Appl Intell
DOI 10.1007/s10489-017-1027-8

Selective ensemble based on extreme learning machine and improved discrete artificial fish swarm algorithm for haze forecast

Xuhui Zhu[1,2,3] · Zhiwei Ni[1,2] · Meiying Cheng[1,2] · Feifei Jin[1,2] · Jingming Li[1,2] · Gary Weckman[3]

Abstract Urban haze pollution is becoming increasingly serious, which is considered very harmful for humans by World Health Organization (WHO). Haze forecasts can be used to protect human health. In this paper, a Selective ENsemble based on an Extreme Learning Machine (ELM) and Improved Discrete Artificial Fish swarm algorithm (IDAFSEN) is proposed, which overcomes the drawback that a single ELM is unstable in terms of its classification. First, the initial pool of base ELMs is generated by using bootstrap sampling, which is then pre-pruned by calculating the pair-wise diversity measure of each base ELM. Second, partial-based ELMs among the initial pool after pre-pruning with higher precision and with greater diversity are selected by using an Improved Discrete Artificial Fish Swarm Algorithm (IDAFSA). Finally, the selected base ELMs are integrated through majority voting. The Experimental results on 16 datasets from the UCI Machine Learning Repository demonstrate that IDAFSEN can achieve better classification accuracy than other previously reported methods. After a performance evaluation of the proposed approach, this paper looks at how this can be used in haze forecasting in China to protect human health.

Keywords Selective ensemble · Artificial fish swarm algorithm · Extreme learning machine · Haze forecast

1 Introduction

With the sustained rapid development of China, the urbanization process and the process of accelerating industrialization, which cause environmental pollution and ecology deterioration (air pollution in particular), is a growing concern. In several large cities in China such as Beijing, Shanghai and Guangzhou, urban air quality is deteriorating and visibility is decreasing in recent years with dramatic increases in energy consumption and pollutant emissions [1]. The main reason for poor air quality is gaseous and particulate emissions from various natural and anthropogenic sources. A huge amount of dust is being added to the ambient air from human activity, including vehicles, wood burning, power plants and industrial processes [2] which leads to high Particulate Matter (PM) concentrations [3]. Hazy weather may then occur under bad meteorological conditions [4]. Haze is a phenomenon [5] that can lead to atmospheric visibility that is less than 10 km, primarily owing to suspended particles, smoke, and vapor in the atmosphere, and may be caused when sunlight encounters tiny pollution particles in the air, which can reduce visibility, particularly during humid conditions. Hazy weather heavily threatens human respiratory health [6, 7] and people should be warned to avoid strenuous activity. It is estimated that more than three million deaths occur globally every year because of air pollution, mainly because of particulate matter [2]. Therefore, haze forecasting is badly needed in any urban area.

The classification prediction methods that were used early on for haze forecasting include Logistic Regression (LR) [8], Artificial Neural Networks (ANNs) [9] and

✉ Xuhui Zhu
zx572916@ohio.edu

[1] School of Management, Hefei University of Technology, Hefei 230009, China

[2] Key Laboratory of Process Optimization and Intelligent Decision-making, Ministry of Education, Hefei 230009, China

[3] Department of Industrial and Systems Engineering, Ohio University, Athens, OH 45701, USA

Published online: 07 September 2017

Support Vector Machine (SVM) [10]. Gordon Reikard [11] used LR to establish a time series model for the air pollution problem. Dhirendra Mishra et al. [2] also applied ANNs in a $PM_{2.5}$ forecast and achieved a good result. A novel hybrid model combining air mass trajectory analysis and wavelet transformation to improve an ANN's forecast accuracy of the daily average concentrations of $PM_{2.5}$ two days in advance was proposed by Xiao et al. [12]. Bai et al. [13] built an air pollutant forecast model by using wavelet techniques and Back Propagation Neural Networks (BPNN). García Nieto et al. [14] established an air pollution model by using the SVM technique in the Oviedo urban area (Northern Spain) at a local scale; Dumitrache et al. [15] also built a PM_{10} concentration prediction model for the Romanian territory, and attained good performance.

ELM was proposed by Huang et al. [16], which is an efficient learning algorithm based on a Single Layer Feedforward Network (SLFN). The ELM includes input, hidden, and output layer nodes [17]. Only hidden layer nodes need to be set in an ELM, and unique solutions can be obtained. It has some advantages, such as a faster learning rate and better generalization ability.

$$\sum_{i=1}^{L} \beta_i g_i(x_j) = \sum_{i=1}^{L} \beta_i g(\omega_i x_j + b_i) = y_i \tag{1}$$

For N arbitrary distinct samples (x_i, y_i), the model of ELM can be expressed as the following formula. Where $x_i = [x_{i1}, x_{i2}, \cdots, x_{in}]^T \in R^n$, $y_i = [y_{i1}, y_{i2}, \cdots, y_{im}]^T \in R^m$, $1 \le i, j \le N$, L indicates the number of hidden layer nodes, $g(x)$ demonstrates a hidden layer activation function, ω_i represents the weight vector connecting the input node and the ith hidden node, β_i stands for the weight vector connecting the ith hidden node and the output node, and b_i shows the threshold of the ith hidden node. The formula (8) can be rewritten as (2).

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

Where $H = \begin{bmatrix} g(\omega_1 x_1 + b_1) & \cdots & g(\omega_L x_1 + b_L) \\ \vdots & \ddots & \vdots \\ g(\omega_1 x_N + b_1) & \cdots & g(\omega_L x_N + b_L) \end{bmatrix}_{N \times L}$,

$Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix}_{N \times m}$, and $\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m}$, H is the hidden layer output matrix of the neural network. This is taken from the Moore-Penrose inverse theorem in a wider sense, and then

$$\beta = H^+ Y \tag{3}$$

Where H^+ can be obtained by using a singular value decomposition.

A single ELM is unstable in its classification prediction. Ensemble learning is primarily considered for improving predictive accuracy. Given an ensemble composed of M base classifiers, the number of nonempty subsets is $2^M - 1$, which has been proven to be an NP-complete problem [18]. To alleviate this computational burden, many ensemble pruning algorithms have been proposed in the literature; please refer to the related work in Section 2.1 for additional details.

However, the level of classification accuracy expected has not been reached. Pair-wise diversity measures have good performance when it comes to measuring diversity among base classifiers [19, 20], which can be used to pre-prune base classifiers [21, 22]; Artificial Fish Swarm Algorithm (AFSA) has a fast convergence speed and good global convergence, which can be regarded as a search strategy [23–28]. Based upon these considerations, this paper proposes a Selective ENsemble based upon ELM and an Improved Discrete Artificial Fish swarm algorithm (IDAFSEN). This mission is attained in three steps: First, basic ELMs are trained by using the bootstrap sampling. Secondly, the base classifiers are pre-pruned by calculating pair-wise diversity measures for each base classifier. Finally, partial base ELMs which perform better are selected by using IDAFSA, and integrated by majority voting.

2 Related work

2.1 Ensemble pruning

To achieve better accuracy in classification, bagging [29, 30] and boosting [31, 32] typical ensemble algorithms can be employed to construct ensemble system [33, 34]. Lan et al. [35] proposed the ensembling of online sequential ELMs, which perform better than the original ELM. Tian et al. [36] proposed a bagging ensemble method based on ELMs. Another ensemble method based on ELMs and a modified AdaBoost RT algorithm was proposed by Liu et al. [37]. Cao et al. [38] proposed an improved ensemble algorithm for classification by voting based on the ELM. Zhang et al. [39] proposed an new ELMs' ensemble selection algorithm based upon the Greedy Randomized Adaptive Search Procedure (GRASP), and is relatively efficient. Lu et al. [21] proposed a Disagreement measure-based, voting-based ELM (D-D-ELM), which prunes one base ELM with the largest disagreement measure. An et al. [22] proposed the Double-Fault measure based voting-based ELM (DF-D-ELM), which prunes some base ELMs by calculating an one-side confidence interval on a double-fault measure, and achieves good performance in terms of classification of gene expression data.

In addition, Martínez-Muñoz et al. [40–42] proposed ordering-based approaches, and found that the base classifiers are complementary. Experimental results on UCI repository datasets show that an ordered ensemble can attain better results than Bagging. Margineantu et al. [43] proposed Kappa pruning, which sorts the base classifiers in order from lowest to highest based on the Kappa measure of each base classifier. It then integrates some base classifiers with small Kappa measure. Guo et al. [44] proposed Margin-based Ordered AGgregation for ensemble pruning (MOAG), which selects those base classifiers with larger margin-based criterion to compose a pruned ensemble. Dai et al. [45] proposed a Reverse Reduce-Error (RRE) pruning algorithm incorporated with subtraction. Zhou et al. [46] proposed a Genetic Algorithm-based Selective ENsemble (GASEN), which trains several base classifiers at first, and then assigns random weights to those base classifiers and employs Genetic Algorithms (GA) to evolve the weights, so that they can represent to some extent the fitness of the base classifiers in constituting an ensemble system. Finally, the optimal subset of base classifiers can be selected to constitute an ensemble based upon minimizing the generalization ensemble error. Zhang et al. [47] proposed a selective ensemble algorithm based on an Improved Discrete Glowworm Swarm Optimization algorithm (IDGSOSEN), which has been applied in haze forecasting. Cavalcanti et al. [48] proposed a Pruning method that combines different pair-wise Diversity matrices (DivP) through a Genetic Algorithm (GA), which transforms the combined diversity matrix into one or more graphs and then applies a graph coloring method for generating candidate ensembles. Ykhlef et al. [49] proposed a novel ensemble Pruning algorithm by using non-monotone Simple Coalitional Games (SCG-P), which evaluates the diversity among the contributions of each classification based on the Banzhaf power index, and obtains the pruned ensemble with the minimal wining coalition made up of the base classifiers.

2.2 Contributions and outline

The contributions of the proposed research are described as follows:

(1) We propose a novel methodology for ensemble pruning based on ELM and IDAFSA.
(2) IDAFSA is proposed for searching in binary solution space.
(3) We find that the double-fault measure among base classifiers can be used for pre-pruning.
(4) We show the efficiency of the proposed methodology through experiments on 16 UCI datasets, which have been applied in haze forecasting in China.

This paper's basic structure is as follows. In Section 3, the basic concept of a pair-wise diversity measure is briefly reviewed. IDAFSA is proposed in Section 4. In Section 5, we introduce IDAFSEN, and how to use the proposed method. Experimental results and comparisons are expressed in Section 6. In Section 7, we discuss the performance of IDAFSEN. Experiments on 16 datasets from the UCI Machine Learning Repository demonstrate that IDAFSEN can achieve better classification accuracy than other selective ensemble approaches, and IDAFSEN can be used to accurately predict haze in China.

