16:HNR	
F17:RPDE	Nonlinear dynamical complexity measures
F18:D2	
F19:DFA	Signal fractal scaling exponent
F20:Spread1	Nonlinear measures of fundamental frequency variation
F21:Spread2	
F22:PPE	

3.1. Feature selection and parameters optimization

In this study, binary and continuous AABC are used for performing feature selection and parameters optimization, respectively. Without feature selection, two continuous parameters, C and γ, are required. Instead, $2+n$ parameters are required with feature selection. These n parameters values are 1 or 0, where 0 denotes the feature is discarded and 1 denotes the feature is selected. Fig. 2 shows parameters representation.

Usually ACC is used to measure the classification performance of PD diagnosis [4–14]. Unfortunately, the method is very sensitive to imbalanced data and can be misled to an extent. For instance, a binary classification problem consists of 99% negative class and 1% positive class. ACC of 99% is easily achieved by classifying all the samples as negative. However, the classification accuracy of the minority class is actually 0 [15]. To give more insight into the classification performance, G-mean is designed as fitness in this study:

$$f = \frac{\sum_{j=1}^{k} TestingG - mean_j}{k} \tag{22}$$

where f is average testing G-mean achieved by KWELM classifier via k-fold CV [36], where $k = 5$. Here the 5-fold CV is used for performing feature selection and parameters optimization. Binary AABC is used as a feature selection approach, then the reduced feature subsets are used as the input of the trained KWELM classifier whose kernel

parameter γ and penalty parameter C are specified by continuous AABC. The pseudo-code of this procedure is illustrated as follows:

Pseudo-code of feature selection and parameters optimization procedure

1) Initialize maximum iteration *MCN*, and *SN* individual by Eq.(11) for continuous parameters and by Eqs.(11), (12) and (13) for discrete parameters;
   Calculate fitness of individual, save the global optimal position as *global_position* and global optimal fitness as *global_fitness*;
2) While (*iter*<=*MCN*), do
   2.1) For *i*=1 : *SN*
      2.1.1) Update the personal position $\boldsymbol{v}_{i,j}$ by Eq.(18) for continuous parameters and by Eqs.(19), (15) and (16) for discrete parameters, then calculate the current fitness $\boldsymbol{f}(\boldsymbol{V}_i)$;
      2.1.2) If ($\boldsymbol{f}(\boldsymbol{V}_i) > \boldsymbol{f}(\boldsymbol{U}_i)$) set $\boldsymbol{U}_i = \boldsymbol{V}_i$;
      2.1.3) Otherwise, if (rand<*P*)
         Update the personal position $\boldsymbol{v}_{i,j}$ by Eq.(20) for continuous parameters and by Eqs.(21), (15) and (16) for discrete parameters, then calculate the current fitness $\boldsymbol{f}(\boldsymbol{V}_i)$;
      2.1.4) If ($\boldsymbol{f}(\boldsymbol{V}_i) > \boldsymbol{f}(\boldsymbol{U}_i)$) set $\boldsymbol{U}_i = \boldsymbol{V}_i$;
   2.2) Set [*index, maxfitness*]= max($\boldsymbol{f}(\boldsymbol{U}_i)$);
   2.3) If (*global_fitness* < *maxfitness*) set *global_fitness* = *maxfitness*, *global_position* = $\boldsymbol{U}_{index}$;
3) Return *global_position*, namely, get the best values of C ,γ and selected features.

3.2. Performance evaluation

KWELM classifier starts to carry on classification after feature subset and the optimal parameters are achieved. Firstly an optimal model is obtained by the trained KWELM using the optimal parameters on selected feature space. Then the optimal model is used for testing on selected feature space. Here the 5-fold CV is used for performing performance evaluation to gain an unbiased estimate. The pseudo-code of this procedure is given bellow:

Pseudo-code of performance evaluation procedure

1) For *i* = 1: *k*, do
   1.1) Setting *k*-1 reduced subsets as training set;
   1.2) Setting remaining reduced subset as testing set;
   1.3) Train the KWELM model using the obtained optimal parameters on the training set;
   1.4) Test the trained KWELM on the testing set;
2) Return the average classification results of KWELM on *i*th testing set.

4. Experiments

The experimental design is described in this section, and many experiments are completed to demonstrate that the proposed method presents better performance.

4.1. Data set description

The main aim of the experiment is to discriminate healthy people from PWP. In this study, PD data set is taken from UCI machine learning repository. This data set is composed of a range of biomedical voice measurements from 31 people, 23 with PD [37]. There are 195 subjects including 147 PD and 48 healthy cases in the data set, and the ages of the subjects range from 46 to 85 with an average age of 65.8. The whole 22 features are all real values and there are no missing values. Table 1 lists descriptions of all the features.

