Engineering Applications of Artificial Intelligence 63 (2017) 54–68

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Grey wolf optimization evolving kernel extreme learning machine: Application to bankruptcy prediction

Mingjing Wang [a], Huiling Chen [a,*], Huaizhong Li [b], Zhennao Cai [a], Xuehua Zhao [c], Changfei Tong [a], Jun Li [a], Xin Xu [d]

[a] College of Physics and Electronic Information Engineering, Wenzhou University, 325035 Wenzhou, China
[b] Department of Computing, Lishui University, Lishui 323000, Zhejiang, China
[c] School of Digital Media, Shenzhen Institute of Information Technology, Shenzhen 518172, China
[d] Electric Power Research Institute, State Grid Jilin Electric Power Company Limited, Changchun 130021, China

A R T I C L E I N F O

Article history:
Received 13 May 2016
Received in revised form
17 February 2017
Accepted 9 May 2017

Keywords:
Kernel extreme learning machine
Parameter tuning
Grey wolf optimization
Bankruptcy prediction

A B S T R A C T

This study proposes a new kernel extreme learning machine (KELM) parameter tuning strategy using a novel swarm intelligence algorithm called grey wolf optimization (GWO). GWO, which simulates the social hierarchy and hunting behavior of grey wolves in nature, is adopted to construct an effective KELM model for bankruptcy prediction. The derived model GWO-KELM is rigorously compared with three competitive KELM methods, which are typical in a comprehensive set of methods including particle swarm optimization-based KELM, genetic algorithm-based KELM, grid-search technique-based KELM, extreme learning machine, improved extreme learning machine, support vector machines and random forest, on two real-life datasets via 10-fold cross validation analysis. Results obtained clearly confirm the superiority of the developed model in terms of classification accuracy (training, validation, test), Type I error, Type II error, area under the receiver operating characteristic curve (AUC) criterion as well as computational time. Therefore, the proposed GWO-KELM prediction model is promising to serve as a powerful early warning tool with excellent performance for bankruptcy prediction.

1. Introduction

Due to financial crisis all over the world, company bankruptcy predication attracts significant attention for financial institutions. It is important for enterprises to build a trustworthy and accurate early warning system to predicate potential risk of company's bankruptcy beforehand.

Bankruptcy predication generally forms a binary classification that needs to be resolved in a rational approach. The output result generated from the classification models has two types, namely, type 1 represents a company with bankruptcy and type 0 otherwise. Input values of the classification models are often financial statistic ratios derived from credible financial statements in the real enterprises. So far, considerable amount of classification models based on different domain knowledge has been proposed for bankruptcy prediction. In general, the proposed predication models can be classified as statistical approaches or artificial intelligence methods (AI).

A great deal of typical statistical approaches that are constructed for bankruptcy prediction models apply simple univariate analysis (Beaver, 1966), multivariate discriminant analysis (Altman, 1968), logistic regression (Ohlson, 1980) and factor analysis (West, 1985). Recently, AI methods are drawing more attention for failure prediction. Approaches that are based on the AI means, such as artificial neural networks (ANN) (Atiya, 2001a), support vector machines (SVM) (Min and Lee, 2005; Shin et al., 2005), k-nearest neighbor (KNN) approach (Chen et al., 2011c), Bayesian network models (Sarkar and Sriram, 2001; Sun and Shenoy, 2007), extreme learning machine and ensemble methods (Fedorova et al., 2013; Abellán and Mantas, 2014), as well as different hybrid approaches, have been widely used in financial area. Reddy and Ravi (2013) constructed two novel kernels based soft computing techniques for classification task. The experimental results indicated that the proposed approaches could perform well for bankruptcy prediction. Sharma et al. (2013) successfully proposed a hybrid algorithm based on ant colony optimization and Nelder-Mead simplex for training neural networks with an application to bankruptcy prediction. Paramjeet and Ravi (2011) modified bacterial foraging technique to train wavelet neural network in order to predict bankruptcy in banks. A hybrid approach based on differential evolution and radial basis function network (DERBF) proposed by Naveen et al. (2010) was applied to

* Corresponding author.
E-mail address: chenhuiling.jlu@gmail.com (H. Chen).

http://dx.doi.org/10.1016/j.engappai.2017.05.003
0952-1976/

bankruptcy prediction. The results showed that DERBF had a good performance of generalization on bank bankruptcy datasets. Chauhan et al. (2009) employed differential evolution algorithm to train wavelet neural network (DEWNN), predicting the bankruptcy in banks. The results on the four bankruptcy datasets revealed that the DEWNN was obviously superior to other existed methods. Ravi and Pramodh (2008) proposed a new architecture called principal component neural network (PCNN) applied to bankruptcy prediction problem in commercial banks. It is inferred that the proposed PCNN hybrids outperformed other classifiers on the bankruptcy dataset. A new neural network architecture kernel principal component neural network (KPCNN) trained by threshold accepting was presented in Ravisankar and Ravi (2009). Its application to bankruptcy prediction in banks reveled that KPCNN yields comparable results with all the techniques. Vasu and Ravi (2011) proposed new principal component analysis-wavelet neural network hybrid (PCATAWNN) architecture trained by threshold accepting algorithm to predict bankruptcy in banks. The experimental results showed that the PCATAWNN could convincingly outperformed other techniques in terms of area under ROC curve (AUC) in 10-fold cross-validation. In all of the employed methods, ANN (Tsai and Wu, 2008; Atiya, 2001b; Zhang et al., 1999) has become more and more popular for financial prediction, thanks to its prominent ability to capture the nonlinearity relationship that exists between different features in real data set. Nevertheless, it is worth to point out that traditional ANN learning methods, such as the back-propagation approach, are based on the gradient descent strategy which may result in local optimum. Furthermore, it is generally required that a fair amount of network parameters be tuned.

In order to avoid ANN's drawbacks, Huang et al. proposed a new machine learning paradigm named extreme learning machine (ELM) (Huang et al., 2006). ELM is a representative learning model of neural network named after single hidden layer feedforward neural networks (SLFNs). The hidden biases and input weights in this method can be randomly generated, and the output weights are mathematically determined using Moore-Penrose (MP) generalized inverse. It is well-known that the universal approximation can reflect the approximation capabilities of the neural networks. The approximation capabilities of multilayer feedforward networks were proved by Hornik (1991), namely, noconstant bounded continuous activation functions and continuous mappings could be approximated in measure by neural networks. Leshno et al. (1993) advocated that continuous functions could be approximated by feedforward networks with a non-polynomial activation function. Guang-Bin and Babri (1998) proposed that SLFNs with N hidden nodes and almost nonlinear activation function could exactly learn N distinct observations. Due to its classification performance, ELM has been adopted in fields such as image classification (Cao et al., 2016a; Jun et al., 2011), disease diagnosis (Chen et al., 2015; Zhang et al., 2007), and engineering application (Cao et al., 2016b, 2015). In addition, methods based on ELM have also been widely applied in financial areas such as bankruptcy prediction (Yu et al., 2014), corporate life cycle prediction (Lin et al., 2013) and corporate credit ratings (Zhong et al., 2014). One limitation of ELM, nevertheless, is that the randomly assigned input weights can increase the variations of accuracies obtained by classifiers in multiple trials. In order to overcome this limitation, Huang et al. (2012) proposes an extension version of ELM, namely, kernel extreme learning machine (KELM), whose connection weights between hidden layers and input are not necessary. Compared with ELM, KELM can achieve comparative or more excellent property with faster training speed and much easier implementation in applications such as hyperspectral remote-sensing image classification (Pal et al., 2013; Chen et al., 2014), activity recognition (Deng et al., 2014), 2-D profiles reconstruction (Liu et al., 2014), disease diagnosis (Chen et al., 2016) and fault diagnosis (Jiang et al., 2014).

We recently applied the KELM to bankruptcy prediction's issue (Zhao et al., 2017), and obtained better performance than other five competitive approaches including SVM, ELM, random forest (RF), particle swarm optimization boosted fuzzy KNN, and Logit model on the same real data set. Nevertheless, it should be noticed that the two significant parameters in KELM with RBF kernel are kernel penalty parameter C and bandwidth γ. C controls the trade-off between the model complexity and the fitting error minimization, while γ defines the non-linear mapping from the input space to some high-dimensional feature space. Several studies have illustrated that these two parameters have an important effect on KELM's performance, similar to that in SVM. Thus, these two key parameters must be properly set prior to its application to realistic problems. These parameters are traditionally obtained using the grid-search method whose main drawback, however, is that it is easy to be trapped in a local optimum. Presently, it has been shown that biologically-inspired methods (such as the genetic algorithm (Liu et al., 2014), particle swarm optimization (PSO) (Zhang and Yuan, 2015), and artificial bee colony (Ma et al., 2016) are more likely to find the global-best solution than the grid-search method. As a new member in the nature-inspired methods, Grey Wolf Optimizer (GWO) (Mirjalili et al., 2014) mimics the social hierarchy and hunting behavior of grey wolves in nature. The main traits of GWO are social hierarchy, encircling prey, hunting, attacking prey (exploitation), and search for prey (exploration).

Due to its good search ability, GWO has been applied in a various fields. Muangkote et al. (2014) used the GWO with improvements to training q-Gaussian Radial Basis Functional-link nets neural networks. The experimental result indicated that the proposed algorithm obtained competitive performance comparing with other meta-heuristic methods. Komaki and Kayvanfar (2015) successfully applied GWO for the two-stage assembly flow shop scheduling problem with release time to greatly improve the efficiency. Sulaiman et al. (2015) used GWO to solve optimal reactive power dispatch problem. Mirjalili (2015) employed GWO to train multi-layer perceptron and eight standard datasets including five classification and three function-approximation datasets ware evaluated. The results demonstrated that a high level of accuracy in classification and approximation of the proposed trainer could be obtained. However, to the best of our knowledge, the potential of GWO has not been explored to fine tune the optimal parameters appeared in KELM. Therefore, this study aims at exploring the GWO technique's ability to address KELM's model selection problem for classification, and further applying the resulted model GWO-KELM to successfully and effectively predict company bankruptcy. For verification purpose, the effectiveness and efficiency of the proposed GWO-KELM is compared against the common methods such as grid-search optimized KELM (GS-KELM), genetic algorithm optimized KELM (GA-KELM), particle swarm optimization optimized KELM (PSO-KELM) and other four advanced machine learning methods including original ELM, self-adaptive evolutionary extreme learning machine proposed by Cao et al. (2012) (SaE-ELM), SVM and RF on the real-life financial dataset. All methods are compared in terms of the training accuracy, the validation accuracy and test accuracy, Type I error, Type II error and the area under the receive operating characteristic curve (AUC) criterion. For the stability of the results, the cross validation (CV) strategy is also adopted including external 10-fold CV and the inner 5-fold CV. The experimental results show that our proposed methodology, GWO-KELM, performs better when compared with some other well-known common methods. The main contribution of this study can be summarized as follows:

a) A new nature-inspired method, GWO is successfully employed to resolve the parameters optimization for KELM for the first time.
b) A potential model, GWO-KELM, is successfully applied to bankruptcy prediction with the purpose of being treated as a potential early warning tool for bankruptcy in the financial field.
c) The proposed GWO-KELM tends to achieve better classification, and generate more stable and robust results in discriminating the bankrupt companies from the healthy ones when compared to several other methods.

The remainder of this paper is structured as following: Section 2 presents a brief description of GWO. Section 3 explains the detailed implementation of the GWO-KELM methodology. In Section 4, the details of the experimental designs are elaborated. The experimental results are presented in Section 5. Conclusions are finally summarized in Section 6.

2. Grey wolf optimization (GWO)

Recently, a new swarm intelligence optimization algorithm called GWO was introduced by Mirjalili et al. (2014). This creative algorithm actually simulates the social hierarchy and hunting behavior of grey wolves in nature. For modeling the social hierarchy behavior of grey wolf, the group is divided into four parts: alpha (α), beta (β), delta (δ), and omega (ω) as shown in Fig. 1.