3 Pair-wise diversity measure

Diversity among the base classifiers in an ensemble system is of great importance. Intuitively speaking, the key to the success of an ensemble system is that the base classifiers are diverse [22]. In spite of the incomplete understanding of the concept of diversity, a lot of complete theories have been proposed to estimate the diversity among base classifiers. Many definitions have been used throughout the literature, but there is not a widely accepted definition of diversity among classifiers. We will present five pair-wise diversity measures as follows [50]. We take binary classification $\{+1, -1\}$ as an example.

The main notation used in this paper is summarized as follows:

N_t : the number of samples;
X : the sample dataset, $X = \{x_1, x_2, \cdots, x_{N_t}\}$;
Y : the labels of sample set X, $Y = \{y_1, y_2, \cdots, y_{N_t}\}$;
M : the number of base classifiers;
F : the initial pool of base classifiers, $F = \{f_1, f_2, \cdots, f_M\}$;
$f_i(x_k)$: classification result of the sample $x_k (k = 1, 2, \cdots, N_t)$ is classified by the base classifier $f_i (i = 1, 2, \cdots, M)$;

To calculate the diversity measure among the base classifiers, we present contingency table [48] between f_i and f_j in Table 1.

In Table 1, a represents the number of examples in X correctly classified by both f_i and f_j; b indicates the number of examples in X correctly classified by f_i and incorrectly classified by f_j; c stands for the number of examples in X incorrectly classified by f_i and correctly classified by f_j; d demonstrates the number of examples in X incorrectly

Table 1 Contingency table for two base classifiers

	$f_i(x_k) = y_k$	$f_i(x_k) \neq y_k$
$f_j(x_k) = y_k$	a	b
$f_j(x_k) \neq y_k$	c	d

classified by both classifiers. Finally, $a + b + c + d = N_t$.

The correlation coefficient measure (ρ) stems from statistics, which can be calculated by (4).

$$\rho_{ij} = \frac{ad - bc}{\sqrt{(a+b)(c+d)(a+c)(b+d)}} \tag{4}$$

The Q-statistic measure (Q) was proposed by Yule [50], which is similar to the correlation coefficient measure and can be modified in (5).

$$Q_{ij} = \frac{ad - bc}{ad + bc} \tag{5}$$

A Pair-wise Kappa measure (Kp) was widely used in statistics for computation amounts less than the Q-statistic measure, the latter of which was used to analyze the diversity among classifiers for the first time by Margineantu and Dietterich [50]. It is rewritten as (6).

$$Kp_{ij} = \frac{2(ad - bc)}{(a+b)(b+d) + (a+c)(c+d)} \tag{6}$$

A disagreement measure was proposed by Skalak in 1996, which can be calculated as (7).

$$Dis_{ij} = \frac{b + c}{N_t} \tag{7}$$

Giacinto and Roli proposed a double-fault measure in 2001, which is expressed as (8).

$$DF_{ij} = \frac{d}{N_t} \tag{8}$$

4 Improvement of AFSA

In order to solve discrete problems in discrete binary space, the proposed IDAFSA is presented, and the outline of IDAFSA is described as follows.

4.1 AFSA analysis

AFSA [23] is a novel random searching algorithm for the global optimum by simulating fish swarm behavior, which contains swarming, following, preying, and random behavior. In AFSA, Artificial Fishes (AFs) are randomly generated by random function [24].

Assume that there are N AFs. Let X_i be the position of the current AF, and $Y = f(X)$ is the fitness function. The distance between X_i and X_j is $d_{ij} = \|X_i - X_j\|$. $Visual$, $Step$ and δ stand for visual scope, moving step, and crowd factor, respectively.

Preying behavior Randomly selecting a position X_j inside the visual scope of X_i, Y_j indicates the fitness. If $Y_i < Y_j$, it moves a step to X_j. Otherwise, we randomly select a position again and judge whether it satisfies the forward condition.

Swarming behavior Let X_c be the center position and n_f be the number of companions inside the visual scope of X_i. If $Y_c/n_f > \delta Y_i$, it goes forward a step to X_c; otherwise, it executes preying behavior.

Following behavior Let $X_{\max}$ be the optimal AF inside the visual scope of X_i, if $Y_{\max}/n_f > \delta Y_i$, it goes forward a step to $X_{\max}$; otherwise, it executes the preying behavior.

Bulletin board We record the global optimum position X_{opt} of AFs.

4.2 Population initialization based on Good-Point Set (GPS)

How to initialize a population can be treated as an optimal design problem, and random initialization is widely used. However, random initialization may lead to mal-distribution of the initial population, which makes heuristic algorithms trapped in a local optimum and affects their convergence. To overcome this drawback, we attempt to use GPS. Considering the deviation of points generated by GPS is smaller than that caused by random selection, GPS can be applied to a high-dimensional approximate calculation [51].

Assuming that the size of the initial population is n, n good-points in s-dimensional space are generated in the following way.

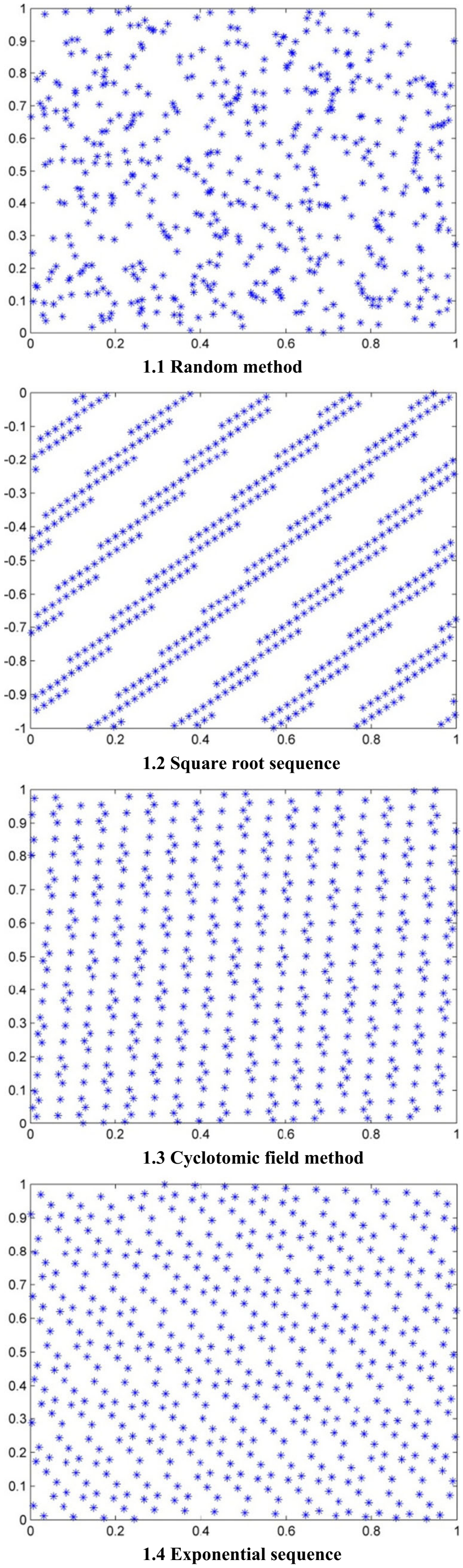
Let GPS be

$$P_n(i) = \{(\{r_1 \times i\}, \{r_2 \times i\}, \cdots, \{r_s \times i\}), i = 1, 2, \cdots, n\}.$$

Generally speaking, there are the following three ways in which this can be solved:

1) Square root sequence: $r_k = \sqrt{p_k}, 1 \leq k \leq s$, where p_k is prime number and if $i \neq j (1 \leq i, j \leq s)$, then $p_i \neq p_j$.
2) Cyclotomic field method: $r_k = 2\cos 2\pi k/p, 1 \leq k \leq s$, where p is the least prime which meets $(p-s)/2 \geq s$.
3) Exponential sequence: $r_k = \{e^k, 1 \leq k \leq s\}$.

By comparing the four methods in Fig. 1, we find that the exponential sequence performs best with more uniformity than that of the random method, square root sequence, and cyclotomic field method. Hence, AFs are generated by using an exponential sequence, which can keep diversity explicit and obtain a relatively good initial population.



1.1 Random method

1.2 Square root sequence

1.3 Cyclotomic field method

1.4 Exponential sequence

Fig. 1 Initial population distribution of 2-dimensional space with population size of 500

The pseudo-code of population initialization based on GPS is shown as follows.

Algorithm population initialization

```
Input: Population size $N$
Output: $N$ AFs
For each AF $i \in N$ {
    for each dimension of the $i$th AF $j \in n$ {
        $x_{ij} = \exp(j) * j - \lfloor \exp(j) * j \rfloor$
        If $(0.5 \le x_{ij} < 1)$, then $x_{ij} = 1$ else $x_{ij} = 0$
    }
}
Return $N$ AFs
```

4.3 Moving way of AF

To make a discrete AFSA simple yet efficient, we can update the position of AF in a simple way. In this work, we update AF's position at a certain probability. Let $X_i = (x_{i1}, x_{i2}, \cdots, x_{in})$ be the current AF; X_i is updated by changing each dimension of X_i at a certain probability, which can be formulated as:

$$x_{ik}(t+1) = \begin{cases} x_{ik}(t), if\ r(k) \le p_1 \\ x_{jk}(t), if\ p_1 < r(k) \le p_2 \\ round(rand), if\ r(k) > p_2 \end{cases} \tag{9}$$

Where r is n-dimensional vector $r = (r_1, r_2, \cdots, r_n)$, which is generated randomly, and $r_k \in (0, 1), k = 1, 2, \cdots, n$, p_1, p_2 are selected parameters, $rand$ is a randomly generated variable, and $round()$ is a function for a rounded integer.

4.4 Competitive operation

The basic AFSA has an obvious weakness: the local premature convergence at a later evolution stage, which makes it easily run into the local optimal solution because of a lack of diversity in the population. Thus, competitive operation was introduced in AFSA. New AFs can be generated by the crossover in AFs with a certain probability, and AFs who are continuous poor performers at searching will be eliminated, which can increase the diversity of the population. This competitive operation is described as (10) and (11).