4.2. Experimental setting

The whole experiment is conducted on MATLAB platform, which runs on windows 8 OS with Intel(R) Core(TM) i5-4460 CPU (3.2 GHz) and 8 GB of RAM. The parameters setting for AABC-KWELM is given as follows. The size of population is set to 40, and the maximum iteration is set to 80. The searching ranges for kernel parameter γ and penalty parameter C are [1,100]. The value of the probability P is 0.7, and the value of the updating probability R is 0.4.

Normalization is applied prior to classification in order to avoid feature values of more difference. In this study, the data are scaled into the interval of [0,1] by Eq. (23), where x is the original value, x' is the scaled value, $\max_a$ is the maximum value of feature a, and $\min_a$ is the minimum value of feature a [36].

$$x' = \frac{x - \min_a}{\max_a - \min_a} \tag{23}$$

Table 2
The confusion matrix.

	Predicted	
Actual	PWP	Healthy people
PWP	TP	FN
Healthy people	FP	TN

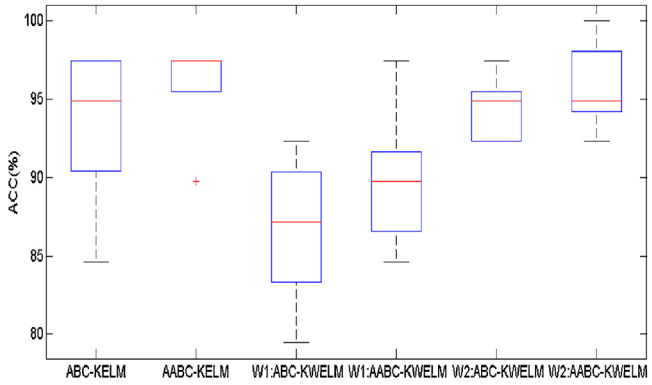

Fig. 3. The box plot representation of six algorithms without feature selection.

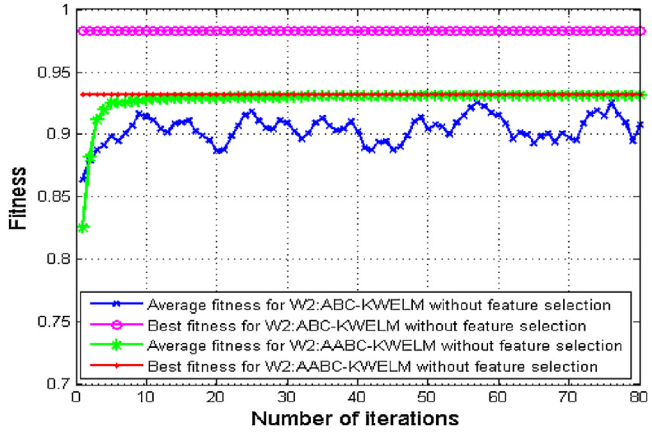

Fig. 4. The best and average fitness obtained by W2: ABC-KWELM and W2: AABC-KWELM without feature selection for fold #1.

In the experiment, a double loop for 5-fold CV is adopted. The inner loop is used for determining the best feature subset and the optimal parameters for KWELM classifier, while the outer loop is used for performing the performance evaluation of KWELM classifier. In order to ensure the reliability of the results, stratified sampling is used in the data set.

4.3. Measure metrics

Specificity, sensitivity, ACC, G-mean and F-measure are used for evaluating the performance of the proposed AABC-KWELM model. According to the confusion matrix which is presented in Table 2, these measure metrics are defined bellow.

$$G-mean=\sqrt{\frac{TP}{TP+FN}\times\frac{TN}{TN+FP}}\times 100\% \tag{24}$$

$$ACC=\frac{TP+TN}{TP+FP+TN+FN}\times 100\% \tag{25}$$

$$Sensitivity=\frac{TP}{TP+FN}\times 100\% \tag{26}$$

$$Specificity=\frac{TN}{TN+FP}\times 100\% \tag{27}$$

$$Precision=\frac{TP}{TP+FP} \tag{28}$$

$$Recall=\frac{TP}{TP+FN} \tag{29}$$

$$F-measure=\frac{2\times Recall\times Precision}{Recall+Precision}\times 100\% \tag{30}$$

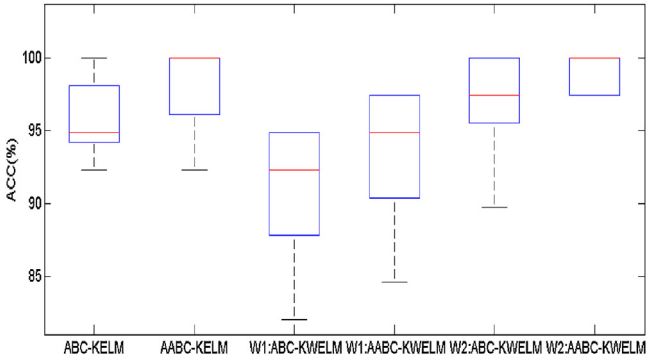

Fig. 5. The box plot representation of six algorithms with feature selection.