α is considered to be the best fittest solution followed by β and δ, respectively, and the rest of solutions are belonging to the ω. The first three fittest wolves that are closest to the prey are α, β and δ who guide ω to search prey in promising search areas. During encircling grey, the wolves update their position surrounding α, β, or δ as shown in Eqs. (1) and (2):

$$\vec{D} = \left|\vec{C}\cdot\vec{X}_{p(t)} - \vec{X}(t)\right| \tag{1}$$

$$\vec{X}(t+1) = \vec{X}_{p(t)} - \vec{A}\cdot\vec{D} \tag{2}$$

where t is the current iteration, $\vec{X}_{P(t)}$ indicates the current position of prey and $\vec{X}(t)$ means the current position of a wolf. $\vec{D}$ is the distance between wolves and prey, and coefficient vectors $\vec{A}$ and $\vec{C}$ are mathematically given as follows.

$$\vec{A} = 2\vec{a}\,\vec{r}_1 - \vec{a} \tag{3}$$

$$\vec{C} = 2\vec{r}_2 \tag{4}$$

where $\vec{r}_1$ and $\vec{r}_2$ are two vectors generating between [0, 1] randomly, the element of $\vec{a}$ is linearly decreasing from 2 to 0 at each process of iteration. In the GWO algorithm, α, β, and δ are always assumed to be likely near the position of the prey. During the process of hunting, the first three best solutions achieved so far in terms of α, β, and δ are saved and remained, then the remaining wolves such as ω which are capable of re-position according to the first three best wolves. The positions of wolves are updated according to the Eqs. (5)–(11):

$$\vec{D}_\alpha = \left|\vec{C}_1\cdot\vec{X}_\alpha - \vec{X}\right| \tag{5}$$

$$\vec{D}_\beta = \left|\vec{C}_2\cdot\vec{X}_\beta - \vec{X}\right| \tag{6}$$

$$\vec{D}_\delta = \left|\vec{C}_3\cdot\vec{X}_\delta - \vec{X}\right| \tag{7}$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1\cdot\left(\vec{D}_\alpha\right) \tag{8}$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2\cdot\left(\vec{D}_\beta\right) \tag{9}$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3\cdot\left(\vec{D}_\delta\right) \tag{10}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{11}$$

where $\vec{X}_\alpha$ shows the position of α, $\vec{X}_\beta$ indicates the position of β, $\vec{X}_\delta$ shows the position of δ, $\vec{X}$ shows the position of current solution, $\vec{C}_1$, $\vec{C}_2$ and $\vec{C}_3$ are vectors generating randomly. The approximate distance between the current solution and α, β and δ are calculated according the Eqs. (5)–(7). Eqs. (8)–(11) calculate the final position of the current solution after defining the distance. Where $\vec{A}_1$, $\vec{A}_2$ and $\vec{A}_3$ are random vectors, and t indicates the number of iteration. As we may see from the above equations, the step size of ω wolves running after α, β, and δ are defined by Eqs.

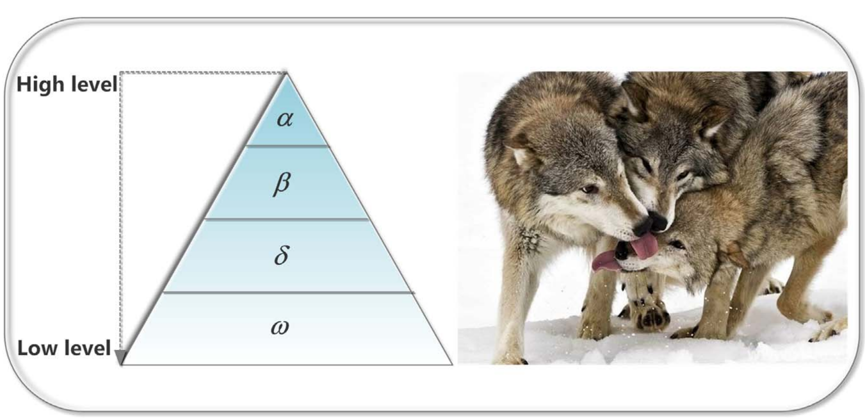

Fig. 1. The social hierarchy of grey wolves.

(5)–(7), respectively. Then the final positions of the ω wolves are calculated based on the Eqs. (8)–(11). In short, the regular steps of the GWO algorithm are descried as follows:

Grey Wolf Optimizer Algorithm
Step 1. Parameter initialization
Initialize the parameters for GWO, such as the maximum iteration number, the population size of wolves, the boundary of search and the dimensionality of the space.
Step2. Population initialization
Give the full parameters to initialize the position of each wolf which is two-dimension in this study.
Step 3. Population location of the assignment
Firstly, C and $\boldsymbol{\gamma}$ are coded as follows:
C = Position (i, 1);
γ = Position (i, 2);
Step 4. Calculate the fitness of each wolf
Fitness(i) = Function (C, γ);
Step 5. Keep the three best fitness of wolves
Retain the three best fitness and positions of coordinate positions. Then, the grey wolves run towards the location after the best wolves of the three best fitness values.
[~, index] = sort (fitness, 'descend');
Alpha_score = fitness (index (1));
Alpha_position= Position (index (1),:);
Beta_score = fitness (index (2));
Beta_position = Position (index (2),:);
Deta_score = fitness (index (3));
Deta_position = Position (index (3),:);
Step 6. Update the position of each wolf
Update the positions located after the best positions of the best three wolves according to the Eqs. (5)–(11).
Step 7. Iterative optimization
Enter the iterative optimization to repeat the implementation of steps 3–6. The circulation stops when the iterative number reaches the maximal iterative number.

3. Evolutionary KELM model

3.1. Kernel extreme learning machine (KELM)

KELM is a new learning techniques developed in recent years. It evolves from ELM, and has been moderately proven to exhibit more honorable generalization in a large number of practical applications. For clarity, a short description of KELM is presented in this study, while the details can be found in Huang et al. (2012).

ELM was proposed firstly by Huang et al. (2006) (Guang-Bin et al., 2004) for the single-hidden-layer feedforward neural networks and was then treated to be the generalized SLFNs without tuning parameters in the hidden layer as like. The output function of ELM for generalized SLFNs is shown in Eq. (12), here as an example, the one output node is taken.

$$f_L(x) = \sum_{i=1}^{L} \beta_i h_i(x) = h(x)\beta \tag{12}$$

where $\beta = [\beta_1, \ldots, \beta_L]^T$ indicates the vector of the output weights between the hidden layer of L nodes and the output node, the output vector of the hidden layer with related to input x is defined as $h(x) = [h_1(x), \ldots, h_L(x)]$. Then the dataset can be mapped from the d-dimensional input space to the L-dimensional hidden-layer feature space H according to $h(x)$, which means that $h(x)$ is a feature mapping. It is known from Bartlett's (1998) theory with the purpose for reaching smaller training error of the feedforward neural networks, the smaller the norms of weights are, the better generalization performance the networks tend to acquire. ELM tends to minimize the training error and the norm of the output weights simultaneously.

$$\min \|H\beta - T\|^2 \text{ and } \|\beta\| \tag{13}$$

H of the former formation means the hidden-layer output matrix

$$H = \begin{bmatrix} h(x_1) \\ h(x_2) \\ h(x_3) \end{bmatrix} = \begin{bmatrix} h_1(x_1) & \cdots & h_L(x_1) \\ \vdots & \ddots & \vdots \\ h_1(x_N) & \cdots & h_L(x_N) \end{bmatrix} \tag{14}$$

Actual meaning of minimizing the norm of the output weights $\|\beta\|$ is to maximize the distance of the separating margins of different classes in the ELM feature space. The minimal normal least square method is also used in ELM

$$\beta = H^{\dagger}T \tag{15}$$

where $H^{\dagger}$ means the Moore-Penrose generalized inverse of matrix.

A real factor should be noticed, that is, when users do not know the feature mapping or the case is multiclass (Huang et al., 2012), a kernel matrix which is called kernel mapping function for the ELM can be taken by employing the following equation:

$$\Omega_{ELM} = HH^T: \Omega_{ELM_{i,j}} = h(x_i)\cdot h(x_j) = K(x_i, x_j) \tag{16}$$

where $h(x)$ is a mapping function which can map the data form input space to ensure it is linearly separable in hidden-layer feature space H. The orthogonal projection approach is taken to calculate the MP generalized matrix's inverse, namely, $H^{\dagger} = H^T(HH^T)^{-1}$, and a positive constant C is proposed to the diagonal of HH^T. In brief, ELM's output function can be described as follow:

$$F(x) = h\beta = h(x)H^{\dagger}\left(\frac{I}{C} + HH^{\dagger}\right)^{-1}T = \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_N) \end{bmatrix}^T \left(\frac{I}{C} + \Omega_{ELM}\right)^{-1} T \tag{17}$$

As shown in Fig. 2, users do not have to know the hidden layer feature that maps, which can be coped with the kernel trick, and usually the radial basis function kernel (RBF) is adopted. RBF kernel function is always defined as $K(x, x_i) = \exp(-\gamma\|x - x_i\|^2)$. These two crucial parameters that presented in RBF kernel are penalty parameter C and kernel parameter gamma γ.

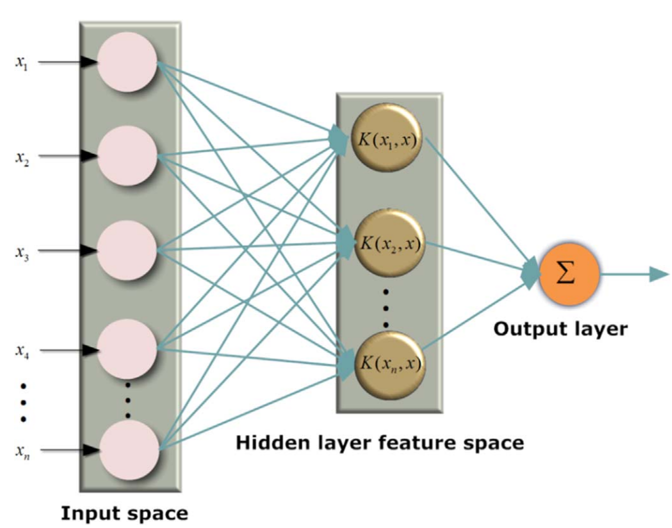

Fig. 2. The structure of KELM model.

3.2. GWO based KELM

As mentioned above, the property of KELM is mainly controlled by two key parameters of C and γ. In this study, we introduce a recently developed global optimization algorithm, GWO, to identify the two parameters in KELM. The resulted evolutionary KELM method, named GWO-KELM, employs the GWO strategy to adaptively determine the two key parameters in KELM. The general framework of the proposed methodology is demonstrated in Fig. 3. GWO-KELM is primarily composed of two procedures including parameter optimization and classification property evaluation. During the process of parameter optimization, the inner 5-fold CV is conducted on the nine tenths of the whole dataset. When the procedure of inner parameter optimization is terminated, the optimal parameter pair is fed to the KELM prediction model to carry out the classification task for bankruptcy predication in the outer loop according to use the external 10-fold CV analysis.

The classification accuracy is taken into consideration in designing the fitness:

$$F = avgACC = \frac{\left(\sum_{i=1}^{K} testACC_i\right)}{K} \tag{18}$$

where *avgACC* in this function *F* means the test accuracy achieved by the KELM classifier via the 5-fold CV scheme through the inner parameter optimization procedure. The pseudo-code of the parameter adjustment is as follow:

Parameter optimization procedure
Begin
  For i=1 to search agent number
    Set the KELM parameters with the initialized positions;
    Calculate the initial fitness;
    Train KELM with the initialized position, and record test results into the fitness array;
  End
Sort the fitness obtained by the function, save the three best fitness and the related positions;
  [~, index1]=sort (fitness,' descend');
  Alpha_score=fitness (index1 (1));
  Alpha_position=Position (index1 (1),:);
  Beta_score=fitness (index 1(2));
  Beta_position=Position (index1 (2),:);
  Deta_score=fitness (index1 (3));
  Deta_position=Position (index1 (3),:);
l=1;
While l< =Maxiteration
The rest of wolves run around the best three wolves; therefore the position of each wolf can be updated.
    a=2-l* (2/Maxiteration);
    For i=1:size (Position, 1)
      For j=1: size (Position, 2)
        Update the positions of wolves according the Eqs. (5)–(11);
      End
    End
    For i=1 to search agent number
      Control the search space of each wolf to avoid going out of the boundary;
    End
      Save the current information of three best wolves including three best fitness and corresponding positions.
    b=[Alpha_score; Beta_score; Delta_score];
    c=[Alpha_position; Beta_position; Deta_position];
    For i= to search agent number
      Calculate the initial fitness;
      Train KELM with the initialized position, and record the test results into the array;
    End
      Calculate the next information of three best wolves and save the corresponding results into c and d arrays.
    d=[genAlpha_score genBeta_score genDeta_score];
    e=[genAlpha_position; genBeta_position; genDeta_position];

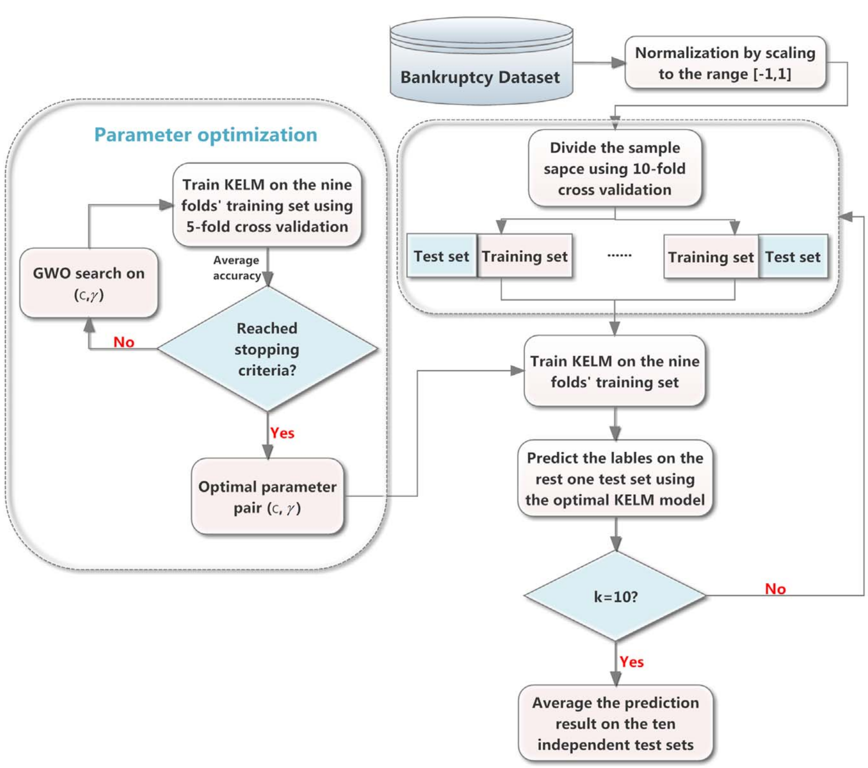

Fig. 3. Flowchart of the proposed GWO-KELM methodology.