If $rand \le r_1$, and then

$$\begin{cases} X_i = X_i', Y_i' > Y_i \\ X_j = X_j', Y_j' > Y_j \end{cases} \tag{10}$$

$$\begin{cases} X_i' = (1.0/2.0) \times ((1+r) \times X_i + (1-r) \times X_j) \\ X_j' = (1.0/2.0) \times ((1-r) \times X_i + (1+r) \times X_j) \end{cases} \tag{11}$$

Where $rand$, r_1 and r are random numbers in (0, 1), r_1 is called competitive probability, X_i and X_j are two different

AFs, X_i' and X_j' are new individuals who are generated by competitive operation.

4.5 Collaborative operation

To further improve the convergence rate and accuracy of AFSA, collaborative operation is introduced, which is inspired by co-evolution theory. In AFSA, the best AF $X_{\max}$ in the visual scope of X_i and the global optimum AF X_{opt} in the bulletin board are brought into the evolutionary process of AF X_i so that it can become a complementary and co-evolutionary effect. It can be expressed mathematically as follows.

If $rand \leq r_2$, and then

$$X_i = c_1 \times X_i + c_2 \times X_{\max} + c_3 \times X_{opt} \tag{12}$$

Where c_1, c_2, c_3 are collaborative factors, and $c_1+c_2+c_3 = 1$ with $0 \leq c_1, c_2, c_3 \leq 1$, r_2 has a collaborative probability, $r_2 \in (0, 1)$. Based on the above operation, AF's position can be updated as shown below.

$$x_{id} = \begin{cases} 1, \text{if } 0.5 \leq x_{ij} \leq 1 \\ 0, \text{if } 0 \leq x_{ij} < 0.5 \end{cases} \tag{13}$$

Where x_{ij} is the jth dimension of AF X_i.

5 IDAFSEN

5.1 Initial pool of ELMs

In this work, the initial pool of ELMs with size of M is generated by using bootstrap sampling in Bagging. Namely, we construct M independent training sample sets by adopting bootstrap sampling, and then training M sample sets with M base ELMs respectively. Thus, we can get an initial pool of base ELMs.

5.2 Pre-pruning

Ensemble pruning is a NP-complete problem, so it is difficult to find the exact solution for heuristic algorithms. To tackle the above problem, we attempt to pre-prune base classifiers before a selective ensemble based on IDAFSA, because pre-pruning some base ELMs who perform worse and with less diversity can reduce the number of base ELMs and greatly improve the efficiency of IDAFSA. Thus, we attempt to measure diversity among the base ELMs by calculating the pair-wise diversity measure of each base ELM. In five pair-wise diversity measures, which one should be taken as evaluative criteria for pre-pruning? We will discuss this question as follows.

The pair-wise diversity measure of each base ELM can be calculated using (14).

$$Div_i = \frac{1}{M} \sum_{j=1}^{M} div_{ij} \tag{14}$$

Where Div_i shows the pair-wise diversity measure of the ith base ELM, div_{ij} represents the pair-wise diversity measure between the ith base ELM and the jth base ELM, $1 \leq i \neq j \leq M$.

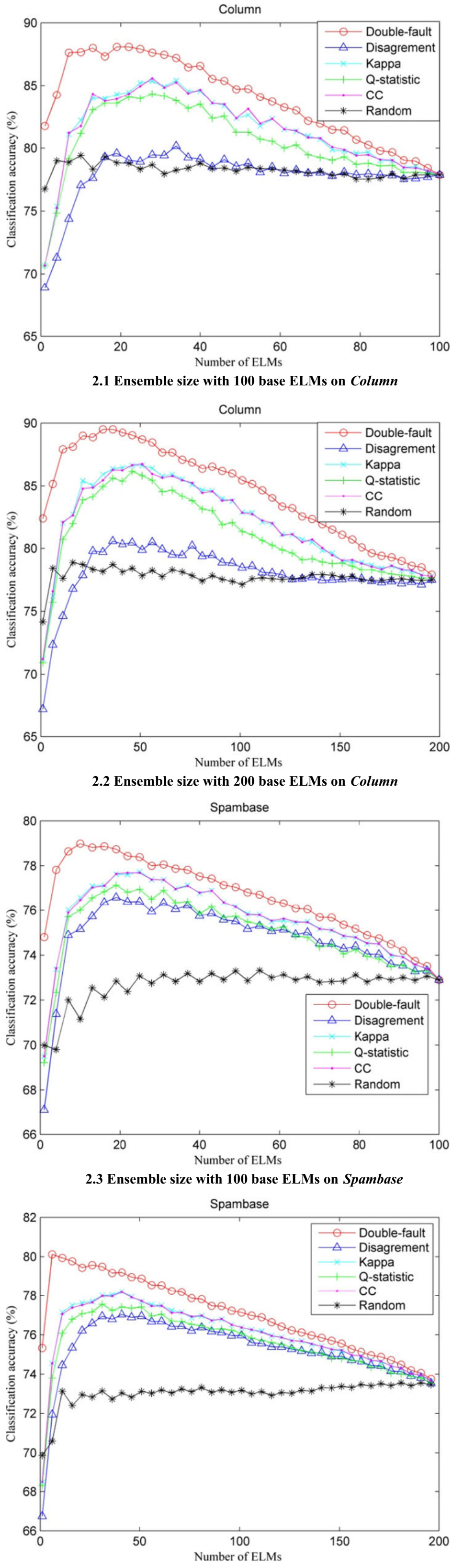
Theorem *Let $P*$ be the average classification accuracy of the base ELMs; then the pair-wise diversity measures in ensemble system are monotone decreasing with $P*$.*

Proof The correlation coefficient measure, Q-statistic measure, and pair-wise Kappa measure in an ensemble system are monotone decreasing with $P*$, which has been proved in literature [19, 20]. Disagreement measure and double-fault measure are monotone decreasing with $P*$, which has also been proved in literature [21, 22]. □

In conclusion, the five pair-wise diversity measures are monotone, decreasing with $P*$. If we remove some base ELMs with large diversity measures, the classification accuracy in the ensemble system can be improved.

To test the above theorem, the average classification for *Column* and *Spambase* datasets for random bagging and ordered bagging-according to pair-wise diversity measures (from lowest to highest) using initial pools with 100 and 200 base ELMs is shown in Fig. 2. From Fig. 2, it is easy to find that the ordered ensemble accuracy curves, according to diversity measures, reach a maximum at an intermediate number of base ELMs, and a double-fault measure can achieve better results than the other four diversity measures. After the maximum, the ensemble accuracy is monotone decreasing as the number of base ELMs increases. In other words, the ensemble accuracy can be improved after pruning base ELMs with larger diversity measures. Hence, a double-fault measure can be employed for pre-pruning base ELMs.

Let ξ_i be the double-fault measure DF_i of the ith base ELM; we can get the double-fault measure vector $\xi = [\xi_1, \xi_2, \cdots, \xi_M]$, and remove the ELMs with large ξ. We try to remove the base ELMs with larger double-fault measure by using mathematical statistics. We calculate the arithmetic average value of ξ, and prune the base ELMs whose double-fault measure $\xi > \bar{\xi}$, where $\bar{\xi} = \frac{1}{M} \sum_{i=1}^{M} \xi_i$. Namely, we remove the base ELMs where $\xi \notin \left(-\infty, \bar{\xi}\right)$, and the



2.1 Ensemble size with 100 base ELMs on *Column*

2.2 Ensemble size with 200 base ELMs on *Column*

2.3 Ensemble size with 100 base ELMs on *Spambase*

2.4 Ensemble size with 200 base ELMs on *Spambase*

Fig. 2 Average classification accuracy for *Column* and *Spambase* datasets for random bagging and ordered bagging according to pairwise diversity measures

remaining base ELMs are used for a selective ensemble based on IDAFSA.

5.3 Encoding method

Let $P = \{p_1, p_2, \cdots, p_{M'}\}$, which is the remaining base ELMs after pre-pruning, and M' is the number of the remaining base ELMs. The base ELM can be expressed as binary strings by adopting a binary encoding method, which is a combination of 0 and 1, and $X = (x_1, \cdots, x_i, \cdots, x_{M'})(i = 1, 2, \cdots, M')$. When $x_i = 1$, then it means that the ith base ELM is selected; when $x_i = 0$, it means that the ith base ELM is not selected. By this method, selection of ELMs can be converted into data which IDAFSA can deal with.

For example, assuming that $M' = 6$ and $X = (1, 0, 0, 0, 1, 1)$, it indicates p_1, p_5 and p_6 are selected; namely, the first, fifth and sixth base ELM are selected.

5.4 Fitness function

The fitness function of a selective ensemble problem is formulated as follows.

$$Fitness = A \tag{15}$$

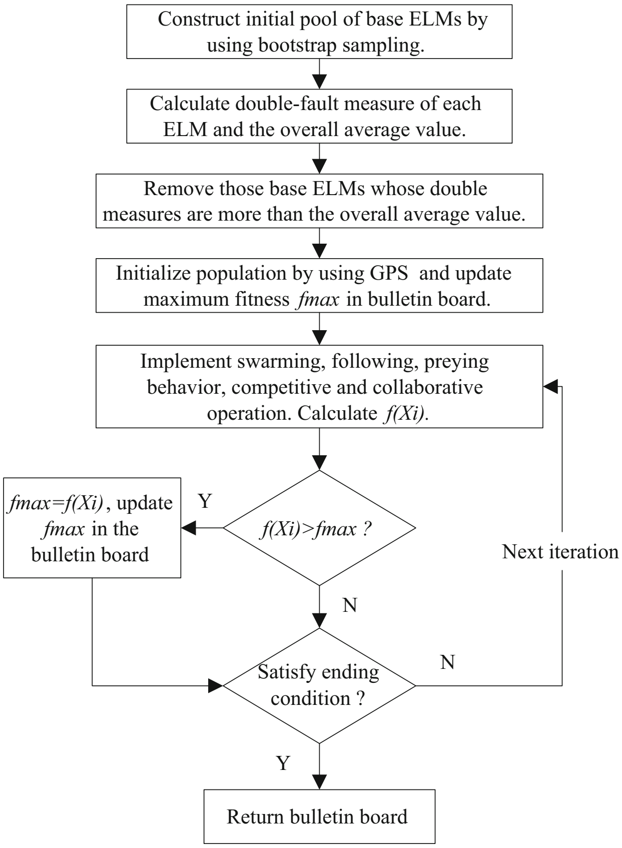

Construct initial pool of base ELMs by using bootstrap sampling.
Calculate double-fault measure of each ELM and the overall average value.
Remove those base ELMs whose double measures are more than the overall average value.
Initialize population by using GPS and update maximum fitness fmax in bulletin board.
Implement swarming, following, preying behavior, competitive and collaborative operation. Calculate f(Xi).
f(Xi)>fmax ?
Y
fmax=f(Xi), update fmax in the bulletin board
N
Satisfy ending condition ?
N
Next iteration
Y
Return bulletin board

Fig. 3 Flow chart of IDAFSEN

Where A is the classification accuracy between the predicted output and the expected output, $A = \frac{1}{m}\sum_{j=1}^{m} Acc(\hat{y}_j, y_j)$, where m is the number of testing samples, $\hat{y}_j$ is the predicted output on the jth testing samples, and y_j is the expected output on the jth testing samples. $Acc(\hat{y}_j, y_j) = \begin{cases} 1, \text{if } \hat{y}_j = y_j \\ 0, \text{if } \hat{y}_j \neq y_j \end{cases}$. The greater the fitness value is, the higher the ensemble accuracy is.