As shown in Table 2, TP is the number which PWP are correctly classified as PWP, and FN is the number which PWP are misclassified as healthy people. TN is the number which some healthy people are correctly classified as healthy people, and FP is the number which some healthy people are misclassified as PWP. Obviously, G-mean is 0 when the classification accuracy for the minority class is 0 [16]. Therefore, G-mean is the evaluation measure for imbalanced data and makes a more fair comparison. F-measure is usually used to assess the performance of binary classifier, which makes more emphasis on the evaluation of minority class.

4.4. Experiment 1: unweighted and weighted classification without feature selection

In this experiment, unweighted [25] and weighted classification performance without feature selection is evaluated, respectively. To show the effectiveness of the proposed algorithm, we also compare our W2-based weighted learning algorithm with W1-based weighted learning algorithm [15]. ABC-KWELM adopts ABC to find penalty parameter C and kernel parameter γ, while AABC-KWELM adopts continuous AABC to find the optimal parameters. Tables 3–5 show the detailed results achieved via 5-fold CV. From Table 3, we can see that unweighted strategy has the bias against the majority class and ignores the minority class. The reason is that the algorithm is based on the assumption that the size of each class is relatively balanced. From Table 4, we can see that the classification accuracy for the minority class is on the increase at the cost of the decrease for the majority class by W1-based weighted strategy. However, from Table 5, it is obvious to observe that applying W2-based weighted strategy has achieved relatively balanced results between the minority class and the majority class. This phenomenon illustrates the effectiveness of W2-based weighted strategy in dealing with imbalanced data.

From Table 5, we can see that W2-based ABC-KWELM has achieved average results (Mean) of 93.96%, 94.59%, 94.36%, 94.25% and 96.20% in accordance with specificity, sensitivity, ACC, G-mean and F-measure, respectively. Compared with it, W2-based AABC-KWELM has achieved the best results of 94.33%, 96.69%, 95.90%, 95.45% and 97.25% in accordance with specificity, sensitivity, ACC, G-mean and F-measure, respectively. From the results it can be concluded that the performance of AABC outperforms ABC. The reason is that the optima of C and γ can always be adaptively specified by continuous AABC algorithm for each fold of the data. The standard deviation (SD) acquired by AABC-based optimization algorithm is also relatively much smaller than that acquired by ABC-based optimization algorithm. In addition, Fig. 3 depicts the box plot to show ACC results of six algorithms. From the results it can be seen that dispersion degree of W2-based AABC-KWELM is relatively low, which indicates the robustness and stability of the proposed model.

Table 3
The detailed results obtained by unweighted classification without feature selection.

	ABC-KELM					
Fold	(C, γ)	G-mean (%)	ACC (%)	Sensitivity (%)	Specificity (%)	F-measure (%)
#1	(56.22,1)	91.08	92.31	93.33	88.89	94.92
#2	(26.78,1)	98.43	97.44	96.88	100	98.41
#3	(51.18,1.23)	88.12	94.87	97.06	80.00	97.06
#4	(79.32,3.31)	75.59	84.62	100	57.14	89.29
#5	(10.21,1)	96.08	97.44	100	92.31	98.11
Mean ± SD	(44.74,1.51) ± (26.85,1.01)	89.86 ± 8.95	93.33 ± 5.32	97.45 ± 2.76	83.67 ± 16.47	95.56 ± 3.76
	AABC-KELM					
Fold	(C, γ)	G-mean (%)	ACC (%)	Sensitivity (%)	Specificity (%)	F-measure (%)
#1	(30.44,1)	97.18	97.44	100	94.44	97.67
#2	(46.05,2.07)	92.58	97.44	100	85.71	98.46
#3	(79.35,1)	90.27	89.74	88.89	91.67	92.31
#4	(50.09,1)	93.54	97.44	100	87.50	98.41
#5	(33.46,7.87)	81.65	97.44	100	66.67	98.63
Mean ± SD	(47.88,2.59) ± (19.43,2.99)	91.04 ± 5.81	95.90 ± 3.44	97.78 ± 4.97	85.20 ± 10.91	97.10 ± 2.70

Table 4
The detailed results obtained by W1 weighted classification without feature selection.