Combine the current information with the next one to construct the new arrays f and g. f means the first three solutions of wolves at present two generations. g means the position at the same mechanism.
f=[b d];
g=[c; f];
Calculate the new three best wolves by sorting f and g, and then other wolves should run around the three best wolves. The wolf with the first fitness is Alpha, wolf with the second fitness is Beta, and the third one is a wolf named after Deta.
Because of the fact that the Alpha wolf has the best fitness at each generation, the *BestC* and *Bestγ* value the position of Alpha.
BestC=Alpha_position (1,1);
Bestγ=Alpha_position (1,2);
l=l+1
End
Return *BestC*, *Bestγ*;
End

Fig. 4 dynamically simulates the process of wolves hunting prey, which represents the parameter set of GWO-KELM model. In the figure, both green and blue dots represent the positions of wolves in the period of hunting. The green dots indicate the continuous updating positions at the every iteration and the blue dots represent the trajectories of the best grey wolves in each group of the 250 iterations of the GWO-KELM model. It can be seen that the density around the blue dots is much larger than that those whose positions are at the edge of the figure, which implies the fact that wolves are gradually approaching close to the prey. Finally, the blue dots are gathered together, which means that wolves have successfully caught the objective prey. In the process of hunting, α, β, and δ lead the other wolves to hunt the prey and they are the best three wolves from the fitness function.

4. Experimental designs

4.1. Data description

Two experimental datasets, the Wieslaw dataset and Japanese Dataset (JPNBDS) from different sources, are adopted to evaluate the effectiveness of the proposed GWO-KELM for bankruptcy prediction. The Wieslaw dataset (Pietruszkiewicz, 2008) has been used in many studies, it is composed of 30 financial ratios and 240 real companies that were collected during a five years period in Polish. This data set can be called a balanced data set with 112 bankrupt enterprises and 128 non-bankrupt ones gathered during the spanning about 3 years before the enterprises go bankrupt. JPNBDS is available at http://goo.gl/IFFzDp whose financial ratios are collected from some Japanese financial statements. This dataset was collected from 1995 to 2009 with a total of 76 non-bankrupt observations and 76 bankrupt observations. To consider the different financial ratios, the whole thirty traditional financial ratios of the Wieslaw dataset were taken as the features of this dataset, and then ten financial ratios were selected in JPNBDS according to the use frequency in the previous literatures such as described in Ravi Kumar and Ravi (2007). Some of ten ratios also existed in the Wieslaw dataset and the remaining ones were different from the whole thirty traditional ratios. In order to provide a unbiased and reliable result of this experiment, the classification performance was evaluated using an external 10-fold CV scheme (Salzberg, 1997), and the inner 5-fold CV scheme was used for searching for the optimal values of the two parameters. It should be noted that the inner 5-fold CV was done on the data which did not appear in the test set. This scheme was adopted in many studies (Chen et al., 2011a, 2011b), and the detailed description was vividly shown in Fig. 5. At each process of one experiment, a specific subset was formed by choosing the nine of the 10 subsets, then merged together as a whole, and the test set was the remaining one according to the external 10-fold CV scheme. Furthermore, one of the 5 subsets abstracted from the former specific subset was treated as validation dataset, and the remaining was treated as training dataset. The final result was obtained by averaging the results of 10 trials. To keep the same proportion of the bankrupt and the non-bankrupt enterprises in the same experiments, the entire dataset was partitioned by the stratified strategy, namely, the stratified CV scheme was employed in this study.

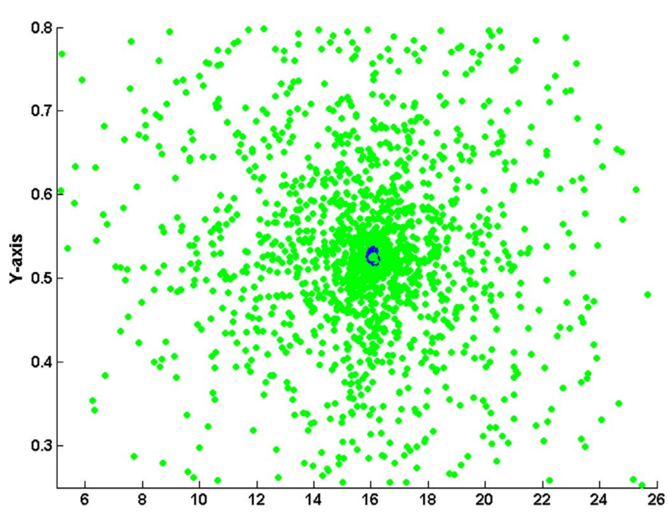

Fig. 4. The wolf swarm trajectories in searching for the parameters of the GWO-KELM model. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article).

4.2. Experimental setup

The GWO-KELM, PSO-KELM, GA-KELM and GS-KELM prediction models were implemented from scratch on the MATLAB platform. The empirical experiment was conducted on Intel (R) Core (TM) i7-4790 CPU @3.60 GHz with 8 GB of RAM and the system is Windows 7. For SVM, LIBSVM software package finished by Chang and Lin (2011) was employed. For RF, the code package from http://code.google.com/p/randomforest-matlab was adopted.

For fair comparison purpose, the same number of generations and swarm size of populations were used for GWO, PSO and GA. According to the results of our preliminary experiment, the proposed methods could fulfill the satisfactory classification performance, when the number of the generations and swarm size were set to 250 and 8, respectively. For the purposed of fair comparison, $C\in\{2^{-5}, 2^{-4.75}, 2^{-4.5} \ldots, 2^{5}\}$ and $\gamma\in\{2^{-5}, 2^{-4.75}, 2^{-4.5} \ldots, 2^{5}\}$ were used

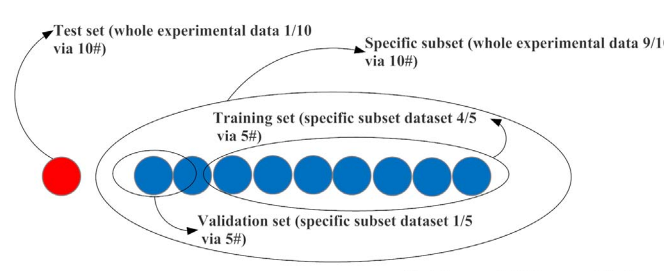

Fig. 5. The description of cross validation for the experimental datasets.

for the four methods. The other parameter values of PSO and GA were set as follows. In PSO, the maximum velocity was set about 60% of the dynamic ranges of the variable on each dimension, acceleration coefficients were both set as 2.05, and inertial weight was set to 1. In GA, the crossover probability and mutation probability were set to 0.4 and 0.01, respectively.

For ELM and SaE-ELM, we used the sigmoid function, sine function, hard-limit function, triangular basis function and radial basis function as the activation functions in turn to determine the best suitable function. The different number of hidden neurons varied from 1 to 200 with the step of 1 was tried to obtain the best optimal hidden neurons. It appeared that the sigmoid activation function with 46 hidden neurons was the best combination scheme for constructing the ELM and SaE-ELM models. Then the same crossover probability and mutation probability similar to that in GA were adopted in the SaE-ELM for the fair comparison. In RF, the number of the trees and the number of the variables chosen to be split varied between [50, 500] with the step of 50 and [1, 6] with the step of 1, respectively. Finally, a RF model with 500 trees and 3 features resulting in the best performance was adopted in this study.

4.3. Measure for performance evaluation

The performance of this proposed models were evaluated using the following metrics: ACC, AUC [61], Type I error, Type II error:

$$ACC = \frac{TP + TN}{TP + FP + FN + TN} \times 100\% \tag{19}$$

$$Type\quad I\quad error = \frac{FP}{(FP + TN)} \times 100\% \tag{20}$$

$$Type\quad II\quad error = \frac{FN}{(TP + FN)} \times 100\% \tag{21}$$

where TP is the number of true positives cases, which represents cases which are correctly categorized in the 'positive' class; FN is the number of false negatives, which represents 'positive' class cases that are classified as negative; TN is the number of true negatives, which represents cases that are correctly categorized in the 'negative' class; and FP is the number of false positive, which represents 'negative' class cases that are classified as positive. Type I and Type II errors are two important measures which describe how well the classifier discriminates between case with non-bankruptcy and with bankruptcy. Type I error (Ty-I-e) measures the proportion of bankrupt cases which are incorrectly identified as non-bankrupt one, which can be defined as FP/(FP+TN). Type II error (Ty-II-e) measures the proportion of non-bankrupt cases which are incorrectly identified as bankrupt ones, which can be defined as FN/(TP+FN). As one of the best popular methods for comparing classifiers in two-class problems, the AUC is the area under the ROC curve whose value is equivalent to the probability that a randomly chosen positive example is ranked higher than a random chosen negative example as described in Fawcett (2006). The method proposed in Fawcett (2006) was implemented to compute the AUC in this study. For more detailed experimental comparison, the validation accuracy (Va-acc), the training accuracy (Tr-acc) and test accuracy (Te-acc) in this study were also taken into account respectively.

Table 1
The statistical details of the Wieslaw dataset.

Attributes	Max	Min	Mean	Std
X_1(cash/current liabilities)	2.7050	0.0002	0.2190	0.4173
X_2(cash/total assets)	0.5295	0.0002	0.0608	0.0867
X_3(current assets/current liabilities)	6.8060	0.1850	1.5390	1.1866
X_4(current assets/total assets)	0.9988	0.0476	0.6024	0.2369
X_5(working capital/total assets)	0.7068	−1.0156	0.0553	0.2823
X_6(working capital/sales)	0.5587	−85.0100	−0.3539	5.5014
X_7(sales/inventory)	76,339	0	538	5272
X_8(sales/receivables)	551.2100	0	12.8839	38.5199
X_9(net profit/total assets)	0.6324	−0.6216	0.0221	0.1554
X_{10}(net profit/current assets)	0.8660	−1.8015	0.0127	0.3319
X_{11}(net profit/sales)	7.9078	−17.3400	−0.0869	1.8630
X_{12}(gross profit/sales)	8.9827	−17.3400	0.0338	1.9446
X_{13}(net profit/liabilities)	4.8850	−0.7466	0.1480	0.5573
X_{14}(net profit/equity)	2.6372	−54.0790	−0.5466	4.5080
X_{15}(net profit/(equity+long term liabilities))	2.6372	−54.0790	−0.2095	3.5414
X_{16}(sales/receivables)	551.2100	0	12.8839	38.5199
X_{17}(sales/total assets)	13.4990	0	2.5240	2.1365
X_{18}(sales/current assets)	26.5850	0	4.2821	3.2192
X_{19}((365[a]receivables)/sales)	4768.5	7	87.7	328.3
X_{20}(sales/total assets)	13.4990	0	2.5240	2.1365
X_{21}(liabilities/total income)	1722.3	0	8.2	111.3
X_{22}(current liabilities/total income)	1722.3	0	7.8	111.2
X_{23}(receivables/liabilities)	2.9571	0.0119	0.6928	0.5476
X_{24}(net profit/sales)	0.5717	−0.4810	0.0013	0.0994
X_{25}(liabilities/total assets)	1.8987	0.0344	0.6075	0.2998
X_{26}(liabilities/equity)	1383.9	−61.3	12.2	91.6
X_{27}(long term liabilities/equity)	755.7600	−43.7750	3.5922	49.1629
X_{28}(current liabilities/equity)	628.1400	−17.5390	8.6463	44.3332
X_{29}(EBIT[a]/total assets)	0.8722	−0.5915	0.0708	0.1842
X_{30}(current assets/sales)	78.2990	0.0376	0.6968	5.0787

[a] (EBIT means earnings before interests and taxes).