5.5 The algorithm of IDAFSEN

We present the pseudo-code of IDAFSEN as follows, which is a selective ensemble based on both IDAFSA and ELM. A chart showing the architecture of IDAFSEN is given in Fig. 3.

Algorithm IDAFSEN

```
Inputs: Ensemble system based on majority voting, Training set, testing set, and parameters of IDAFSA
Outputs: An optimal subset of base ELMs $X_{opt}$ and the ensemble results
(1) Construct initial pool of base ELMs with size $M$ by adopting bootstrap sampling
(2) Remove those base ELMs whose double-fault measures have an above-average value.
(3) Generate $N$ AFs by GPS and compute fitness $f$ using formula (14)
(4) $X_{opt} = maxfitness(X_1, X_2, \cdots, X_N)$, $f_{\max} = \max\{f_1, f_1, \cdots, f_N\}$
(5) Initialize parameters and iteration $t = 1$
(6) While($t \le t_{\max}$){
(7)   Do{
(8)     For each fish $i \in N$ {
(9)       $X_s = swarming(X_i)$
(10)      $X_f = following(X_i)$
(11)      $X_p = preying(X_i)$
(12)      $X_i = maxfitness(X_s, X_f, X_p)$
(13)   If $rand \le r_1$, then randomly select $j \in N$
       Update $X_i$ and $X_j$ by using (10) and (11)
(14) If $rand \le r_2$, then update $X_i$ by using (12) and (13)
(15)   If $(f(X_i) > f_{\max})$, then
         $X_{opt} = X_i$,   $f_{\max} = f(X_i)$
(16)   }
(17)   } Until each AF executes all behaviors
(18)   $t = t + 1$
(19) }
(20) Output $X_{opt}$ and ensemble results by majority
     voting
```

Algorithm swarming

```
Input: Ensemble system and a current position $X_i$
Output: A next position $X_s$
(1) $n_f = 0$, $X_c = 0$
(2) For each friend fish $X_j$ {
(3)   If $(distance(X_i, X_j) < visual)$, then
        $n_f = n_f + 1$,   $X_c = X_c + X_j$
(4)   }
(5)   $X_c = round(X_c / n_f)$
(6) If $(fitness(X_c)/n_f > \delta \times fitness(X_i))$, then $X_s = X_c$
        else $X_s = preying(X_i)$
(7)   Return the next position $X_s$
```

Algorithm following

```
Input: Ensemble system and a current position $X_i$
Output: A next position $X_f$
(1) $X_{\max} = X_i$
(2) For each friend fish $X_j$
(3) {
(4)     If $(fitness(X_{\max}) < fitness(X_j))$, then
          $X_{\max} = X_j$
(5) }
(6) $n_f = 0$
(7) For each friend fish $X_j$ {
(8)     If $(distance(X_{\max}, X_j) < visual)$, then
          $n_f = n_f + 1$
(9)     }
(10)    If $(fitness(X_{\max})/n_f > \delta \times fitness(X_i))$, then
        move to $X_{\max}$ by using (4) and get $X_f$
        Else $X_f = preying(X_i)$
(11)    Return the next position $X_f$
```

Algorithm preying

```
Input: Ensemble system and a current position $X_i$
Output: A next position $X_p$
(1) $j = 1$
(2) While $(j \le try\_number)$ {
(3)     Select a position $X_j$ randomly
(4)     If $(fitness(X_j) > fitness(X_i))$, then $X_p = X_j$ and break
(5)     $j = j + 1$
(6)   If $(j > try\_number)$, then select a position $X_j$ randomly and $X_p = X_j$ }
(7)   Return the next position $X_p$
```

Table 2 Description of UCI datasets used in this work

Datasets	Instances	Attributes	Classes
Hayes	160	5	3
Wine	178	13	3
Seeds	210	7	3
Heart	270	13	2
Column	310	6	2
Bupa	345	6	2
Ionosphere	351	34	2
ILP	583	10	2
Balance-scale	625	4	3
Breast-cancer	683	9	2
Diabetes	768	8	2
German	1000	20	2
QSAR	1055	41	2
CMC	1473	9	3
Abalone	4177	8	3
Spambase	4601	57	2

5.6 The complexity of IDAFSEN

By analyzing the computational complexity, we can understand the algorithm better. Assuming that there are N AFs, the size of the initial pool is M, the number of testing samples is m, and the maximum number of iterations is $t_{\max}$, A complexity analysis of the proposed algorithm is as follows.

(1) Pre-pruning process: the cost of calculating the double-fault measure of each base ELM is $O(M^2 \times m)$, and the cost of pre-pruning those base ELMs whose double-fault measures are less than the overall average value is $O(M)$.
(2) Initialization process: the cost of population initialization is $O(N)$.
(3) Iteration process: the cost of each iteration is $O(N^2)$, and the maximum iteration is $t_{\max}$; therefore, the total computational cost is $O(t_{\max} * N^2)$.

Considering all of the above, the entire computational cost of the proposed algorithm is

$$O(t_{\max} * N^2).$$

6 Experimental results

In this section, in order to show how IDAFSEN performs, we have carried out experiments on 16 classification problems from the UCI Machine Learning Repository, and then we have applied IDAFSEN to haze forecasting in China. The experiments were implemented in Matlab 2012a. The algorithm is tested on a computer running 32-bit Windows 7 with a 2.9 GHz processor and 2GB memory. To reduce

Table 3 Classification accuracy of the ensembles for different sizes of ELMs (50, 100, 150) on the test datasets

Datasets	50				100				150			
	IDAFSEN	Best	Mean	Worst	IDAFSEN	Best	Mean	Worst	IDAFSEN	Best	Mean	Worst
Hayes	79.56	73.85	53.48	40.73	79.89	74.79	53.28	39.06	78.96	76.67	53.17	38.23
Wine	80.92	81.20	60.25	45.92	82.99	81.03	60.49	43.48	83.05	81.90	60.49	43.62
Seeds	100	99.28	94.20	85.89	100	99.89	94.18	85.56	100	99.78	94.16	84.61
Heart	84.81	74.10	63.74	52.33	85.62	75.19	63.86	50.62	85.38	75.48	63.73	49.85
Column	91.67	84.17	75.41	64.83	92.71	85.96	75.55	63.96	92.21	86.29	75.56	62.92
Bupa	78.88	69.61	60.45	50.00	80.53	70.98	60.26	48.21	80.03	71.40	60.19	47.72
Ionosphere	94.28	92.38	87.88	82.57	94.71	92.77	87.88	82.15	94.49	93.00	87.87	81.52
ILP	74.03	74.09	71.76	65.59	74.41	74.46	71.74	64.74	73.76	74.46	71.71	63.81
Balance-scale	93.12	91.23	87.58	83.04	93.17	91.71	87.69	82.43	92.99	91.95	87.64	82.05
Breast-cancer	100	99.22	96.06	91.35	100	99.55	96.02	90.75	100	99.52	96.05	90.60
Diabetes	72.71	69.37	61.75	53.13	73.17	70.34	61.70	52.20	73.11	70.85	61.72	51.21
German	78.58	74.92	69.73	64.13	78.93	75.32	69.63	63.18	78.82	75.63	69.63	63.12
QSAR	87.93	82.26	74.53	66.58	88.15	82.92	74.51	65.89	87.74	83.16	74.49	65.59
CMC	64.04	57.80	54.23	47.97	64.90	60.35	54.12	45.84	65.21	60.84	54.20	45.75
Abalone	58.99	57.38	55.11	52.88	59.63	57.55	55.07	52.49	59.60	57.67	55.07	52.41
Spambase	80.75	76.71	70.00	63.97	82.22	77.46	70.14	63.28	82.16	77.83	70.12	63.00

Table 4 Classification accuracy of the ensembles for different sizes of ELMs (200, 250, 300) on the test datasets

Datasets	200				250				300			
	IDAFSEN	Best	Mean	Worst	IDAFSEN	Best	Mean	Worst	IDAFSEN	Best	Mean	Worst
Hayes	78.54	77.29	53.15	37.08	78.02	77.81	53.25	36.46	77.40	78.12	53.29	36.15
Wine	82.47	83.22	60.45	43.62	82.36	83.96	60.49	42.93	81.61	83.97	60.43	42.37
Seeds	100	100	94.15	83.89	100	100	94.16	83.56	100	100	94.14	83.17
Heart	83.76	76.00	63.69	49.67	83.14	76.05	63.68	48.76	83.04	76.29	63.67	48.29
Column	92.13	86.50	75.57	62.25	92.04	86.75	75.55	62.17	91.71	87.04	75.52	61.63
Bupa	79.75	71.75	60.23	47.16	79.26	71.96	60.22	46.67	79.09	72.14	60.22	46.28
Ionosphere	94.42	93.20	87.86	81.39	94.22	93.36	87.86	81.16	94.26	93.39	87.85	81.02
ILP	73.48	74.84	71.74	63.28	73.38	75.06	74.72	62.48	73.36	75.11	71.73	62.41
Balance-scale	92.74	91.97	87.66	81.79	92.53	92.16	87.66	81.63	92.61	92.18	87.68	81.55
Breast-cancer	100	99.65	96.04	90.45	100	99.72	96.04	90.20	100	99.80	96.04	90.03
Diabetes	73.27	71.25	61.76	51.03	72.92	71.51	61.76	50.81	72.76	71.85	61.74	50.56
German	78.68	76.00	69.65	62.88	78.48	76.15	69.64	62.65	78.32	76.22	69.63	62.58
QSAR	87.63	83.48	74.47	64.88	87.38	83.66	74.46	64.49	86.86	83.94	74.48	64.17
CMC	65.01	60.92	54.21	45.49	64.91	61.01	54.24	45.34	64.77	61.23	54.23	45.16
Abalone	59.42	57.75	55.08	52.36	59.60	57.80	55.08	52.28	59.44	57.89	55.07	52.16
Spambase	81.81	78.20	70.12	62.69	81.89	78.38	70.13	62.66	81.45	78.42	70.12	62.50

the random effects of experiments, the experiments were repeated 30 times independently for the same algorithm to achieve the mean results. The parameters of IDAFSA are set as follows: the visual is half the number of base ELMs after pre-pruning, the crowd factor $\delta = 0.9$, $p_1 = 0.15$, $p_2 = 0.85$ [47], the try number $try\_number = 5$, the maximum number of iterations $t_{\max} = 400$, the competitive probability $r_1 = 0.6$, the collaborative probability $r_2 = 0.4$ and the collaborative factors $c_1 = c_2 = c_3 = 1/3$. All of these parameters were defined empirically.