	W1:ABC-KWELM					
Fold	(C, γ)	G-mean (%)	ACC (%)	Sensitivity (%)	Specificity (%)	F-measure (%)
#1	(1,18.05)	80.68	79.49	76.92	84.62	83.33
#2	(85.13,1)	88.08	87.18	86.21	90.00	90.91
#3	(88.58,1)	93.54	89.74	87.50	100	93.33
#4	(60.44,1)	86.07	84.62	83.33	88.89	89.29
#5	(52.94,1)	94.87	92.31	90.00	100	94.74
Mean ± SD	(57.62,4.41) ± (35.18,7.62)	88.65 ± 5.77	86.67 ± 4.93	84.79 ± 5.01	92.70 ± 6.96	90.32 ± 4.44
	W1:AABC-KWELM					
Fold	(C, γ)	G-mean (%)	ACC (%)	Sensitivity (%)	Specificity (%)	F-measure (%)
#1	(92.43,1.13)	93.54	89.74	87.50	100	93.33
#2	(94.22,1)	89.83	89.74	89.66	90.00	92.86
#3	(91.92,1)	90.75	84.62	82.35	100	90.32
#4	(90.56,1)	96.08	97.44	100	92.31	98.11
#5	(96.18,1)	86.52	87.18	88.46	84.62	90.20
Mean ± SD	(93.06,1.03) ± (2.18,0.06)	91.34 ± 3.65	89.74 ± 4.80	89.59 ± 6.45	93.38 ± 6.65	92.96 ± 3.21

Table 5
The detailed results obtained by W2 weighted classification without feature selection.

	W2:ABC-KWELM					
Fold	(C, γ)	G-mean (%)	ACC (%)	Sensitivity (%)	Specificity (%)	F-measure (%)
#1	(82.24,1.40)	98.26	97.44	96.55	100	98.25
#2	(94.68,1)	91.54	92.31	93.10	90.00	94.74
#3	(29.65,1)	96.72	94.87	93.55	100	96.67
#4	(36.65,2.38)	91.08	92.31	93.33	88.89	94.92
#5	(83.43,1)	93.63	94.87	96.43	90.91	96.43
Mean ± SD	(65.33,1.36) ± (29.88,0.59)	94.25 ± 3.16	94.36 ± 2.15	94.59 ± 1.74	93.96 ± 5.56	96.20 ± 1.44
	W2:AABC-KWELM					
Fold	(C, γ)	G-mean (%)	ACC (%)	Sensitivity (%)	Specificity (%)	F-measure (%)
#1	(80.24,1)	93.22	94.87	96.55	90.00	96.55
#2	(91.21,1.32)	94.87	97.44	100	90.00	98.31
#3	(68.59,1)	93.95	94.87	96.30	91.67	96.30
#4	(89.01,1)	100	100	100	100	100
#5	(49.60,1)	95.20	92.31	90.63	100	95.08
Mean ± SD	(75.73,1.06) ± (17.10,0.14)	95.45 ± 2.66	95.90 ± 2.92	96.69 ± 3.84	94.33 ± 5.22	97.25 ± 1.92

To understand the evolutionary process of AABC-based and ABC-based optimization algorithm without feature selection, Fig. 4 presents the evolution of the average and best fitness obtained by W2-based weighted strategy for fold #1. In each generation, the average and best fitness are calculated based on the average position and the global best position of population. From this figure, it can be seen that the best fitness has no variation during the evolution. The average fitness for W2-based ABC-KWELM has also no significant variation from iteration 1–80. Nevertheless, the average fitness for W2-based AABC-KWELM increases rapidly in the beginning of the evolution, and it starts increasing slowly after iteration 8. Thereafter, the average fitness gradually improves and it keeps stable after iteration 31. This phenomenon illustrates that AABC algorithm can adjust efficiently the solutions and converge more quickly toward the global optima than ABC algorithm.

Furthermore, statistical testing is a meaningful way to study the difference among six algorithms. In this study, statistical software packages SPSS 19 is used to judge the difference by one-way anal-

Table 6
The detailed results obtained by unweighted classification with feature selection.