5. Experimental results

5.1. Wieslaw dataset

Table 1 lists the description of the 30 features and the statistical information of each feature in terms of the maximum (max) and minimum of (min), mean and standard deviation (std) value. To avoid the feature values in the greater numerical ranges plying a dominant role in those smaller numerical ranges, the value of each feature is normalized to the range from −1 to 1.

Table 2 illustrates the detailed results of the Tr-acc, Va-acc, Te-acc, Ty-I-e, Ty-II-e and AUC for each fold of the 10-fold CV and corresponding pair values of C and γ obtained by GWO-KELM. As shown in Table 2, GWO-KELM has achieved a high performance

Table 2
The detailed results obtained by GWO-KELM model with optimal parameter pair on Wieslaw dataset.

Fold	(C, γ)	Tr-acc	Va-acc	Te-acc	Ty-I-e	Ty-II-e	AUC
#1	(32, 0.0593)	0.99305	0.8611	0.8750	0.1428	0.1000	0.8786
#2	(3.6785, 0.2166)	1	0.8748	0.8333	0.1538	0.1818	0.8444
#3	(16.3413, 0.2561)	0.9907	0.8609	0.8750	0	0.1538	0.9231
#4	(22.3418, 0.5458)	0.9988	0.8614	0.8750	0.0909	0.1538	0.8776
#5	(2.9951, 0.1390)	0.9976	0.8658	0.8333	0.1538	0.1818	0.8322
#6	(4.7596, 0.2628)	1	0.8569	0.7917	0.1000	0.2857	0.8071
#7	(1.7792, 0.0752)	0.9791	0.8613	0.8333	0	0.2667	0.8667
#8	(5.0754, 0.2197)	0.9965	0.8609	0.8750	0.0833	0.1000	0.8786
#9	(15.5571, 0.4898)	0.9803	0.8732	0.8333	0.1538	0.1818	0.8321
#10	(31.7396, 0.3930)	0.9861	0.8631	0.8333	0.1539	0.1818	0.8322
Avg.		**0.9922**	**0.8639**	**0.8458**	**0.1032**	**0.1787**	**0.8573**
Dev.		**0.0079**	**0.0057**	**0.0281**	**0.0612**	**0.0604**	**0.0339**

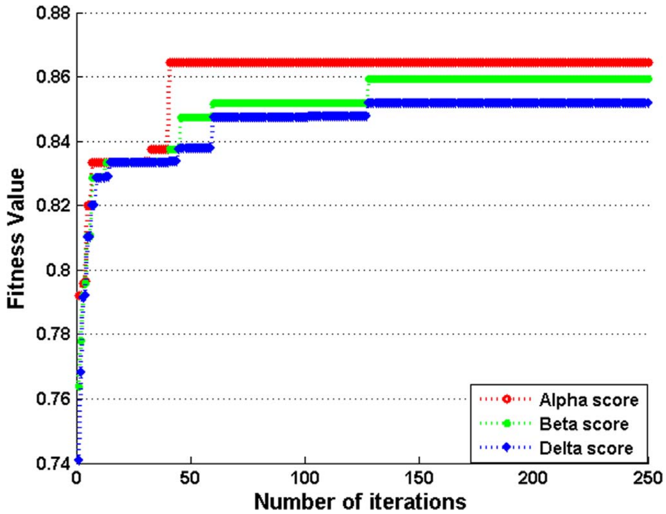

Fig. 6. The scores of Alpha, Beta, and Delta obtained by GWO-KELM on Wieslaw dataset.

with the average, Tr-acc of 0.9922, Va-acc of 0.8639, Te-acc of 0.8458, AUC of 0.8573, Ty-I-e of 0.1032 and Ty-II-e of 0.1787 and standard deviations of 0.0079, 0.0057, 0.0281, 0.0612, 0.0604, and 0.0339 respectively.

Fig. 6 illustrates the detailed scores obtained by α, β and δ with the increase of iterations from 1 to 250 on the Wieslaw dataset during the training step. It can be seen from the figure that α has a fast convergence towards the best score at the iterations of 48 approximately and then no obvious improvement can be made at all.

Learning curves during the training stage for fold 2(a), fold 4(b), fold 6(c) and fold 8(d) in 10-fold CV are exhibited in terms of the Best validation fitness, Average validation fitness, and Local best validation fitness in Fig. 7. A phenomenon can be found that accuracy curves achieved from fold 2(a), fold 4(b), and fold 8 (d) gradually improve from iteration 1 to about 48 and exhibited no significant improvements after iteration 48 even if the highest accuracy value in fold 6(c) is fulfilled at about iteration 58. However, the average iteration in the four folds GWO needs to achieve the highest fitness value is still about 48.

To verify the effectiveness of the proposed model, GWO-KELM was compared against the other methods on the same Wieslaw dataset in perspective of Train accuracy, Validate accuracy, Test accuracy, Type I error, Type II error and AUC via external 10-fold CV strategy. The detailed experimental results of four optimized-KELMs including GWO-KELM, PSO-KELM, GA-KELM, and GS-KELM are shown in Table 3 in terms of the Tr-acc, Va-acc, Te-acc, Ty-I-e, Ty-II-e, and AUC on this Wieslaw dataset. It can be observed that Tr-acc, Va-acc, Te-acc and AUC obtained by GWO-KELM are clearly better than others achieved by the comparative models. Another

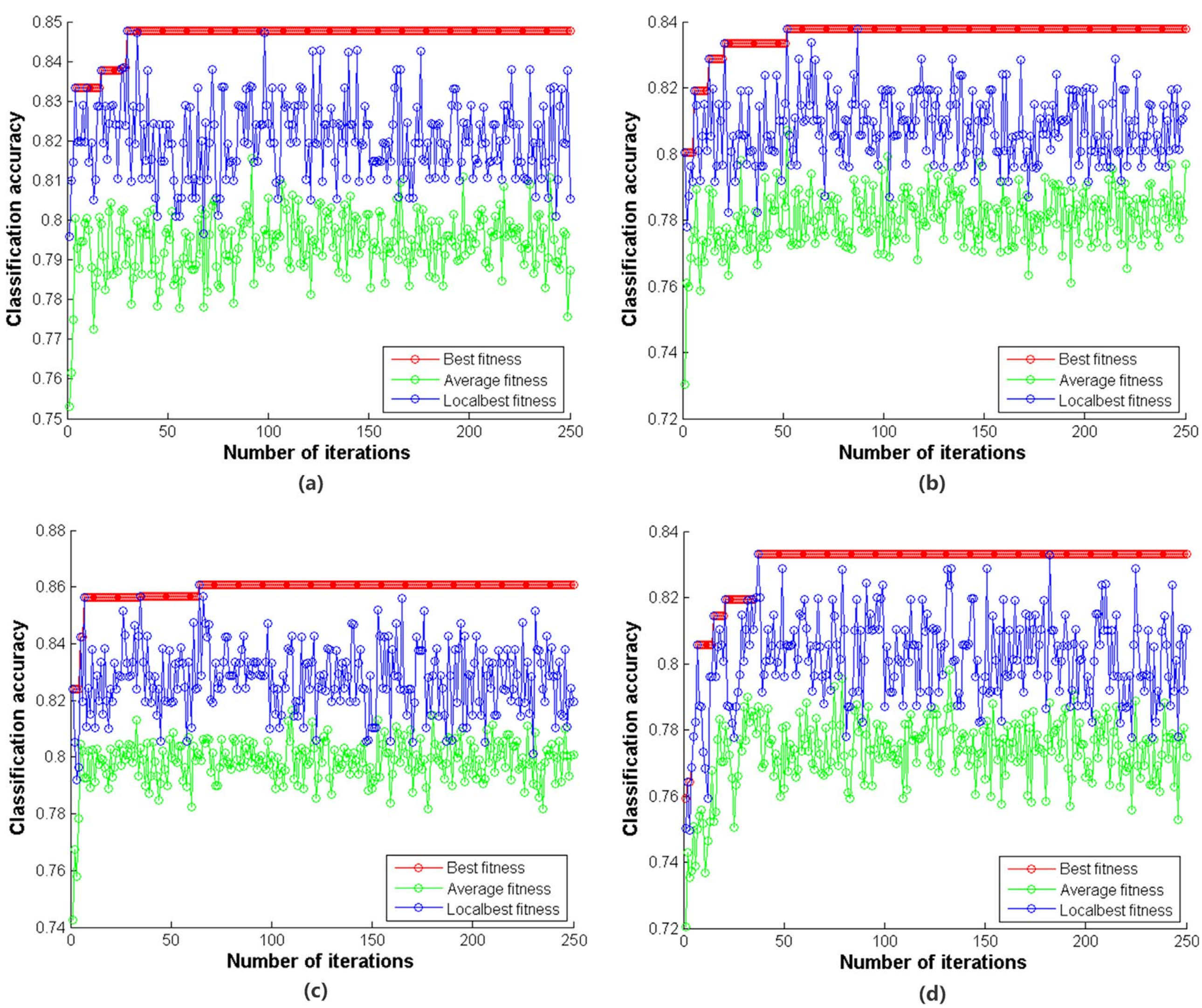

Fig. 7. Learning curves during the training stage for fold 2(a), fold 4(b), fold 6(c) and fold 8(d) in 10-fold CV on Wieslaw dataset.

Table 3
The detailed results of the all KELM models on the Wieslaw dataset.

Method	Metrics					
	Tr-acc	Va-acc	Te-acc	Ty-I-e	Ty-II-e	AUC
GWO-KELM	0.9922 ± 0.0079	0.8639 ± 0.0057	0.8458 ± 0.0281	0.1032 ± 0.0339	0.1787 ± 0.0612	0.8573 ± 0.0604
PSO-KELM	0.9792 ± 0.0251	0.8513 ± 0.0286	0.8250 ± 0.0938	0.1368 ± 0.0938	0.2023 ± 0.0952	0.8250 ± 0.102
GA-KELM	0.9583 ± 0.019	0.8428 ± 0.0125	0.8167 ± 0.0657	0.1525 ± 0.0608	0.2025 ± 0.0539	0.8222 ± 0.0951
GS-KELM	0.9549 ± 0.0234	0.8520 ± 0.025	0.8000 ± 0.0383	0.1721 ± 0.0463	0.2072 ± 0.0802	0.8130 ± 0.0819

Table 4
Confusion matrix obtained by the four methods on Wieslaw dataset.

Methods	Output/target	Bankruptcy	Non-Bankruptcy	Accuracy (%)
GWO-KELM	Bankruptcy	101	23	90.18
	Non-Bankruptcy	11	105	82.03
	Total	112	128	**84.58**
PSO-KELM	Bankruptcy	95	25	84.82
	Non-Bankruptcy	17	103	80.47
	Total	112	128	82.50
GA-KELM	Bankruptcy	93	25	83.04
	Non-Bankruptcy	19	103	80.47
	Total	112	128	81.67
GS-KELM	Bankruptcy	89	25	79.46
	Non-Bankruptcy	23	103	80.47
	Total	112	128	80.00

competitive advantage of GWO-KELM is that the Ty-I-e, and Ty-II-e obtained by GWO-KELM are less than corresponding values achieved by PSO-KLEM, GA-KELM and GS-KELM. GWO-KELM has obtained the best value with Tr-acc of 0.9922, Va-acc of 0.8639, Te-acc of 0.8458, Ty-I-e of 0.1032, Ty-II-e of 0.1787, and AUC of 0.8573. PSO-KELM is next to GWO-KELM, which yielded Tr-acc of 0.9792, Va-acc of 0.8513, Te-acc of 0.8250, Ty-I-e of 0.1368, Ty-II-e of 0.2023, and AUC of 0.8250. GA-KELM yielded Tr-acc of 0.9583, Va-acc of 0.8428, Te-acc of 0.8167, Ty-I-e of 0.1525, Ty-II-e of 0.2025, and AUC of 0.8225. Followed by GS-KELM, this model yielded Tr-acc of 0.9549, Va-acc of 0.8520, Te-acc of 0.8000, Ty-I-e of 0.1721, Ty-II-e of 0.2072, and AUC of 0.8130. Meanwhile, an intuitive fact can be observed that the classification performance of PSO-KELM is slightly better than GA-KELM. Based on the above analysis, a conclusion can be made that the GWO-based KELM obtains not only the highest Tr-acc, Va-acc and Te-acc, but also the smallest Ty-I-e and Ty-II-e as well. In terms of AUC, the GWO-KELM also has the highest value, which means this model has the best classification capability among all KELM models. From the above analysis, we can deduct that the KELM has gained a lot benefits from the GWO strategy owing to its unique global search ability. In other words, the most suitable parameter set of KELM have been identified with the aid of GWO.