Table 5 Comparison with Bagging on different sizes of ELMs (50, 100, 150)

Datasets	50				100				150			
	Bagging	n	IDAFSEN	n	Bagging	n	IDAFSEN	n	Bagging	n	IDAFSEN	n
Hayes	61.25	50	79.56	9	61.64	100	79.89	18	62.89	150	78.96	31
Wine	66.26	50	80.92	9	66.83	100	82.99	19	67.18	150	83.05	26
Seeds	98.55	50	100	12	98.22	100	100	21	98.44	150	100	37
Heart	71.29	50	84.81	10	71.76	100	85.62	20	72.57	150	85.38	31
Column	77.67	50	91.67	10	77.33	100	92.71	15	77.83	150	92.21	32
Bupa	65.96	50	78.88	11	65.89	100	80.53	21	65.12	150	80.03	31
Ionosphere	90.66	50	94.28	10	90.26	100	94.71	19	90.30	150	94.49	34
ILP	72.82	50	74.03	6	72.93	100	74.41	15	72.93	150	73.76	24
Balance-scale	90.29	50	93.12	11	90.45	100	93.17	21	90.45	150	92.99	32
Breast-cancer	98.55	50	100	12	98.37	100	100	23	98.50	150	100	37
Diabetes	63.98	50	72.71	10	63.51	100	73.17	18	63.49	150	73.11	32
German	74.18	50	78.58	11	74.30	100	78.93	20	74.28	150	78.82	31
QSAR	79.70	50	87.93	10	80.28	100	88.15	17	80.22	150	87.74	32
CMC	57.78	50	64.04	10	57.99	100	64.90	21	58.38	150	65.21	31
Abalone	55.93	50	58.99	10	55.94	100	59.63	18	56.03	150	59.60	30
Spambase	72.66	50	80.75	8	72.98	100	82.22	16	73.31	150	82.16	25

n: the number after pruning

Table 6 Comparison with Bagging on different sizes of ELMs (200, 250, 300)

Datasets	200				250				300			
	Bagging	n	IDAFSEN	n	Bagging	n	IDAFSEN	n	Bagging	n	IDAFSEN	n
Hayes	61.16	200	78.54	45	61.06	250	78.02	55	61.17	300	77.40	70
Wine	67.24	200	82.47	37	67.30	250	82.36	47	67.53	300	81.61	58
Seeds	98.44	200	100	58	98.33	250	100	63	98.33	300	100	77
Heart	72.14	200	83.76	42	71.81	250	83.14	51	71.86	300	83.04	65
Column	77.50	200	92.13	45	77.63	250	92.04	54	77.58	300	91.71	67
Bupa	65.54	200	79.75	42	65.26	250	79.26	54	65.19	300	79.09	65
Ionosphere	94.42	200	94.42	47	90.17	250	94.22	60	90.13	300	94.26	73
ILP	72.93	200	73.48	35	72.93	250	73.38	44	72.93	300	73.36	51
Balance-scale	90.67	200	92.74	44	90.72	250	92.53	56	90.64	300	92.61	67
Breast-cancer	98.52	200	100	48	98.50	250	100	63	98.50	300	100	75
Diabetes	63.63	200	73.27	43	63.93	250	72.92	52	63.77	300	72.76	65
German	74.43	200	78.68	40	74.28	250	78.48	53	74.37	300	78.32	66
QSAR	79.96	200	87.63	42	80.15	250	87.38	55	80.15	300	86.86	69
CMC	58.44	200	65.01	43	58.60	250	64.91	54	58.38	300	64.77	64
Abalone	55.96	200	59.42	45	56.00	250	59.60	55	55.95	300	59.44	65
Spambase	73.36	200	81.81	35	73.39	250	81.89	46	73.43	300	81.45	57

n: the number after pruning

6.1 Experimental results on the 16 UCI datasets

The performance of IDAFSEN was evaluated on 16 datasets from the UCI machine Learning Repository, which are presented in Table 2.

Tables 3 and 4 indicate the results of IDAFSEN against the results extracted from the initial base ELMs. "Best", "Mean" and "Worst" represent the best, average and worst classification accuracies of the base ELMs in the pool on 16 different datasets. It is easy to discover that IDAFSEN

Table 7 Classification accuracy (%) and number of ELMs after pruning achieved by comparative algorithms

Datasets	IDAFSEN	n	Bagging	n	Kappa	n	AGOB	n	D-D-ELM	n	DF-D-ELM	n	GASEN	n
Hayes	79.89	18	61.64	100	66.41	20	67.34	28	68.75	99	71.15	55	65.73	44
Wine	82.99	19	66.83	100	71.47	20	73.01	30	67.41	99	77.53	56	73.16	47
Seeds	100	21	98.22	100	100	20	96.91	23	98.44	99	100	54	99.44	50
Heart	85.62	20	71.76	100	71.48	20	73.29	17	71.67	99	73.95	53	74.57	51
Column	92.71	15	77.33	100	85.21	20	81.81	14	77.83	99	84.50	54	80.92	49
Bupa	80.53	21	65.89	100	64.56	20	66.68	29	65.61	99	70.07	56	69.16	49
Ionosphere	94.71	19	90.26	100	93.27	20	92.06	18	90.13	99	92.64	56	91.78	47
ILP	74.41	15	72.93	100	72.73	20	72.84	19	72.93	99	72.96	53	72.95	47
Balance-scale	93.17	21	90.45	100	90.27	20	89.92	17	90.64	99	91.33	55	91.23	50
Breast-cancer	100	23	98.37	100	100	20	97.67	31	98.50	99	99.72	57	99.00	48
Diabetes	73.17	18	63.51	100	63.17	20	66.93	30	63.23	99	67.24	56	65.67	50
German	78.93	20	74.30	100	73.97	20	73.65	28	74.35	99	75.62	52	75.28	50
QSAR	88.15	17	80.28	100	82.09	20	82.03	16	80.02	99	82.26	55	81.66	49
CMC	64.90	21	57.99	100	56.07	20	56.99	29	58.34	99	59.35	55	59.84	52
Abalone	59.63	18	55.94	100	55.66	20	56.61	24	56.07	99	56.59	61	56.63	52
Spambase	82.22	16	72.98	100	77.62	20	78.36	23	73.38	99	77.06	54	74.72	49
Win/Tie/Loss	12/2/2		0/0/16		0/2/14		0/1/15		0/0/16		0/1/13		0/0/16	

n: Number of ELMs after pruning

Table 8 Classification accuracy (%) and number of ELMs after pruning achieved by comparative algorithms

Datasets	IDGSOSEN	n	MOAG	n	RRE	n	DivP	n	SCG-P	n
Hayes	72.08	45	67.29	23	76.04	12	77.22	3	78.67	29
Wine	82.01	44	74.54	26	85.58	17	81.69	2	86.35	23
Seeds	99.78	49	99.56	28	99.94	11	100	2	100	13
Heart	77.95	46	74.48	25	79.71	16	80.09	5	83.61	35
Column	84.83	46	82.79	20	87.17	14	90.37	7	91.29	27
Bupa	72.14	47	69.40	21	74.46	15	75.86	9	79.47	14
Ionosphere	92.81	47	92.11	26	93.63	11	95.15	4	95.15	19
ILP	72.93	50	72.96	23	72.93	11	74.26	3	73.41	18
Balance-scale	91.44	49	91.23	21	92.08	13	91.21	7	92.01	30
Breast-cancer	99.42	47	99.07	21	99.72	12	100	5	99.92	22
Diabetes	67.74	49	66.71	28	70.95	14	71.51	4	72.72	25
German	76.33	48	75.08	29	77.17	16	77.81	2	77.91	31
QSAR	83.33	48	81.83	22	85.31	12	85.45	6	80.43	34
CMC	61.20	47	59.55	25	62.55	15	62.67	9	62.61	35
Abalone	57.44	49	56.79	27	58.26	16	58.01	7	56.88	26
Spambase	76.78	45	75.38	24	78.81	13	80.57	5	80.92	31
Win/Tie/Loss	0/0/16		0/0/16		0/0/16		0/3/13		1/2/13	

n: Number of ELMs after pruning

can achieve a higher classification accuracy than the base ELMs.

Tables 5 and 6 show the results of IDAFSEN on 16 datasets, and compares the performance of IDAFSEN and the original ensemble (Bagging). IDAFSEN does better at pruning the initial base ELMs. It can also reduce more than 80% of the base ELMs in initial pool and attain a higher classification accuracy, because IDAFSA also performs very efficiently after pre-pruning some base ELMs. Meanwhile, the observation from Tables 5 and 6 implies that it may be better to ensemble many instead of all of the base ELMs at hand. We also find that IDAFSEN can achieve the maximum when the size of initial pool is set as 100.