	ABC-KELM					
Fold	(C, γ)	G-mean (%)	ACC (%)	Sensitivity (%)	Specificity (%)	F-measure (%)	#S.F.
#1	(55.39,1.13)	96.49	94.87	93.10	100	96.43	14
#2	(64.34,1)	92.70	94.87	96.67	88.89	96.67	14
#3	(68.72,1.32)	95.35	97.44	100	90.91	98.25	18
#4	(26.89,1.30)	85.28	92.31	100	72.73	94.92	12
#5	(46.28,3.28)	100	100	100	100	100	11
Mean ± SD	(52.32,1.61) ± (16.63,0.94)	93.96 ± 5.52	95.90 ± 2.92	97.95 ± 3.07	90.51 ± 11.17	97.25 ± 1.94	13.8 ± 2.68
	AABC-KELM					
Fold	(C, γ)	G-mean (%)	ACC (%)	Sensitivity (%)	Specificity (%)	F-measure (%)	#S.F.
#1	(87.34,1)	100	100	100	100	100	14
#2	(49.02,93.10)	83.67	92.31	100	70.00	95.08	13
#3	(57.04,1)	100	100	100	100	100	13
#4	(24.05,1)	100	100	100	100	100	12
#5	(48.83,1)	98.20	97.44	96.43	100	98.18	14
Mean ± SD	(53.25,19.42) ± (22.73,41.19)	96.37 ± 7.15	97.95 ± 3.34	99.29 ± 1.60	94.00 ± 13.42	98.65 ± 2.15	13.2 ± 0.84

Table 7
The detailed results obtained by W1 weighted classification with feature selection.

	W1:ABC-KWELM					
Fold	(C, γ)	G-mean (%)	ACC (%)	Sensitivity (%)	Specificity (%)	F-measure (%)	#S.F.
#1	(61.56,1)	96.92	94.87	93.94	100	96.88	14
#2	(95.61,66.59)	93.33	89.74	87.10	100	93.10	11
#3	(13.11,1.86)	96.49	94.87	93.10	100	96.43	13
#4	(36.16,1)	92.52	92.31	91.30	93.75	93.33	9
#5	(73.89,97.94)	84.00	82.05	80.65	87.50	87.72	8
Mean ± SD	(56.07,33.68) ± (32.22,45.72)	92.65 ± 5.20	90.77 ± 5.32	89.22 ± 5.47	96.25 ± 5.59	93.49 ± 3.66	11 ± 2.55
	W1:AABC-KWELM					
Fold	(C, γ)	G-mean (%)	ACC (%)	Sensitivity (%)	Specificity (%)	F-measure (%)	#S.F.
#1	(78.77,1)	93.63	94.87	96.43	90.91	96.43	10
#2	(100,1)	98.47	97.44	96.97	100	98.46	14
#3	(81.28,100)	89.44	84.62	80.00	100	88.89	7
#4	(35.69,29.51)	94.49	92.31	89.29	100	94.34	12
#5	(100,1)	98.20	97.44	96.43	100	98.18	17
Mean ± SD	(79.15,26.50) ± (26.28,42.90)	94.85 ± 3.71	93.33 ± 5.32	91.82 ± 7.33	98.18 ± 4.07	95.26 ± 3.92	12 ± 3.81

Table 8
The detailed results obtained by W2 weighted classification with feature selection.

	W2:ABC-KWELM					
Fold	(C, γ)	G-mean (%)	ACC (%)	Sensitivity (%)	Specificity (%)	F-measure (%)	#S.F.
#1	(35.78,1)	98.32	97.44	96.67	100	98.31	14
#2	(34.87,1.22)	98.43	97.44	96.88	100	98.41	11
#3	(59.15,1)	90.27	89.74	88.89	91.67	92.31	14
#4	(36.91,1)	100	100	100	100	100	16
#5	(85.93,1)	100	100	100	100	100	11
Mean ± SD	(50.52,1.04) ± (22.23,0.09)	97.40 ± 4.07	96.92 ± 4.21	96.49 ± 4.54	98.33 ± 3.73	97.81 ± 3.18	13.2 ± 2.17
	W2:AABC-KWELM					
Fold	(C, γ)	G-mean (%)	ACC (%)	Sensitivity (%)	Specificity (%)	F-measure (%)	#S.F.
#1	(100,1)	98.32	97.44	96.67	100	98.31	14
#2	(39.70,1)	98.20	97.44	96.43	100	98.18	13
#3	(80.11,1)	100	100	100	100	100	12
#4	(57.65,1)	100	100	100	100	100	14
#5	(82.35,1)	100	100	100	100	100	16
Mean ± SD	(71.96,1) ± (23.48,0)	99.30 ± 0.95	98.97 ± 1.40	98.62 ± 1.89	100 ± 0	99.30 ± 0.96	13.8 ± 1.48

ysis of variance (ANOVA) based on ACC. The result shows ACC has homogeneity of variance, and the value of the probability is 0.008 which is lower than 5% significance level. It illustrates average ACC among six algorithms exists significant difference.