Table 4 lists the detailed classification results of the four methods in terms of confusion matrix. From the fact revealed in the table, a phenomenon can be clearly observed that GWO-KELM achieves the best classification results with the highest average correct classification rate of 84.58% among the four methods, there are only 11 bankrupt enterprises misclassified as non-bankrupt ones, and 23 non-bankrupt enterprises misclassified as bankrupt ones. Compared with GWO-KELM, PSO-KELM, GA-KELM and GS-KELM show consistently inferior results in respect of the accurate classification rate with 17, 19, 23 bankrupt companies being misclassified as non-bankrupt ones, and with 25, 25, 25 non-bankrupt enterprises being misclassified as bankrupt ones, respectively. It reveals that GWO-KELM has the best capability of discriminating the bankrupt companies from the healthy ones.

The detailed comparison results of the four methods from the point of view of Tr-acc, Va-acc, Te-acc, Ty-I-e, Ty-II-e, and AUC and standard deviations on Wieslaw dataset are presented in Fig. 8. From this figure, an obvious conclusion can be made that the GWO-KELM acquire not only the best property in terms of the six metrics than other KELM models, but also the smallest standard deviations, which means the best classification accuracy and the robustness can be met synchronously. Furthermore, the PSO-KELM method produced the largest standard deviation value among the four methods with respect to the six performance metrics. In a nutshell, we can draw the conclusion that the proposed GWO-KELM method has not only higher classification accuracy but also good robustness as well.

For the purpose of exploring the efficiency of the proposed method, we have also compared the computational time of the four methods. Fig. 9 plots the CPU time of each method in detail, where the value of vertical axis means the current total time with the increase of the fold number. It can be seen that GWO-KELM

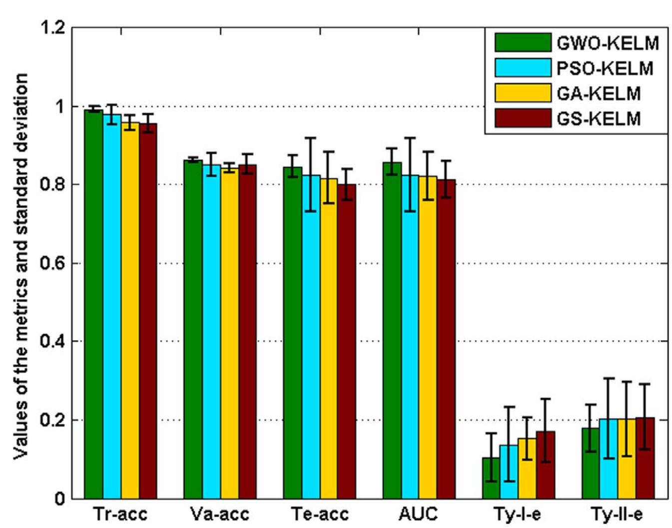

Fig. 8. The average Tr-acc, Va-acc, Te-acc, Ty-I-e, Ty-II-e, AUC and standard deviations obtained by GWO-KELM, PSO-KELM, GA-KELM and GS-KELM on Wieslaw dataset.

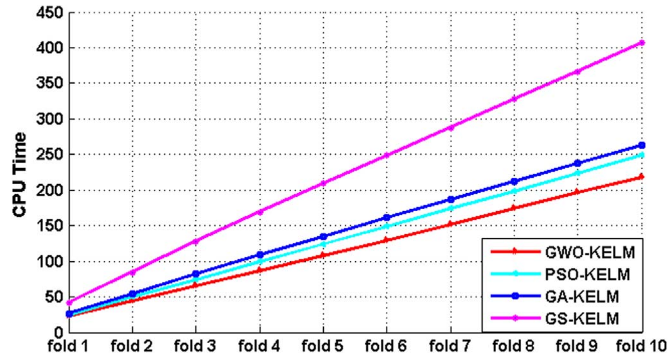

Fig. 9. Comparison results among the four methods in terms of CPU time via 10-fold CV.

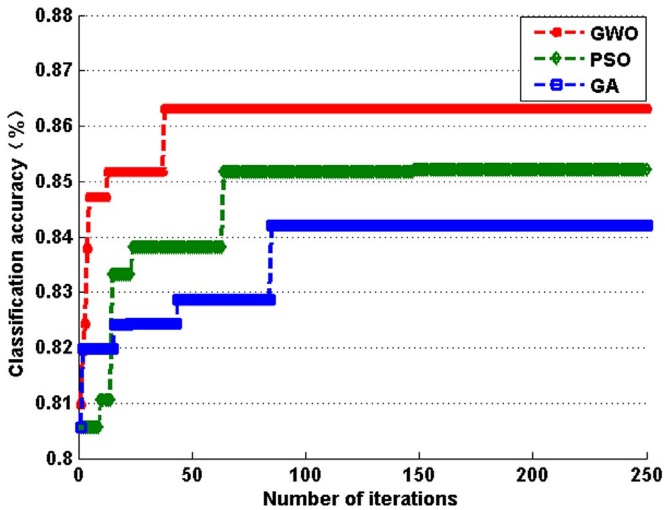

Fig. 10. The average results of best fitness during the training stage in one run of 10-fold CV obtained by GWO-KELM, PSO-KELM and GA-KELM on Wieslaw dataset.

needs the least CPU time among the four methods, while GS-KELM appears to the most time consuming among all methods; PSO-KELM needs almost the same CPU time as that of GA-KELM. It is meaningful to note that GWO-KELM took only about 21.77 s on average across the 10 folds, even when there are so many generations involved. However, GS-KELM, PSO-KELM and GA-KELM consumed 40.65 s, 24.85 s and 26.31 s respectively when the same generations and swarm size were considered. From the above analysis, we can reach a conclusion that GWO-KELM is more appropriate for being as a viable early warning tool for bankruptcy due to its efficiency.

In order to explore the optimization procedures of the meta-heuristic optimization methods including GWO-KELM, PSO-KELM and GA-KELM, we have recorded the evolutionary process of the three methods. Fig. 10 displays the average evolution results of the objective function values, namely, the mean results of the best validation fitness obtained by GWO-KELM, PSO-KELM and GA-KELM across 10 folds respectively on Wieslaw dataset. As shown, the trends of the three fitness curves gradually rise from iteration 1 to 250 and displayed no significant amelioration after iteration 48, 65 and 82 with respect to GWO-KELM, PSO-KELM and GA-KELM, respectively. Therefore a rigorous fact may be found that GWO-KELM considerably outperformed other two methods in terms of the rate of convergence. In other words, GWO-KELM can converge with a fast rate to the global optima, and find the best solution very efficiently. Another obvious experimental result can be found from Fig. 9 is that the maximum accuracy achieved by GWO-KELM is up to 86.30%, which is the highest among the three

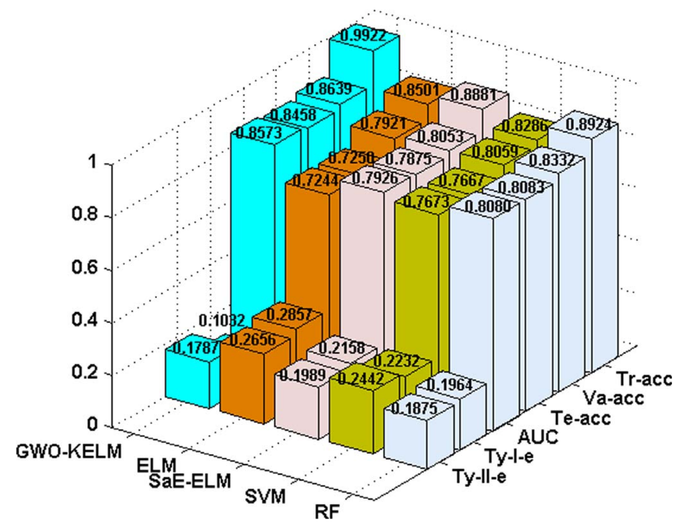

Fig. 11. The average of ACC, AUC, Type I error and Type II error obtained by GWO-KELM, ELM, SaE-ELM, SVM and RF on the Wieslaw dataset.

methods. It indicates the superiority of GWO-KELM over PSO-KELM and GA-KLEM in discriminating the bankrupt enterprises from the non-bankrupt ones.

To further illustrate the advantageous characteristics of the proposed model (GWO-KELM), we evaluated other four competitive bankruptcy prediction models based on the advanced machine learning methods including ELM, SaE-ELM, SVM and RF, using the same Wieslaw dataset combined with an external 10-fold CV scheme. Fig. 11 shows the average experimental results of GWO-KELM, ELM, SaE-ELM, SVM and RF in terms of mean of Tr-acc, Va-acc, Te-acc, Ty-I-e, Ty-II-e and AUC, respectively. It is observed that GWO-KELM achieves the best values with Tr-acc in 0.9922, Va-acc in 0.8639, Te-acc in 0.845, AUC in 0.8573, Ty-I-e in 0.1032 and Ty-II-e in 0.1787 respectively. Compared with ELM, SaE-ELM, SVM and RF, the mean classification performance of GWO-KELM is raised 12.08%, 7.91%, 5.83% 3.75% and 13.29%, 9%, 6.47%, 4.93% respectively in terms of Te-acc and AUC, while the values of Ty-I-e and Ty-II-e are lower than those of ELM, SaE-ELM, SVM and RF by 18.57%, 12%, 9.32% and 8.69%, 6.35%, 0.88%, respectively. In addition, GWO-KELM also has obvious superiorities over other four competitive bankruptcy prediction models in terms of Tr-acc and Va-acc on this dataset. Another interesting fact is that ELM with self-adaptive evolutionary strategy proposed by Cao et al. (2012), namely, SaE-ELM, is better than the original ELM in all metrics on this dataset. Based on the detailed comparative analyses, it can be concluded that the proposed GWO-KELM methodology provides better classification performance than other competitive methods on the Wieslaw dataset.

To further explore the ability of the proposed method, parameter optimization and feature selection were conducted simultaneously (GWOFS-KELM) through 10-fold CV. Fig. 12 shows the number of times that each feature was selected. It can be seen from this figure that two categories bars exists. Some features were selected in a large amount, and others were selected in a small amount. A reasonable assumption can be made that features selected over 7 times during the entire folds are pivotal features. Therefore, the features X_1, X_3, X_9, X_{17}, X_{21}, and X_{29} are critical features in distinguish the bankruptcy banks from the healthy ones. In order to further verify the results, two financial analysts with a wealth of market experience had been involved in this study. According to their professional analysis, a fact can be observed that features X_1, X_3, X_9, X_{17}, X_{21}, and X_{29} are major concern in the practical cases, especially the features X_1(cash/current liabilities), X_9(net profit/total assets) and X_{29}($EBIT^*$/total assets) can

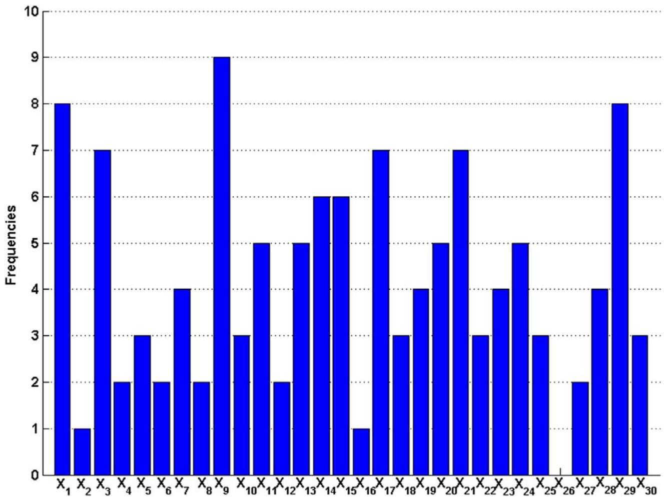

Fig. 12. Frequencies of selected features via 10-fold CV on Wieslaw dataset.

Table 5
Paired *t*-test results of GWOFS-KELM and GWO-KELM in terms of the six metrics on the Wieslaw dataset.

Methods	*p*-value (significance)					
	Tr-acc	Va-acc	Te-acc	Ty-I-e	Ty-II-e	AUC
GWO-KELM	**2.559 (0.031)**	1.887 (0.092)	**2.477 (0.035)**	**−1.571 (0.041)**	**−2.845 (0.019)**	**2.719 (0.024)**

be paid much more attentions. This also implies that practical engineers should pay more attention to the valuable information given by these features.