To further assess the performance of IDAFSEN, we have compared it with the following algorithms: Bagging (Bootstrap aggregating) [29], Kappa (Kappa pruning) [43], AGOB (Aggregation Ordering in Bagging) [40], D-D-ELM [21], DF-D-ELM [22], GASEN [46], IDGSOSEN [47], MOAG [44], RRE [45], DivP [48], and SCG-P [49]. Bagging can increase the diversity of the ensemble system by selecting part of the samples, and each sample has an equal probability of being selected. Kappa pruning attempts to select the subset of the most diverse classifiers, but the final size of the ensemble is a parameter which needs to input by the user. AGOB and MOAG study the idea that the order in the base classifiers of being aggregated is important. D-D-ELM attempts to prune the ELM in an ensemble system with the largest disagreement measure. DF-D-ELM prunes the base ELMs by calculating the one-side confidence interval based on the double-fault measure of the base ELMs. GASEN employs a Genetic Algorithm (GA) to optimize the weights which are randomly assigned to the base classifiers; the best subset of base classifiers will be selected to constitute an ensemble based upon minimizing the generalization ensemble error. IDGSOSEN attempts to employ IDGSO to prune base classifiers. RRE can make full use of base classifiers which are not selected into the pruned ensemble. DivP combines different pair-wise diversity matrices to compose the sub-ensemble. SCG-P evaluates the diversity contribution of each classification based on the Banzhaf power index, and the pruned ensemble with the minimal winning coalition created from the base classifiers

Table 9 Results of the Wilcoxon signed-rank test

Comparison	p-value	Significant or not
IDAFSEN vs Bagging	0.00044	Yes
IDAFSEN vs Kappa	0.00012	Yes
IDAFSEN vs AGOB	0.00044	Yes
IDAFSEN vs D-D-ELM	0.00044	Yes
IDAFSEN vs DF-D-ELM	0.00006	Yes
IDAFSEN vs GASEN	0.00044	Yes
IDAFSEN vs IDGSOSEN	0.00044	Yes
IDAFSEN vs MOAG	0.00044	Yes
IDAFSEN vs RRE	0.00270	Yes
IDAFSEN vs DivP	0.00037	Yes
IDAFSEN vs SCG-P	0.01030	Yes

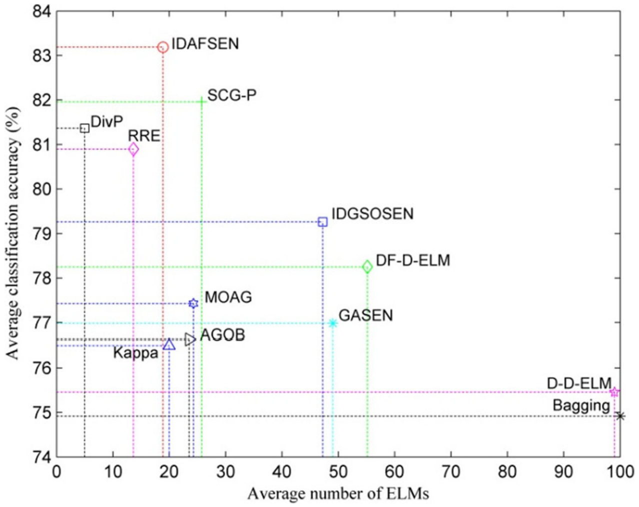


Fig. 4 Relationship between ensemble size and average performance of different approaches

can be achieved. These selective ensemble algorithms are implemented as described in their respective papers.

The classification accuracy achieved by all approaches using an initial pool with 100 base ELMs is reported in Tables 7 and 8. “Win”/“Tie”/“Loss” represents the number of times in which IDAFSEN scores better/neutral/inferior than Bagging, and the number of “Win” is much greater than “Tie” and “Loss”. From Tables 7 and 8, we can see that IDAFSEN performs better than Bagging, Kappa, AGOB, D-D-ELM, DF-D-ELM, GASEN, IDGSOSEN, MOAG, and SCG-P with a smaller number of the base ELMs. The number of the base ELMs after pruning is more than RRE and DivP, but IDAFSEN can achieve a higher classification accuracy than RRE and DivP on most validation datasets. To test the significance of the difference between the proposed approach and other approaches, we took the Wilcoxon signed-rank test at a significance level of 0.05. When the p-value is less than 0.05, the difference between the two approaches is significant. The p-values produced in the test are shown in Table 9, which expresses that the proposed approach attains better results with statistically significant differences in all eleven cases. Additionally, Fig. 4 demonstrates the trade-off between the average number of ELMs and the average classification accuracy in a more intuitive way, and IDAFSEN performs better than other selective ensemble approaches.

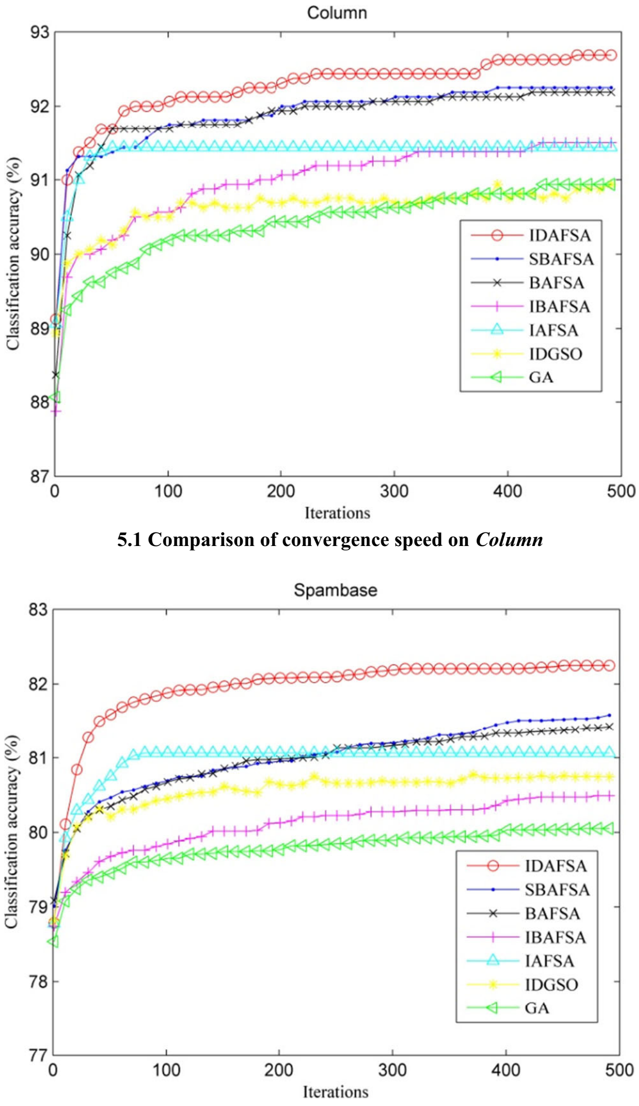
The comparison of executive time (the average value) running by all comparative algorithms is listed in Table 10. IDAFSEN uses more time than Bagging, Kappa, AGOB, D-D-ELM, DF-D-ELM, MOAG, RRE, and SCG-P, but less time than GASEN, IDGSOSEN, and DivP. For IDAFSEN, after pre-pruning for the initial pool based on ELM, it searches the optimal sub-ensemble by using IDAFSA, which has to call the ensemble system based on majority voting iteratively over each candidate ensemble or sub-ensemble. In IDAFSA, each AF represents a candidate ensemble and needs to move at least once per iteration, which needs to call an ensemble system based on majority voting once per movement for each AF, while IDAFSA searches the optimal sub-ensemble with a maximum iterations number of 400 and a population size of 25. The above

Table 10 Comparison of executive time with other selective algorithms

Datasets	Executive time (s)											
	IDAFSEN	Bagging	Kappa	AGOB	D-D-ELM	DF-D-ELM	GASEN	IDGSOSEN	MOAG	RRE	DivP	SCG-P
Hayes	15.38	0.35	0.34	1.28	0.39	0.38	17.10	18.46	0.37	1.32	21.88	1.56
Wine	22.74	0.38	0.39	1.96	0.41	0.41	27.53	30.66	0.45	2.01	34.29	0.98
Seeds	24.20	0.31	0.31	1.85	0.33	0.35	25.47	28.05	0.37	1.88	34.48	1.26
Heart	13.53	0.43	0.61	1.85	0.46	0.46	13.13	22.09	0.49	1.92	20.83	0.91
Column	14.95	0.53	0.44	2.19	0.50	0.55	14.52	22.01	0.51	2.15	23.84	1.43
Bupa	17.96	0.54	0.43	2.53	0.55	0.49	28.72	26.22	0.48	2.46	28.33	0.77
Ionosphere	18.31	0.66	0.56	2.62	0.60	0.59	17.00	25.47	0.55	2.69	27.93	1.80
ILP	20.98	0.61	0.63	3.28	0.65	0.65	35.57	32.97	0.63	3.35	34.36	1.84
Balance-scale	46.73	0.93	1.29	4.28	1.05	1.06	35.44	57.01	1.07	4.33	65.74	1.14
Breast-cancer	22.47	1.05	1.01	3.70	1.11	1.12	36.40	32.42	1.07	3.82	36.16	1.76
Diabetes	28.18	0.75	0.75	4.23	0.80	0.79	45.99	42.52	0.79	4.41	45.05	1.78
German	33.25	1.55	1.35	5.54	1.54	1.49	54.43	49.04	1.48	5.43	53.87	1.93
QSAR	28.08	2.05	2.27	5.11	1.93	2.09	27.05	38.88	2.05	5.58	44.25	2.51
CMC	101.14	1.88	1.73	9.27	2.01	2.07	79.40	124.09	2.02	9.74	149.86	2.62
Abalone	241.38	5.37	5.04	23.66	5.41	5.59	187.84	293.41	5.83	25.44	352.44	6.90
Spambase	95.28	7.07	7.09	19.71	7.08	7.03	100.38	148.16	6.99	19.40	158.09	7.23

process of iteratively searching the optimal sub-ensemble is relatively time-consuming. GASEN, IDGSOSEN and DivP search the optimal sub-ensemble without pre-pruning, and thus they cost more time than the proposed IDAFSEN. For the remaining selective ensemble approaches, all the base ELMs are simply packaged into the final ensemble without the iterative selection and evaluation, which costs less time, but cannot achieve as good of a performance in classification. Although IDAFSEN costs considerably more time, it can achieve notably better results than other selective ensemble algorithms. It is worth spending more time on significantly improving the ensemble accuracy.

In conclusion, IDAFSEN is superior to other selective ensemble approaches on most validation datasets, which reveals that the proposed algorithm is quite capable in terms of classification. Thus, the proposed approach can be applied in haze forecasting in the following Section 6.3.