4.5. Experiment 2: unweighted and weighted classification with feature selection

In this experiment, unweighted and weighted classification performance with feature selection is also evaluated, respectively. The purpose of this experiment is to examine whether feature selection can improve PD diagnostic performance. Both binary and continuous versions of ABC are used for performing feature selection and parameters optimization by ABC-based optimization algorithm, while both binary and continuous versions of AABC are used for performing feature selection and parameters optimization by AABC-based optimization algorithm. Tables 6–8 show the detailed results achieved via 5-fold CV. we find that the classification performance with feature selection is much more superior to that without feature selection from these results. From Table 8,

Table 9
Features obtained by the proposed method.

Fold	W2: AABC-KWELM Selected features
#1	F1, F2, F6, F7, F9, F12, F14, F15, F16, F17, F19, F20, F21, F22
#2	F1, F3, F4, F5, F6, F7, F9, F13, F14, F18, F20, F21, F22
#3	F1, F2, F3, F7, F8, F12, F14, F17, F18, F20, F21, F22
#4	F1, F2, F3, F4, F7, F8, F12, F15, F16, F18, F19, F20, F21, F22
#5	F2, F3, F4, F5, F6, F7, F9, F10, F12, F13, F14, F15, F18, F20, F21, F22

we can see that W2-based AABC-KWELM has achieved the best results of 100%, 98.62%, 98.97%, 99.30%, and 99.30% in accordance with specificity, sensitivity, ACC, G-mean and F-measure, respectively. With the aid of feature selection, W2-based AABC-KWELM has improved the performance by 5.67%, 1.93%, 3.07%, 3.85%, and 2.05% in accordance with specificity, sensitivity, ACC, G-mean and F-measure, respectively. Meanwhile, the average size of selected features namely #S.F. is 13.8 compared with the whole 22 features.

In addition, from the results we can see that SD of W2-based AABC-KWELM is much smaller than before. Fig. 5 depicts the box plot to show ACC results of six algorithms with feature selection. The box generated by W2-based AABC-KWELM is shorter than the boxes generated by other algorithms, which indicates that the proposed method is more reliable and robust through feature selection. Furthermore, one-way ANOVA based on ACC is also used to judge the difference among six algorithms with feature selection. The value of the probability is 0.032 which is lower than 5% significance level. It illustrates average ACC among six algorithms exists significant difference. In particular, combining LSD testing, the probability value between W2-based AABC-KWELM and W1-based ABC-KWELM is 0.003, and the probability value between W2-based AABC-KWELM and W1-based AABC-KWELM is 0.035 which illustrate there is significant difference between W2-based weighted strategy and W1-based weighted strategy.

We also record the evolutionary process of AABC-based and ABC-based optimization algorithm with feature selection. Fig. 6 presents the evolution of the average and best fitness obtained by W2-based weighted strategy for fold #1. From this figure, it can

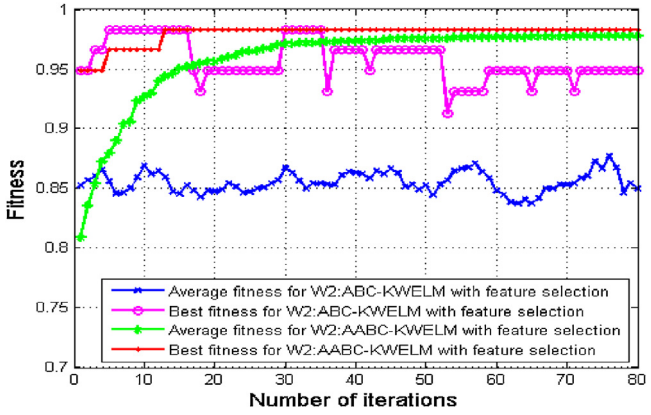

Fig. 6. The best and average fitness obtained by W2: ABC-KWELM and W2: AABC-KWELM with feature selection for fold #1.

be observed that the best fitness for W2-based AABC-KWELM is more stable than that W2-based ABC-KWELM. The average fitness for W2-based AABC-KWELM increases rapidly in the beginning of the evolution, and it starts increasing slowly after iteration 15. Thereafter, the average fitness gradually improves and it keeps stable after iteration 58. This phenomenon illustrates that AABC algorithm through feature selection is more reliable and robust than ABC algorithm. In order to demonstrate the detail of feature selection procedure, Table 9 lists selected features by W2-based AABC-KWELM during the whole 5-fold CV procedure.