In order to reveal subtle differences in the classification performances of the proposed methods, a paired *t*-test was carried out. A *p*-value of less than 0.05 indicates statistical significance in the experiment. Table 5 shows that GWOFS-KELM has almost significantly better results than the GWO-KELM in terms of Tr-acc, Va-acc, Te-acc, Ty-I-e, Ty-II-e, and AUC. In Table 5, significant values that are greater than 0.05 are shown in bold. Positive values mean that the *i*th classifier fulfills better properties than the *j*th one.

5.2. Japanese bankruptcy dataset (JPNBDS)

The financial ratios of JPNBDS are abstracted from the Japanese true financial statements in the years of 1995–2009. This dataset consists of two types including Non-Bankrupt or Bankrupt. The financial ratios for JPNBDS used in this study are selected according to the principle where the financial ratios should be frequently used in the previous literatures described by Ravi Kumar and Ravi (2007). Eventually, ten most commonly used financial ratios are selected as the features. Features R_1, R_2, R_4, R_5, R_6, R_9 and R_{10} correspond to $\mathbf{X_9}$, $\mathbf{X_3}$, X_5, $\mathbf{X_{29}}$, $\mathbf{X_{17}}$, X_4, and $\mathbf{X_1}$ features in the Wieslaw dataset, while features R_3, R_7, and R_8 are absent in the Wieslaw dataset. Descriptions of the ten features and statistical information of each feature in terms of the maximum (max) and minimum of (min), mean and standard deviation (std) value are given in Table 6.

Table 7 below shows detailed results of the Tr-acc, Va-acc, Te-acc, Ty-I-e, Ty-II-e and AUC for each fold of the 10-fold CV and corresponding pair values of *C* andγ obtained by GWO-KELM on JPNBDS. It can be observed from Table 6 that GWO-KELM achieves better performance for the average and standard deviations of Tr-acc, Va-acc, Te-acc, AUC, Ty-I-e and Ty-II-e, respectively.

Fig. 13 exhibits the detailed scores obtained by α, β and δ wolf with the increase of iteration numbers from 1 to 250 on the JPNBDS during the training step. An interesting fact observed from this figure is that α wolf has a fast convergence towards the best score at approximately iteration 58, which means that the convergence rate of α is 58 iterations.

Furthermore, learning curves of the proposed GWO-KELM on JPNBDS, in terms of the Best validation fitness, Average validation fitness, and Local best validation fitness during the training stage, are exhibited in Fig. 14 for fold 2(a), fold 4(b), fold 6(c) and fold 8 (d) via external 10-fold CV. As seen from this figure, fold 2(a), fold 4(b), and fold 6(c) all possess the same phenomenon where the best accuracy curve gradually improves from iteration 1 to about 58 and steadies afterwards, while the highest accuracy value for fold 8(d) appears at less 50 iterations. Therefore, it is clear that the

Table 6
The statistical information of the JPNBDS.

Attributes	Max	Min	Mean	Std
R_1 (net profit/total assets)	1.8404	−3.3847	−0.1175	0.4579
R_2 (current assets/current liabilities)	8.7813	0.0065	1.2332	1.0495
R_3 (retained earnings/total assets)	0.8351	−40.785	−0.3194	3.3389
R_4 (working capital/total assets)	0.5759	−26.017	−0.2346	2.1345
R_5 (EBIT/total assets)	0.2139	−0.4935	0.0074	0.0883
R_6 (sales/total assets)	2.7770	0.0437	1.0163	0.4544
R_7 (market value of equity/total debt)	3.8076	0.0014	0.2868	0.5075
R_8 (cash flow/total debt)	0.9460	0.0399	0.5188	0.1973
R_9 (current assets/total assets)	8.0495	−0.9630	0.6341	1.1311
R_{10} (cash/current liabilities)	3.8076	0.0014	0.3336	0.5246

Table 7
The detailed results obtained by GWO-KELM model with optimal parameter pair on JPNBDS.

Fold	(C, γ)	Tr-acc	Va-acc	Te-acc	Ty-I-e	Ty-II-e	AUC
#1	(21.4633 1.2332)	0.8978	0.8833	0.8667	0.125	0.1429	0.8661
#2	(27.8731 7.3152)	0.9283	0.8743	0.8750	0.125	0.1250	0.8750
#3	(18.8705 18.870)	0.9315	0.8568	0.8763	0.1109	0.1313	0.8789
#4	(17.9129 2.2120)	0.9210	0.8728	0.8550	0.1159	0.1550	0.8646
#5	(31.1282 2.1154)	0.9106	0.8841	0.8667	0.1428	0.1250	0.8661
#6	(32.000 4.51375)	0.9264	0.8637	0.8563	0.1230	0.1230	0.8626
#7	(19.7269 1.7551)	0.8944	0.8782	0.8425	0.1637	0.1407	0.8478
#8	(29.4978 1.9546)	0.9273	0.8745	0.8808	0.0825	0.1263	0.8956
#9	(18.4839 1.6074)	0.9273	0.8767	0.8754	0.1344	0.0948	0.8854
#10	(29.4978 1.9546)	0.9037	0.8625	0.8625	0.1087	0.1525	0.8693
Avg.		**0.9168**	**0.8727**	**0.8657**	**0.1232**	**0.1317**	**0.8711**
Dev.		**0.0140**	**0.0090**	**0.0119**	**0.0217**	**0.0174**	**0.0133**

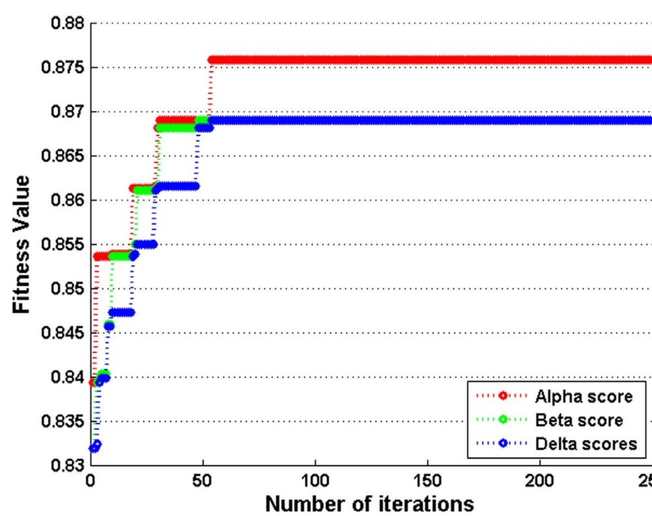

Fig. 13. The scores of Alpha, Beta, and Delta obtained by GWO-KELM on JPNBDS.

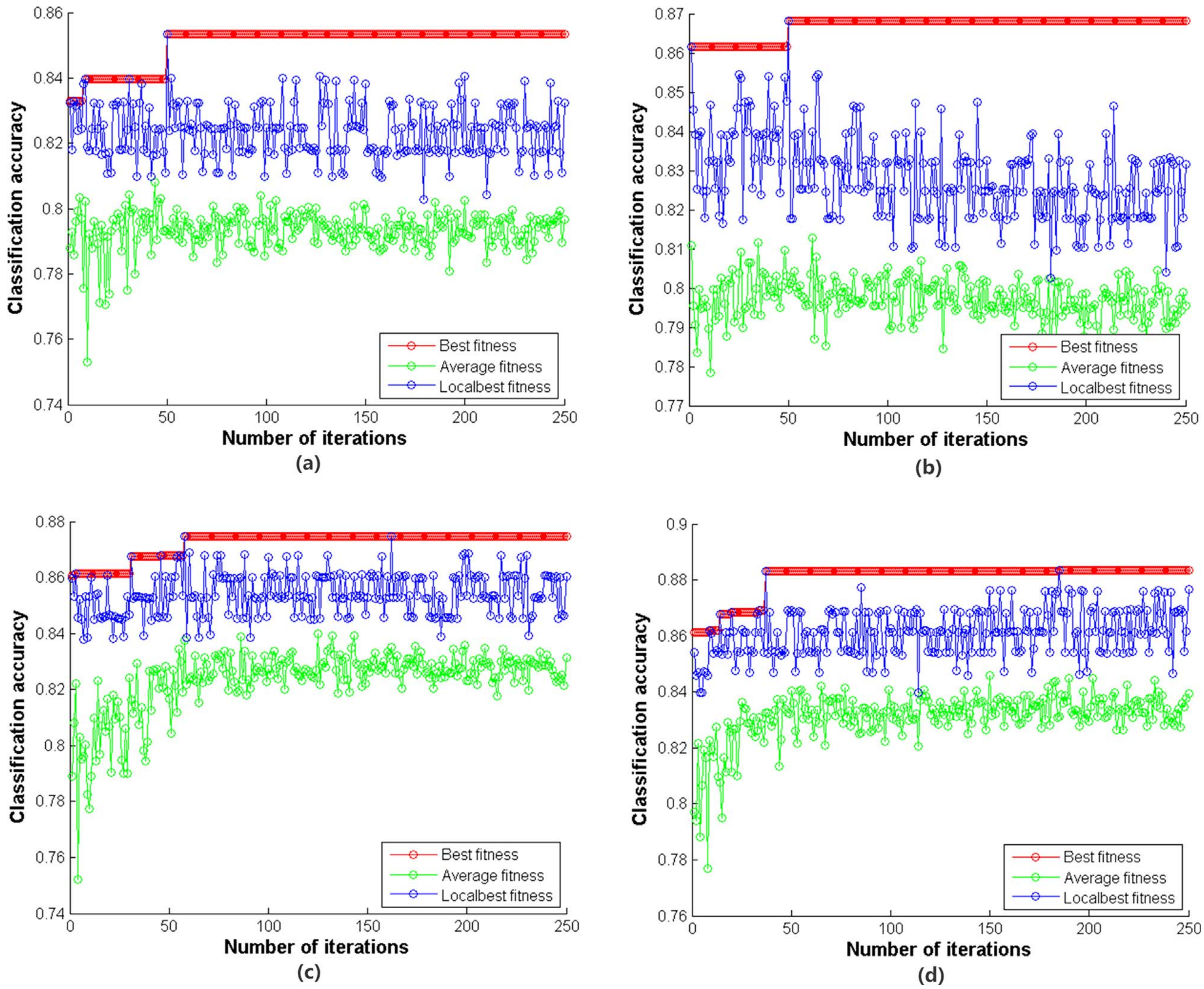

Fig. 14. Learning curves during the training stage for fold 2(a), fold 4(b), fold 6(c) and fold 8(d) in 10-fold CV on JPNBDS.

Table 8
The detailed results of the all KELM models on the JPNBDS.

Method	Metrics					
	Tr-acc	Va-acc	Te-acc	Ty-I-e	Ty-II-e	AUC
GWO-KELM	0.9168 ± 0.0140	0.8727 ± 0.0090	0.8657 ± 0.0119	0.1232 ± 0.0217	0.1317 ± 0.0174	0.8711 ± 0.0133
PSO-KELM	0.9092 ± 0.0234	0.8623 ± 0.0125	0.8492 ± 0.0356	0.1256 ± 0.0456	0.1453 ± 0.0256	0.8645 ± 0.0276
GA-KELM	0.9083 ± 0.0300	0.8501 ± 0.0234	0.8021 ± 0.0452	0.1609 ± 0.0321	0.2141 ± 0.0346	0.8125 ± 0.0485
GS-KELM	0.8849 ± 0.0245	0.8520 ± 0.0145	0.8154 ± 0.0255	0.1617 ± 0.0256	0.1794 ± 0.0254	0.8294 ± 0.0231

average iteration for the four folds GWO to achieve the highest fitness value overall is about 58.

Detailed experimental results of four optimized-KELMs including GWO-KELM, PSO-KELM, GA-KELM, and GS-KELM are shown in Table 8 in terms of Tr-acc, Va-acc, Te-acc, Ty-I-e, Ty-II-e, and AUC on the JPNBDS dataset. It can be observed that Tr-acc, Va-acc, Te-acc and AUC obtained by GWO-KELM are better than those achieved by the three comparative KELM models. A further competitive advantage of GWO-KELM is that the values of Ty-I-e and Ty-II-e of GWO-KELM are smaller than those of PSO-KLEM, GA-KELM and GS-KELM. GWO-KELM achieves the best values with Tr-acc of 0.9168, Va-acc of 0.8727, Te-acc of 0.8657, Ty-I-e of 0.1232, Ty-II-e of 0.1317, and AUC of 0.8711, respectively. PSO-KELM is second in the rank, followed by GA-KELM, and GS-KELM. In view of Tr-acc, Va-acc, and Te-acc, GWO-KELM has the best ability to categorize the cases into correct classes. In terms of AUC, GWO-KELM also has the highest value among all methods compared, which indicates that GWO-KELM has the highest probability to identify the true positives cases from the negative ones. As a by-product of the experiment, it can be observed that the classification performance of GS-KELM is slightly better than that of GA-KELM in six metrics.