5.1 Comparison of convergence speed on *Column*

5.2 Comparison of convergence speed on *Spambase*

Fig. 5 Relationship between performance of heuristic algorithms and iterations on *Column* and *Spambase* datasets

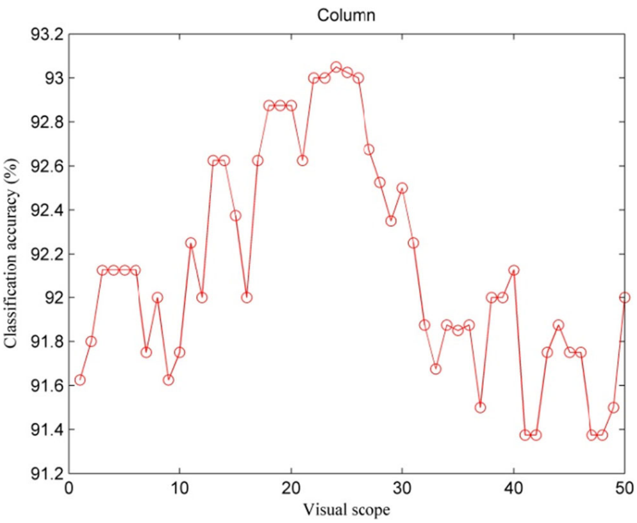


Fig. 6 Relationship between performance of IDAFSA and the parameter *Visual scope* on *Column*

6.2 Comparison with other heuristic algorithms

In IDAFSEN, we have employed IDAFSA to optimize the remaining base ELMs after pre-pruning. In order to evaluate the performance of the proposed IDAFSA, we compared it with the following algorithms: SBAFSA (Simplified Binary Artificial Fish Swarm Algorithm) [25], BAFSA (Binary Artificial Fish Swarm Algorithm) [26], IBAFSA (Improved Binary Artificial Fish Swarm Algorithm) [27], IAFSA (Improved Artificial Fish Swarm Algorithm) [28], IDGSO [47], and GA [46]. All of the aforementioned heuristic algorithms are binary searching algorithms. The following

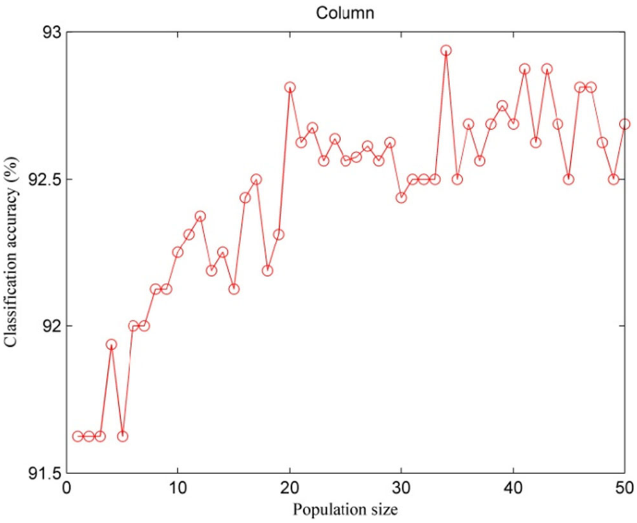


Fig. 7 Relationship between performance of IDAFSA and the parameter *population size* on *Column*

Table 11 Classification accuracy of the ensembles for different pool sizes (50, 100, 150) on haze datasets

Datasets	50				100				150			
	IDAFSEN	Best	Mean	Worst	IDAFSEN	Best	Mean	Worst	IDAFSEN	Best	Mean	Worst
Beijing	80.12	72.88	65.22	54.49	81.17	73.82	65.08	51.83	80.84	74.09	65.00	51.09
Shanghai	86.77	83.91	79.56	73.85	87.24	84.47	79.62	72.70	86.87	84.55	79.60	72.14
Guangzhou	87.51	83.83	79.16	72.92	87.86	84.55	79.27	70.84	87.41	84.61	79.28	70.72

experiments using an initial pool with 100 base ELMs were implemented on UCI datasets (*Column* and *Spambase*), which were shown in Fig. 5. After pre-pruning, the above seven heuristic algorithms were employed to find the optimal sub-ensemble. The population size of all the algorithms was half the number of base ELMs after pre-pruning.

From Fig. 5, we can see IDAFSA has a faster convergence speed than the other six binary heuristic searching algorithms, and the performance of IDAFSA increases at the beginning, then levels off or levels out. The performance of IDAFSA cannot be significantly improved after the 400th generation. Roughly speaking, the suitable parameter in terms of the number of iterations is thus 400. In Fig. 6, we can see that when the *Visual scope* reaches about 25, IDAFSA performs at its best. The *Visual scope* varies from 1 to 50 (the number of the remaining base ELMs after pre-pruning the initial pool with 100 base ELMs is about 50). Thus, the suitable *Visual scope* is half the number of base ELMs after pre-pruning. In Fig. 7, it is easy to see that the performance of IDAFSA can be improved with the increase of population size at first, and then it levels off. We thus advise that the suitable population size is half the number of base ELMs after pre-pruning (the population size of IDAFSA should be a monotonous increase alongside the size of the base ELMs after pre-pruning).

6.3 Haze forecasting in China based on IDAFSEN

In this subsection, the experiment is implemented on three real haze datasets (Beijing, Shanghai and Guangzhou) from the China National Environmental Monitoring Centre and the China Meteorological Administration between January 1st, 2015 and December 31th, 2016. There are 12 indicators in total, which are listed as follows: $PM_{2.5}$, PM_{10}, CO, NO_2, SO_2, O_3, relative humidity, maximum temperature, minimum temperature, rain capacity, wind direction and level of wind. The 12 indicators from the first day are used to predict whether or not there is haze in the second day. Namely, these 12 predictive factors from the first day are treated as condition features, and whether or not there is haze in the second day is taken as a decision feature.

Tables 11 and 12 show the prediction results of IDAFSEN vs the best, average, and worst results of the whole initial base ELMs on the haze datasets in China. We found that IDAFSEN can achieve a better result on haze forecasting than the average ELM. We also found that there are great differences in classification accuracies between Beijing and Shanghai. The haze in Beijing is more complicated to predict than in Shanghai, which leads to the great difference in classification accuracy [52]. Tables 13 and 14 reveal the results of IDAFSEN vs Bagging on the haze datasets. It is easy to find that IDAFSEN can achieve a higher accuracy with a fewer number of base ELMs than Bagging when it comes to haze forecasting in China.

The prediction results on haze datasets by using an initial pool with 100 base ELMs in Tables 11 and 12, the prediction results on haze datasets achieved by all approaches using an initial pool with 100 base ELMs are reported in Tables 15 and 16. From Tables 15 and 16, we can see that IDAFSEN can attain the best prediction results among all the approaches on haze forecasting in China; the executive time of all selective ensemble approaches is shown in Table 17.

In summary, IDAFSEN performs better than other selective ensemble approaches, and pre-pruning can greatly enhance the performance of ensemble pruning. We also find that IDAFSEN can be applied in haze forecasting in China in order to help protect human health. IDAFSEN provides a new approach for haze forecasting in China.

Table 12 Classification accuracy of the ensembles for different pool sizes (200, 250, 300) on haze datasets

Datasets	200				250				300			
	IDAFSEN	Best	Mean	Worst	IDAFSEN	Best	Mean	Worst	IDAFSEN	Best	Mean	Worst
Beijing	80.45	74.34	65.03	50.54	80.02	74.53	65.03	50.08	79.65	74.67	65.03	49.96
Shanghai	86.54	84.86	79.59	71.36	86.36	84.96	79.58	71.17	86.09	85.08	79.59	70.60
Guangzhou	87.08	84.77	79.22	70.58	86.87	84.90	79.24	69.81	86.71	85.00	79.25	69.65

Table 13 Comparison with Bagging on different sizes of ELMs (50, 100, 150)

Datasets	50				100				150			
	Bagging	n	IDAFSEN	n	Bagging	n	IDAFSEN	n	Bagging	n	IDAFSEN	n
Beijing	72.63	50	80.12	12	72.88	100	81.17	20	72.92	150	80.84	32
Shanghai	79.12	50	86.77	9	78.74	100	87.24	18	78.91	150	86.87	29
Guangzhou	81.67	50	87.51	9	81.91	100	87.86	19	81.95	150	87.41	30

n: the number after pruning

Table 14 Comparison with Bagging on different sizes of ELMs (200, 250, 300)

Datasets	200				250				300			
	Bagging	n	IDAFSEN	n	Bagging	n	IDAFSEN	n	Bagging	n	IDAFSEN	n
Beijing	73.23	150	80.45	43	73.17	150	80.02	54	73.19	300	79.65	66
Shanghai	78.66	150	86.54	39	78.54	150	86.36	52	78.54	300	86.09	63
Guangzhou	81.08	150	87.08	43	82.18	150	86.87	54	82.10	300	86.71	67

n: the number after pruning

Table 15 Classification accuracy (%) and number of ELMs after pruning achieved by comparative algorithms

Datasets	IDAFSEN	n	Bagging	n	Kappa	n	AGOB	n	D-D-ELM	n	DF-D-ELM	n	GASEN	n
Beijing	81.17	16	72.88	100	69.88	20	70.85	26	72.74	99	74.69	54	74.38	50
Shanghai	87.24	18	78.74	100	80.58	20	80.49	28	78.79	99	82.26	51	80.57	44
Guangzhou	87.86	19	81.91	100	82.63	20	80.74	21	82.20	99	82.96	53	83.00	49

n: Number of ELMs after pruning

Table 16 Classification accuracy (%) and number of ELMs after pruning achieved by comparative algorithms

Datasets	IDGSOSEN	n	MOAG	n	RRE	n	DivP	n	SCG-P	n
Beijing	76.44	48	74.17	18	77.53	25	74.48	4	79.84	32
Shanghai	82.10	40	81.75	24	84.16	22	84.51	5	85.42	25
Guangzhou	84.26	47	83.44	25	85.23	23	84.9	3	86.20	29

n: Number of ELMs after pruning

Table 17 Comparison of executive time with other selective algorithms

Datasets	Executive time (s)											
	IDAFSEN	Bagging	Kappa	AGOB	D-D-ELM	DF-D-ELM	GASEN	IDGSOSEN	MOAG	RRE	DivP	SCG-P
Hayes	27.78	0.79	0.79	4.17	0.84	0.85	45.69	51.48	0.83	4.19	46.93	1.41
Wine	27.26	0.86	0.87	4.18	0.93	0.89	43.96	44.68	0.93	4.26	45.33	1.97
Seeds	26.77	0.93	0.84	4.21	0.94	1.02	46.58	50.72	1.10	4.37	43.61	1.83

7 Conclusion

The traditional pattern recognition systems usually use a single classifier to classify haze samples, but a single classifier can lead to less robust models. For solving such problems in practical applications, Multiple Classifier Systems (MCSs) should be primarily considered for haze forecasting; however, there exists a large number of redundant base classifiers in MCSs. When finding the optimal sub-ensemble efficiently, a double-fault measure performs well in terms of measuring diversity among base classifiers, which can be used for pre-pruning. It is here that AFSA is used as a powerful searching algorithm. IDAFSA is proposed by taking the population initialization based on GPS, competitive operation, and collaborative operation. IDAFSEN is proposed based on IDAFSA and ELM. The experimental results on 16 UCI datasets demonstrate that IDAFSEN can achieve higher accuracy with a fewer number of base ELMs, and IDAFSA outperforms other binary heuristic searching algorithms. By comparing the proposed approach against Bagging and other selective ensemble approaches available in literature, we show its good classification performance and prediction ability on 16 different UCI datasets. Hence, it is shown that IDAFSEN can be used for haze forecasting in China. Furthermore we also assume that the ensemble many could be better than all in ensemble system [46].