Table 10 also presents the comparison results of the confusion matrix obtained by four algorithms without feature selection and with feature selection. As seen from Table 10, W2-based ABC-KWELM without feature selection can only correctly classify 184 cases in all 195 cases, misclassifies 8 PWP as healthy people and 3 healthy people as PWP. W2-based AABC-KWELM without feature selection correctly classifies 187 cases in all 195 cases, misclassifies 5 PWP as healthy people and 3 healthy people as PWP. While W2-based ABC-KWELM with feature selection correctly classifies

Table 10
Confusion matrix of four algorithms.

Algorithms	Actual	Predicted	
		PWP	Healthy people
W2:ABC-KWELM without feature selection	PWP	139	8
	Healthy people	3	45
W2: AABC-KWELM without feature selection	PWP	142	5
	Healthy people	3	45
W2:ABC-KWELM with feature selection	PWP	142	5
	Healthy people	1	47
W2: AABC-KWELM with feature selection	PWP	145	2
	Healthy people	0	48

Table 11
Comparative results by our method and other methods.

Study	Method	Accuracy (%)
Ozcift and Gulten [7]	CFS-RF	87.1 (10-fold CV)
Little et al. [4]	Pre-selection + kernel SVM	91.4 (bootstrap with 50 replicates)
AStröm and Koker [8]	Parallel NN	91.20 (hold-out)
Das [5]	ANN	92.9 (hold-out)
Guo et al. [6]	GP-EM	93.1 (10-fold CV)
Li [10]	Fuzzy-based transformation + SVM	93.47 (hold-out)
Luukka [9]	Fuzzy entropy measures + Similarity classifier	85.03 (hold-out)
Chen et al. [12]	PCA-FKNN	96.07 (average 10-fold CV)
Polat [11]	Fuzzy c-means clustering-based feature weighting + KNN	97.93 (hold-out)
Zuo [13]	PSO-FKNN	97.47 (10-fold CV)
Gök [14]	RF ensemble KNN classifier	98.46 (hold-out)
This study	AABC-KWELM	98.97 (5-fold CV)

189 cases in all 195 cases, misclassifies 5 PWP as healthy people and 1 healthy people as PWP. W2-based AABC-KWELM with feature selection correctly classifies 193 cases in all 195 total cases, only misclassifies 2 PWP as healthy people. From the above analysis, we can find that feature selection can improve PD diagnostic performance.

Table 11 lists the classification accuracy achieved by the previous methods for PD diagnosis. By comparison, our developed method can achieve better classification performance. The proposed method will show great potential in the area of clinical PD diagnosis.

5. Conclusions

In this study, we have proposed AABC-KWELM model to handle PD diagnosis problem. The core components of AABC-KWELM are feature selection, parameters optimization and performance evaluation. It is proven that AABC-KWELM has achieved better classification performance of 100%, 98.62%, 98.97%, 99.30%, and 99.30% in accordance with specificity, sensitivity, ACC, G-mean and F-measure, respectively. Therefore, it can be concluded the proposed method can well enough discriminate healthy people from PWP, and can assist the physicians to make accurate PD diagnosis. Furthermore, the future work is that the proposed method is to be evaluated in other medical diagnosis areas.