Based on the observations, it can be concluded that the GWO-based KELM obtains not only the highest Tr-acc, Va-acc, Te-acc, and AUC, but also the smallest Ty-I-e and Ty-II-e. The GWO strategy brings advantages into the KELM due to its unique global search ability to acquire the most suitable parameter set of KELM.

Table 9 exhibits the detailed classification results of the four methods in terms of confusion matrix on JPNBDS. It can be seen from this table that GWO-KELM achieves the best classification

Table 9
Confusion matrix obtained by the four methods on JPNBDS.

Methods	Output/target	Bankruptcy	Non-Bankruptcy	Accuracy (%)
GWO-KELM	Bankruptcy	67	11	88.16
	Non-Bankruptcy	9	65	85.53
	Total	76	76	**86.84**
PSO-KELM	Bankruptcy	65	12	85.53
	Non-Bankruptcy	11	64	84.21
	Total	76	76	84.87
GA-KELM	Bankruptcy	64	18	84.21
	Non-Bankruptcy	12	58	76.32
	Total	76	76	80.23
GS-KELM	Bankruptcy	63	15	82.89
	Non-Bankruptcy	13	61	80.26
	Total	76	76	81.58

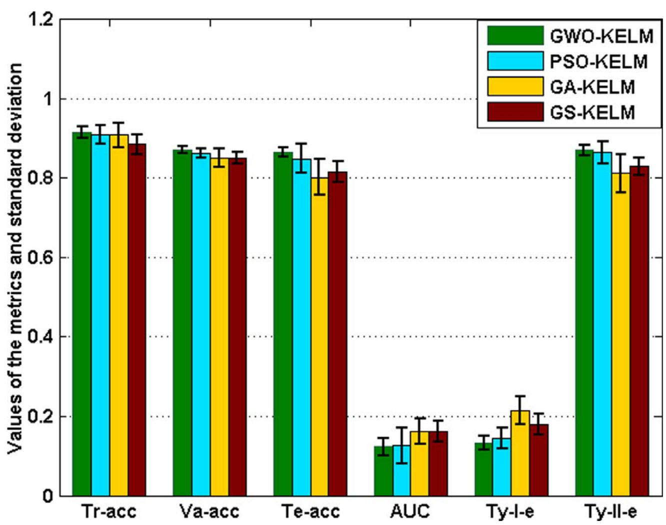

Fig. 15. The average Tr-acc, Va-acc, Te-acc, Ty-I-e, Ty-II-e, AUC and standard deviations obtained by GWO-KELM, PSO-KELM, GA-KELM and GS-KELM on JPNBDS.

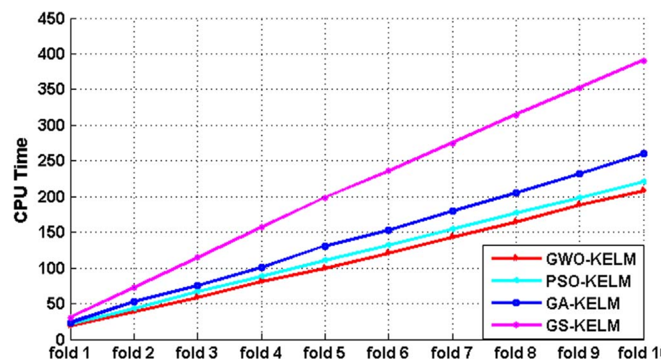

Fig. 16. Comparison results among the four methods in terms of CPU time via 10-fold CV on JPNBDS.

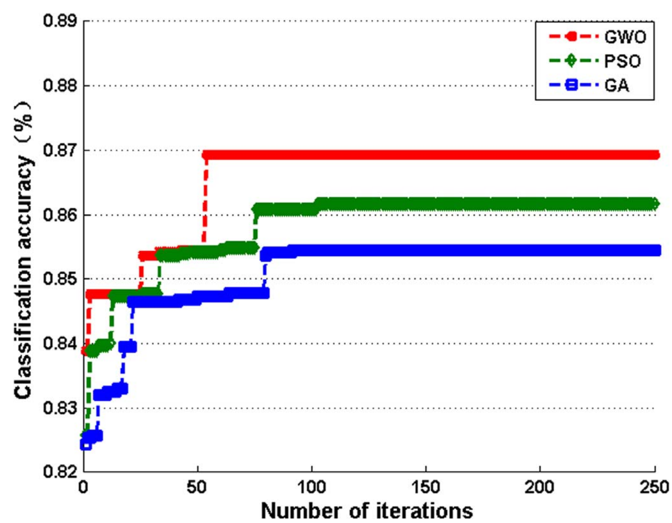

Fig. 17. The average results of best fitness during the validation stage in one run of 10-fold CV obtained by GWO-KELM, PSO-KELM and GA-KELM on JPNBDS.

results with the highest average rate of 86.84% for correct classification, there are only 11 bankrupt enterprises being misclassified as non-bankruptcy ones and 9 non-bankruptcy enterprises misclassified as bankrupt ones. PSO-KELM, GA-KELM and GS-KELM show consistently inferior results in respect of the classification accuracy. Overall, GWO-KELM has the best ability to discriminate the bankrupt companies from the healthy ones.

Fig. 15 presents the detailed comparison results of the four methods from the point of view of Tr-acc, Va-acc, Te-acc, Ty-I-e, Ty-II-e, and AUC and their standard deviations on JPNBDS. As shown in the figure, GWO-KELM acquires not only the best property in terms of the six metrics, but also the smallest standard deviations, thus achieves the best classification accuracy and the robustness synchronously. It is also interesting to note that GA-KELM has the biggest standard deviations values in the group in terms of Tr-acc, Va-acc, Te-acc, and Ty-II-e, and AUC. It implies that the robustness of GA-KELM might be poorer than that of others. Form Fig. 13, it can be concluded that the proposed GWO-KELM method has not only higher classification accuracy, but also good robustness in discriminating the bankruptcy enterprises from the non-bankruptcy ones.

The computational times of the four methods on JPNBDS are shown in Fig. 16. It can be seen that GS-KELM appears to consume the most time among all methods and GWO-KELM uses the least CPU time of only 208.27 s for the whole external 10-fold CV. This indicates that the proposed GWO-KELM can be used as a potential early warning tool for bankruptcy prediction due to its efficiency in the financial field.

The optimization procedures of the meta-heuristic optimization methods including GWO-KELM, PSO-KELM and GA-KELM are all recorded in Fig. 17. In this figure, the evolutionary process curves of the three methods display the average evolution result of the best validation fitness obtained by GWO-KELM, PSO-KELM, and GA-KELM respectively on JPNBDS. It can be seen from this figure that the trends of the three fitness curves gradually rise from iteration 1 to 250, and then show no significant amelioration after iteration 59, 101 and 84 for GWO-KELM, PSO-KELM and GA-KELM, respectively. Though the fitness of PSO-KELM still changes after 83 iterations, the change is nevertheless subtle. In summary, GWO-KELM is faster to achieve global optima than the two other

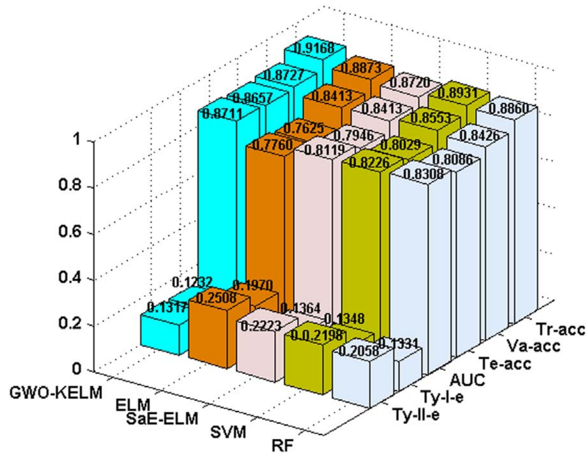

Fig. 18. The average ACC, AUC, Type I error and Type II error obtained by GWO-KELM, ELM, SaE-ELM, SVM and RF on the JPNBDS.

methods, thus has better capability of discriminating the bankrupt enterprises from the non-bankrupt ones.

In order to further evaluate the performance of GWO-KELM, other four bankruptcy prediction models ELM, SaE-ELM, SVM and RF were used for comparison on the JPNBDS dataset. The detailed experimental results of these methods are shown in Fig. 18 in terms of mean of Tr-acc, Va-acc, Te-acc, Ty-I-e, Ty-II-e and AUC, respectively. It can be observed that GWO-KELM has the best values among these methods with Tr-acc of 0.9168, Va-acc of 0.8727, Te-acc of 0.8657, AUC of 0.8711, Ty-I-e of 0.1232 and Ty-II-e of 0.1317 respectively. ELM yields Tr-acc of 0.8873, Va-acc of 0.8413, Te-acc of 0.7625, AUC of 0.7760, Ty-I-e of 0.1970 and Ty-II-e of 0.2508. SaE-ELM yields Tr-acc of 0.8720, Va-acc of 0.8413, Te-acc of 0.7946, AUC of 0.8199, Ty-I-e of 0.1364 and Ty-II-e of 0.2223. Due to the self-adaptive evolutionary strategy introduced, SaE-ELM has obvious superiority over the original ELM. SVM yields Tr-acc of 0.8931, Va-acc of 0.8553, Te-acc of 0.8029, AUC of 0.8226, Ty-I-e of 0.1348 and Ty-II-e of 0.2198. And RF obtains Tr-acc of 0.8860, Va-acc of 0.8426, Te-acc of 0.8086, AUC of 0.8305, Ty-I-e of 0.1331 and Ty-II-e of 0.2058 respectively. Hence, it is justified that the proposed GWO-KELM methodology provides the most promising classification performance for the comparison study on the JPNBDS dataset.

Fig. 19 shows the number of times that each feature was selected through the run of 10-fold CV. This figure shows that the

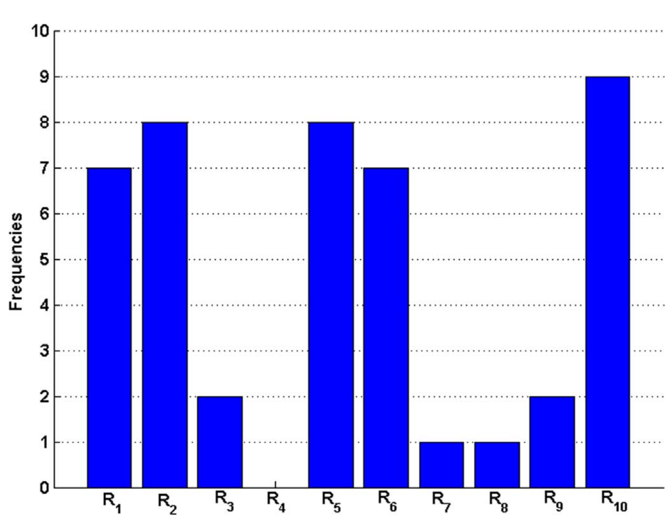

Fig. 19. Frequencies of selected features via 10-fold CV on the JPNBDS.

Table 10
Paired *t*-test results of MFOFS-KELM and GWO-KELM in terms of these six metrics on the JPNBDS.

Methods	*p*-value (significance)					
	Tr-acc	**Va-acc**	**Te-acc**	**Ty-I-e**	**Ty-II-e**	**AUC**
GWO-KELM	**5.394 (0.000)**	**7.653 (0.000)**	**3.448 (0.007)**	−1.490 (0.170)	**−2.670 (0.026)**	**4.916 (0.001)**

bars can be divided into two categories. The first category encompasses the features that are selected in a large amount. The second category contains the features which are selected in a small amount. Based on the feature selection results of GWOFS-KELM, the features R_1, R_2, R_5, R_6, and R_{10} were chosen as the important features to identify the bankruptcy banks from the healthy ones. Based on the analysis of the financial analysts, R_2 (current assets/current liabilities), R_5 (EBIT/total assets) and R_{10} (cash/current liabilities) are really the key factors of affecting bankruptcy problems in the actual cases. The suggestion was also given that these features should be paid more attention in practical case, which means that these features can give valuable information.

A paired *t*-test was also conducted on the JPNBDS for validating the significance of the proposed method. In this experiment, a *p*-value less than 0.05 were considered statistically significant. Table 10 shows that GWOFS-KELM has nearly significantly better results than the GWO-KELM in terms of Tr-acc, Va-acc, Te-acc, Ty-I-e, Ty-II-e, and AUC. In the table, positive values represent that the *i*th classifier has fulfilled a better property than the *j*th one.