In future work, we will attempt to use other diversity measures and criterion to pre-prune base classifiers, and then find the optimal sub-ensemble by using heuristic searching algorithms. We believe that these combinations of base classifiers should generate promising results, which can be applied in haze forecasting.

Acknowledgement This work was supported by National Nature Science Foundation of China under Grant No. 91546108, No. 71271071, No. 71490725, and No. 71301041, the National Key Research and Development Plan under Grant No. 2016YFF0202604 and the state scholarship fund. The authors would like to thank the reviewers for their comments and suggestions.

References

1. Gao LN, Jia GS, Zhang RJ et al (2011) Visibility trends in the Yangtze River Delta of China during 1981–2005. J Air Waste Manage 61:843–849
2. Mishra D, Goyal P, Upadhyay A (2015) Artificial intelligence based approach to forecast PM2.5 during haze episodes: a case study of Delhi, India. Atmos Environ 102:239–248
3. Deng J, Wang T, Jiang Z et al (2011) Characterization of visibility and its affecting factors over Nanjing, China. Atmos Res 101(3):681–691
4. Wang T, Jiang F, Deng J et al (2012) Urban air quality and regional haze weather forecast for Yangtze River Delta region. Atmos Environ 58:70–83
5. Zhang F, Chen J, Qiu T et al (2013) Pollution characteristics of $PM_{2.5}$ during a typical haze episode in Xiamen, China. Atmos Clim Sci 3(4):427–439
6. McLaren J, Williams ID (2015) The impact of communicating information about air pollution events on public health. Sci Total Environ 538:478–491
7. Li L, Lin GZ, Liu HZ et al (2015) Can the Air Pollution Index be used to communicate the health risks of air pollution? Environ Pollut 205:153–160
8. Sohn SY, Kim DH, Yoon JH (2016) Technology credit scoring model with fuzzy logistic regression. Appl Soft Comput 43:150–158
9. Chiteka K, Enweremadu CC (2016) Prediction of global horizontal solar irradiance in Zimbabwe using artificial neural networks. J Clean Prod 135:701–711
10. Malvoni M, De Giorgi MG, Congedo PM (2016) Data on support vector machines model to forecast photovoltaic power. Data Brief 9:13–16
11. Reikard G (2012) Forecasting volcanic air pollution in Hawaii: tests of time series models. Atmos Environ 60:593–600
12. Feng X, Li Q, Zhu Y et al (2015) Artificial neural networks forecasting of $PM_{2.5}$ pollution using air mass trajectory based geographic model and wavelet transformation. Atmos Environ 107:118–128
13. Bai Y, Li Y, Wang X et al (2016) Air pollutants concentrations forecasting using back propagation neural network based on wavelet decomposition with meteorological conditions. Atmos Pollut Res 7(3):557–566
14. García Nieto PJ, Combarro EF, del Coz Díaz JJ et al (2013) A SVM-based regression model to study the air quality at local scale in Oviedo urban area (Northern Spain): a case study. Appl Math Comput 219(17):8923–8937
15. Dumitrache RC, Iriza A, Maco BA et al (2016) Study on the influence of ground and satellite observations on the numerical air-quality for PM10 over Romanian territory. Atmos Environ 143:278–289
16. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70:489–501
17. Huang GB, Zhu QY, Siew CK (2006) Universal appximation using incremental constructive feed-forward network with random hidden nodes. IEEE Trans Neural Netw 17:879–892
18. Partalas I, Tsoumakas G, Vlahavas I (2009) Pruning an ensemble of classifiers via reinforcement learning. Neurocomputing 72:1900–1909
19. Yang C, Yin X, Hao H et al (2014) Classifier ensemble with diversity: effectiveness analysis and ensemble optimization. Acta Automat Sin 40(4):660–674
20. Kuncheva LI, Whitaker CJ (2003) Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Mach Learn 51(2):181–207
21. Lu H, An C, Ma X et al (2013) Disagreement measure based ensemble of extreme learning machine for gene expression data classification. Chin J Comput 36(2):341–348
22. Lu H, An C, Zheng E, Lu Y (2014) Dissimilarity based ensemble of extreme learning machine for gene expression data classification. Neurocomputing 128:22–30
23. Li X, Shao Z, Qian J (2002) An optimizing method based on autonomous animates: fish swarm algorithm. Syst Eng Theory Pract 22(11):32–38
24. Zhu X, Ni Z, Cheng M (2015) Self-adaptive improved artificial fish swarm algorithm with changing step. Comput Sci 42(2):210–216+246
25. Azad MAK, Rocha AMAC, Fernandes EMGP (2014) A simplified binary artificial fish swarm algorithm for 01 quadratic knapsack problems. J Comput Appl Math 259:897–904

26. Chen Y, Zhu Q, Xu H (2015) Finding rough set reducts with fish swarm algorithm. Knowl-Based Syst 81:22–29
27. Azad MAK, Rocha AMAC, Fernandes EMGP (2014) Improved binary artificial fish swarm algorithm for the 01 multidimensional knapsack problems. Swarm Evol Comput 14:66–75
28. Luan XY, Li ZP, Liu TZ (2016) A novel attribute reduction algorithm based on rough set and improved artificial fish swarm algorithm. Neurocomputing 174:522–529
29. Breiman L (1996) Bagging predictors. Mach Learn 24:123–140
30. Mordelet F, Vert JP (2014) A bagging SVM to learn from positive and unlabeled examples. Pattern Recogn Lett 37:201–209
31. Zou PC, Wang JD, Chen SC et al (2014) Bagging-like metric learning for support vector regression. Knowl-Based Syst 65:21–30
32. Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 55:119–139
33. Shigei N, Miyajima H, Maeda M et al (2009) Bagging and AdaBoost algorithms for vector quantization. Neurocomputing 73(1):106–114
34. García-Pedrajas N, Haro-García A (2014) Boosting instance selection algorithms. Knowl-Based Syst 67:342–360
35. Suresh S, Venkatesh Babu R, Kim HJ (2009) No-reference image quality assessment using modified extreme learning machine classifier. Appl Soft Comput 9(2):541–552
36. Tian H, Meng B (2010) A new modeling method based on bagging ELM for day-ahead electricity price prediction. Bio-Inspired Comput: Theor Appl (BIC-TA) 1076–1079
37. Tian HX, Mao ZZ (2010) An ensemble ELM based on modified AdaBoost. RT algorithm for predicting the temperature of molten steel in ladle furnace. IEEE Trans Autom Sci Eng 7(1):73–80
38. Cao W, Lin ZP, Huang GB, Liu N (2012) Voting based extreme learning machine. Inf Sci 185(1):66–77
39. Zhang T, Dai Q, Ma Z (2015) Extreme learning machines' ensemble selection with GRASP. Appl Intell 43(2):439–459
40. Martínez-Muñoz G, Suárez A (2004) Aggregation ordering in bagging. In: Proceedings of the IASTED international conference on artificial intelligence and applications, pp 258–263
41. Martínez-Muñoz G, Suárez A (2006) Pruning in ordered bagging ensembles. In: Proceedings of the twenty-third international conference on machine learning, pp 609–616
42. Martínez-Muñoz G, Hernández-Lobato D, Suárez A (2009) An analysis of ensemble pruning techniques based on ordered aggregation. IEEE Trans Pattern Anal Mach Intell 31(2):245–259
43. Margineantu DD, Dietterich TG (1997) Pruning adaptive boosting. In: Proceedings of the fourteenth international conference on machine learning, vol 97, pp 211–218
44. Guo L, Boukir S (2013) Margin-based ordered aggregation for ensemble pruning. Pattern Recogn Lett 34:603–609
45. Dai Q, Zhang T, Liu N (2015) A new reverse reduce-error ensemble pruning algorithm. Appl Soft Comput 28:237–249
46. Zhou ZH, Wu J, Tang W (2002) Ensembling neural networks: many could be better than all. Artif Intell 137(1):239–263
47. Ni Z, Zhang C, Ni L (2016) A haze forecast method of selective ensemble based on glowworm swarm optimization algorithm. Int J Pattern Recognit Artif Intell 29(2):143–153
48. Cavalcanti GDC, Oliveira LS, Moura TJM et al (2016) Combining diversity measures for ensemble pruning. Pattern Recogn Lett 74:38–45
49. Ykhlef H, Bouchaffra D (2017) An efficient ensemble pruning approach based on simple coalitional games. Inf Fusion 34:28–42
50. Tang EK, Suganthan PN, Yao X (2006) An analysis of diversity measures. Mach Learn 65:247–271
51. Zhang L, Zhang B (2001) Good point set based genetic algorithm. Chin J Comput 24(9):917–922
52. Zhang C, Ni Z, Ni L et al (2016) Feature selection method based on multi-fractal dimension and harmony search algorithm and its application. Int J Syst Sci 47(14):3476–3486

Xuhui Zhu received a B.Sc. degree from School of Mathematics from Hefei University of Technology (HFUT), Hefei, PR China. Then he enter the School of management of HFUT, as a Ph.D. student. His main research interests include evolution computation, and machine learning.

Zhiwei Ni received a B.E. and M.E. degrees in computer software and theory from Anhui University (AHU). In June, 2002, he completed his Ph.D. degree in University of Science and Technology of China. Since 2002, he has become a professor and Ph.D. supervisor in HFUT. His main research interests include artificial intelligence, machine learning and cloud computing.

Meiying Cheng received B.E. degree from Anhui University of Technology, and M.E. degree from Ningbo University. Then she entered HFUT, as a Ph.D. student. Her main research interests include intelligence computation and data mining.

Feifei Jin received B.Sc. degree from Hefei Normal University, and M.Sc. degree from AHU. Then he entered HFUT, as a Ph.D. student. His interests include intelligence decision and intelligence computation.

Jingming Li received Ph.D. degree from HFUT. His interests include intelligence computation and data mining.

Gary Weckman received B.Sc. and M.E. degrees from University of Louisville, and Ph.D. degree from University of Cincinnati. Since 2002, he worked in Ohio University as a professor. His main research interests include artificial neural networks, safety and health engineering, decision support, and intelligent systems.