References

[1] J. Jankovic, Parkinson's disease: clinical features and diagnosis, J. Neurol. Neurosurg. Psychiatry 79 (4) (2008) 368–376.
[2] N. Singh, V. Pillay, Y.E. Choonara, Advances in the treatment of Parkinson's disease, Prog. Neurobiol. 81 (1) (2007) 29–44.
[3] A.K. Ho, et al., Speech impairment in a large sample of patients with Parkinson's disease, Behav. Neurol. 11 (1998) 131–138.
[4] M.A. Little, et al., Suitability of dysphonia measurements for telemonitoring of Parkinson's disease, IEEE Trans. Biomed. Eng. 56 (4) (2009) 1015–1022.
[5] R. Das, A comparison of multiple classification methods for diagnosis of Parkinson disease, Expert Syst. Appl. 37 (2) (2010) 1568–1572.
[6] P.F. Guo, P. Bhattacharya, N. Kharma, Advances in detecting Parkinson's disease, Med. Biom. (2010) 306–314.
[7] A. Ozcift, A. Gulten, Classifier ensemble construction with rotation forest to improve medical diagnosis performance of machine learning algorithms, Comput. Methods Progr. Biomed. 104 (3) (2011) 443–451.
[8] F. Aström, R. Koker, A parallel neural network approach to prediction of Parkinson's disease, Expert Syst. Appl. 38 (10) (2011) 12470–12474.
[9] P. Luukka, Feature selection using fuzzy entropy measures with similarity classifier, Expert Syst. Appl. 38 (4) (2011) 4600–4607.
[10] D.C. Li, C.W. Liu, S.C. Hu, A fuzzy-based data transformation for feature extraction to increase classification performance with small medical data sets, Artif. Intell. Med. 52 (1) (2011) 45–52.
[11] K. Polat, Classification of Parkinson's disease using feature weighting method on the basis of fuzzy C-means clustering, Int. J. Syst. Sci. 43 (4) (2012) 597–609.
[12] H.-L. Chen, et al., An efficient diagnosis system for detection of Parkinson's disease using fuzzy k-nearest neighbor approach, Expert Syst. Appl. 40 (1) (2013) 263–271.
[13] W.-L. Zuo, et al., Effective detection of Parkinson's disease using an adaptive fuzzy k-nearest neighbor approach, Biomed. Signal Process. Control 8 (4) (2013) 364–373.
[14] M. Gök, An ensemble of k-nearest neighbors algorithm for detection of Parkinson's disease, Int. J. Syst. Sci. 46 (6) (2015) 1108–1112.
[15] W.-W. Zong, G.-B. Huang, Y.-Q. Chen, Weighted extreme learning machine for imbalance learning, Neurocomputing 101 (3) (2013) 229–242.
[16] Y. Zhang, B. Liu, J. Cai, S.-H. Zhang, Ensemble weighted extreme learning machine for imbalanced data classification based on differential evolution, Neural Comput. Appl. (2016) 1–9.
[17] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.
[18] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over- sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.
[19] X.-Y. Liu, J. Wu, Z.-H. Zhou, Exploratory under-sampling for class-imbalance learning, IEEE Trans. Syst. Man Cybern.—Part B 39 (2) (2009) 539–550.
[20] Z.-H. Zhou, X.-Y. Liu, Training cost-sensitive neural networks with methods addressing the class imbalance problem, IEEE Trans. Knowl. Data Eng. 18 (1) (2006) 63–77.
[21] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (2006) 489–501.
[22] Q.-Y. Zhu, A.K. Qin, P.N. Suganthan, G.-B. Huang, Evolutionary extreme learning machine, Pattern Recognit. 38 (10) (2005) 1759–1763.
[23] Y. Kaya, M. Uyar, A hybrid decision support system based on rough set and extreme learning machine for diagnosis of hepatitis disease, Appl. Soft Comput. 13 (8) (2013) 3429–3438.
[24] H.-L. Chen, et al., An efficient hybrid kernel extreme learning machine approach for early diagnosis of Parkinson's disease, Neurocomputing 184 (4745) (2016) 131–144.
[25] G.-B. Huang, et al., Extreme learning machine for regression and multiclass classification, IEEE Trans. Syst. Man Cybern.—Part B: Cybern. 42 (2) (2012) 513–529.
[26] D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization, Technical Report-tr 06, Erciyes University, Engineering Faculty Computer Engineering Department, 2005.
[27] Y.-M. Huang, J.-C. Lin, A new bee colony optimization algorithm with idle-time-based filtering scheme for open shop-scheduling problems, Expert Syst. Appl. 38 (5) (2011) 5438–5447.
[28] A. Singh, S. Sundar, An artificial bee colony algorithm for the minimum routing cost spanning tree problem, Soft Comput. 15 (12) (2011) 2489–2499.
[29] W.-L. Xiang, M.-Q. An, An efficient and robust artificial bee colony algorithm for numerical optimization, Comput. Oper. Res. 40 (5) (2013) 1256–1265.
[30] C. Ma, J. Ouyang, H.-L. Chen, J.-C. Ji, A novel kernel extreme learning machine algorithm based on self-adaptive artificial bee colony optimization strategy, Int. J. Syst. Sci. 47 (6) (2014) 1342–1357.
[31] P.L. Bartlett, 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) (1998) 525–536.
[32] W.-F. Gao, S.-Y. Liu, L.-L. Huang, Enhancing artificial bee colony algorithm using more information-based search equations, Inf. Sci. 270 (1) (2014) 112–133.
[33] W.-F. Gao, S.-Y. Liu, L.-L. Huang, A novel artificial bee colony algorithm based on modified search equation and orthogonal learning, IEEE Trans. Cybern. 43 (3) (2013) 1011–1024.
[34] J. Kennedy, R.C. Eberhart, A discrete binary version of the particle swarm algorithm, Proceedings of IEEE Conference on Systems Man and Cybernetics (1997) 4104–4108.
[35] R. Storn, K. Price, Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (4) (1997) 341–359.
[36] L.-N. Li, et al., A computer aided diagnosis system for thyroid disease using extreme learning machine, J. Med. Syst. 36 (5) (2012) 3327–3337.
[37] University of California Irvine (UCI) machine learning repository. http://archive.ics.uci.edu/ml/datasets/Parkinsons.