6. Conclusions and future work

In this study, we have constructed an accurate and robust methodology, GWO-KELM, to accurately discriminate the enterprises with bankruptcy from the non-bankruptcy ones. The main innovation of this paper lies in using one of the latest swarm intelligence algorithms to maximize the generalization capability of KELM classifier by exploring the latest swarm intelligence technique for optimal parameter tuning for bankruptcy prediction. The experimental results have demonstrated that the evidential superiority of the proposed GWO-KELM over other six advanced bankruptcy prediction models in terms of the Tr-acc, Va-acc, Te-acc, Ty-I-e, Ty-II-e and AUC for two real life datasets with different financial ratios. The GWO-KELM can perform better when the parameter optimization and feature selection are conducted simultaneously. Therefore, it can be concluded that the proposed GWO-KELM can be adopted as a potential alternative early warning system in financial decision-making to assist the financial institution in making accurate decisions.

However, it should be noted that only 240 or 152 samples were involved in this experiment. Datasets with large data samples are needed to further verify the effectiveness of the proposed methodology. Furthermore, it was observed that both feature selection and parameter setting played a pivotal role in ameliorating the performance of KELM as they influence each other. Therefore, it is planned as a further research direction to implement the feature selection and the parameter optimization simultaneously using GWO to further improve the property of KELM for more datasets or other fields. Additionally, it may be also helpful to try other promising meta-heuristics such as fruit fly optimizer (Pan, 2012; Shen et al., 2016) and multi-verse optimizer (Mirjalili et al., 2015) to optimize the parameters of KELM.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (NSFC) under Grant nos. of 61303113, 61373166, 61402337 and 61571444. This research is also funded by the Zhejiang Provincial Natural Science Foundation of China under Grant nos. of LY17F020012, LQ13G010007, LY14F020035 and LQ13F020011, the Guangdong Natural Science Foundation under Grant no. of 2016A030310072, the Science and Technology Plan Project of Wenzhou of China under Grant nos. of G20140048 and H20110003.

References

Abellán, J., Mantas, C.J., 2014. Improving experimental studies about ensembles of classifiers for bankruptcy prediction and credit scoring. Expert Syst. Appl. 41 (8), 3825–3830.
Altman, E.I., 1968. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. J. Financ. 23 (4), 589–609.
Atiya, A.F., 2001a. Bankruptcy prediction for credit risk using neural networks: a survey and new results. IEEE Trans. Neural Netw. 12 (4), 929–935.
Atiya, A.F., 2001b. Bankruptcy prediction for credit risk using neural networks: a survey and new results. IEEE Trans. Neural Netw. 12 (4), 929–935.
Bartlett, P.L., 1998. 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.
Beaver, W.H., 1966. Financial ratios as predictors of failure. J. Account. Res. 4, 71–111.
Cao, J., et al., 2015. An enhance excavation equipments classification algorithm based on acoustic spectrum dynamic feature. Multidimens. Syst. Signal Process., 1–23.
Cao, J., et al., 2016a. Extreme learning machine and adaptive sparse representation for image classification. Neural Netw. 81, 91–102.
Cao, J., et al., 2016b. Excavation equipment recognition based on novel acoustic statistical features. IEEE Trans. Cybern. 99, 1–13.
Cao, J., Lin, Z., Huang, G.-B., 2012. Self-adaptive evolutionary extreme learning machine. Neural Process. Lett. 36 (3), 285–305.
Chang, C.-C., Lin, C.-J., 2011. LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. 2 (3), 1–27.
Chauhan, N., Ravi, V., Chandra, D.K., 2009. Differential evolution trained wavelet neural networks: application to bankruptcy prediction in banks. Expert Syst. Appl. 36 (4), 7659–7665.
Chen, C., et al., 2014. Spectral-spatial classification of hyperspectral image based on kernel extreme learning machine. Remote Sens. 6 (6), 5795–5814.
Chen, H., et al., 2015. Using blood indexes to predict overweight statuses: an extreme learning machine-based approach. PLoS One 10 (11), e0143003.
Chen, H.L., et al., 2011. A novel bankruptcy prediction model based on an adaptive fuzzy k-nearest neighbor method. Knowl. Based Syst. 24 (8), 1348–1359.
Chen, H.L., et al., 2011b. An adaptive fuzzy K-nearest neighbor method based on parallel particle swarm optimization for bankruptcy prediction. In: Huang, J., Cao, L., Srivastava, J. (Eds.), Advances in Knowledge Discovery and Data Mining. Springer, Berlin/Heidelberg, pp. 249–264.
Chen, H.L., et al., 2016. An efficient hybrid kernel extreme learning machine approach for early diagnosis of Parkinson's disease. Neurocomputing 184(C), 131–144.
Chen, H.-L., et al., 2011a. A novel bankruptcy prediction model based on an adaptive fuzzy k-nearest neighbor method. Knowl. Based Syst. 24 (8), 1348–1359.
Deng, W.-Y., Zheng, Q.-H., Wang, Z.-M., 2014. Cross-person activity recognition using reduced kernel extreme learning machine. Neural Netw. 53, 1–7.
Fawcett, T., 2006. An introduction to ROC analysis. Pattern Recognit. Lett. 27 (8), 861–874.
Fedorova, E., Gilenko, E., Dovzhenko, S., 2013. Bankruptcy prediction for Russian companies: application of combined classifiers. Expert Syst. Appl. 40 (18), 7285–7293.
Guang-Bin, H., Babri, H.A., 1998. Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions. IEEE Trans. Neural Netw. 9 (1), 224–229.
Guang-Bin, H., Qin-Yu, Z., Chee-Kheong, S., 2004. Extreme learning machine: a new learning scheme of feedforward neural networks. In: Proceedings of the 2004 IEEE International Joint Conference on Neural Networks.
Hornik, K., 1991. Approximation capabilities of multilayer feedforward networks. Neural Netw. 4 (2), 251–257.
Huang, G.-B., et al., 2012. Extreme learning machine for regression and multiclass classification. IEEE Trans. Syst. Man Cybern. Part B: Cybern. 42 (2), 513–529.
Huang, G.-B., Zhu, Q.-Y., Siew, C.-K., 2006. Extreme learning machine: theory and applications. Neurocomputing 70 (1–3), 489–501.
Jiang, Y., Wu, J., Zong, C., 2014. An effective diagnosis method for single and multiple defects detection in gearbox based on nonlinear feature selection and kernel-based extreme learning machine. J. Vibroeng. 16 (1), 499–512.
Jun, W., Shitong, W., Chung, F.-l., 2011. Positive and negative fuzzy rule system, extreme learning machine and image classification. Int. J. Mach. Learn. Cybern. 2 (4), 261–271.
Komaki, G.M., Kayvanfar, V., 2015. Grey Wolf Optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time. J. Comput. Sci. 8, 109–120.
Leshno, M., et al., 1993. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural Netw. 6 (6), 861–867.
Lin, S.-J., Chang, C., Hsu, M.-F., 2013. Multiple extreme learning machines for a two-class imbalance corporate life cycle prediction. Knowl. Based Syst. 39, 214–223.
Liu, B., et al., 2014. 2-D defect profile reconstruction from ultrasonic guided wave signals based on QGA-kernelized ELM. Neurocomputing 128, 217–223.
Ma, C., et al., 2016. A novel kernel extreme learning machine algorithm based on self-adaptive artificial bee colony optimisation strategy. Int. J. Syst. Sci. 47 (6), 1342–1357.
Min, J.H., Lee, Y.-C., 2005. Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. Expert Syst. Appl. 28 (4), 603–614.
Mirjalili, S., 2015. How effective is the Grey Wolf optimizer in training multi-layer perceptrons. Appl. Intell. 43 (1), 150–161.
Mirjalili, S., Mirjalili, S.M., Lewis, A., 2014. Grey wolf optimizer. Adv. Eng. Softw. 69, 46–61.
Mirjalili, S., Mirjalili, S.M., Hatamlou, A., 2015. Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput. Appl., 1–19.
Muangkote, N., Sunat, K., Chiewchanwattana, S., 2014. An improved grey wolf optimizer for training q-Gaussian radial basis functional-link nets. In: Proceedings of the 2014 International 2014 Conference on Computer Science and Engineering (ICSEC).
Naveen, N., et al., 2010. Differential evolution trained radial basis function network: application to bankruptcy prediction in banks. Int. J. BioInspir. Comput. 2 (3–4), 222–232.
Ohlson, J.A., 1980. Financial ratios and the probabilistic prediction of bankruptcy. J. Account. Res., 109–131.
Pal, M., Maxwell, A.E., Warner, T.A., 2013. Kernel-based extreme learning machine for remote-sensing image classification. Remote Sens. Lett. 4 (9), 853–862.
Pan, W.-T., 2012. A new fruit fly optimization algorithm: taking the financial distress model as an example. Knowl. Based Syst. 26, 69–74.
Paramjeet, Ravi, V., 2011. Bacterial foraging trained wavelet neural networks: application to bankruptcy prediction in banks. Int. J. Data Anal. Tech. Strateg. 3 (3), 261–280.
Pietruszkiewicz, W., 2008. Dynamical systems and nonlinear Kalman filtering applied in classification. In: Proceedings of the 7th IEEE International Conference on Cybernetic Intelligent Systems, CIS 2008.
Ravi, V., Pramodh, C., 2008. Threshold accepting trained principal component neural network and feature subset selection: application to bankruptcy prediction in banks. Appl. Soft Comput. 8 (4), 1539–1548.
Ravi Kumar, P., Ravi, V., 2007. Bankruptcy prediction in banks and firms via statistical and intelligent techniques – A review. Eur. J. Oper. Res. 180 (1), 1–28.
Ravisankar, P., Ravi, V., 2009. Failure prediction of banks using threshold accepting trained kernel principal component neural network. In: Proceedings of the IEEE World Congress on Nature & Biologically Inspired Computing, NaBIC 2009.
Reddy, K.N., Ravi, V., 2013. Differential evolution trained kernel principal component WNN and kernel binary quantile regression: application to banking. Knowl. Based Syst. 39, 45–56.
Salzberg, S.L., 1997. On comparing classifiers: pitfalls to avoid and a recommended approach. Data Min. Knowl. Discov. 1 (3), 317–328.
Sarkar, S., Sriram, R.S., 2001. Bayesian models for early warning of bank failures. Manag. Sci. 47 (11), 1457–1475.
Sharma, N., Arun, N., Ravi, V., 2013. An ant colony optimisation and Nelder-Mead simplex hybrid algorithm for training neural networks: an application to bankruptcy prediction in banks. Int. J. Inf. Decis. Sci. 5 (2), 188–203.
Shen, L., et al., 2016. Evolving support vector machines using fruit fly optimization for medical data classification. Knowl. Based Syst. 96, 61–75.
Shin, K.-S., Lee, T.S., Kim, H.-j., 2005. An application of support vector machines in bankruptcy prediction model. Expert Syst. Appl. 28 (1), 127–135.
Sulaiman, M.H., et al., 2015. Using the gray wolf optimizer for solving optimal reactive power dispatch problem. Appl. Soft Comput. 32, 286–292.
Sun, L., Shenoy, P.P., 2007. Using Bayesian networks for bankruptcy prediction: some methodological issues. Eur. J. Oper. Res. 180 (2), 738–753.
Tsai, C.-F., Wu, J.-W., 2008. Using neural network ensembles for bankruptcy prediction and credit scoring. Expert Syst. Appl. 34 (4), 2639–2649.
Vasu, M., Ravi, V., 2011. Bankruptcy prediction in banks by principal component analysis threshold accepting trained wavelet neural network hybrid. In: Proceedings of the International Conference on Data Mining. USA.
West, R.C., 1985. A factor-analytic approach to bank condition. J. Bank. Financ. 9 (2), 253–266.
Yu, Q., et al., 2014. Bankruptcy prediction using extreme learning machine and financial expertise. Neurocomputing 128, 296–302.
Zhang, G., et al., 1999. Artificial neural networks in bankruptcy prediction: general framework and cross-validation analysis. Eur. J. Oper. Res. 116 (1), 16–32.
Zhang, L., Yuan, J., 2015. Fault diagnosis of power transformers using kernel based extreme learning machine with particle swarm optimization. Appl. Math. Inf. Sci. 9 (2), 1003–1010.
Zhang, R., et al., 2007. Multicategory classification using an extreme learning machine for microarray gene expression cancer diagnosis. IEEE/ACM Trans. Comput. Biol. Bioinform. 4 (3), 485–495.
Zhao, D., et al., 2017. An effective computational model for bankruptcy prediction using kernel extreme learning machine approach. Comput. Econ. 49 (2), 325–341.
Zhong, H., et al., 2014. Comparing the learning effectiveness of BP, ELM, I-ELM, and SVM for corporate credit ratings. Neurocomputing 128, 285–295.