

Research Article

Hybrid Ensemble Pruning Using Coevolution Binary Glowworm Swarm Optimization and Reduce-Error

Xuhui Zhu,^{1,2} Zhiwei Ni,^{1,2} Pingfan Xia ^{1,2} and Liping Ni^{1,2}

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

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

Correspondence should be addressed to Pingfan Xia; xiapingfan@mail.hfut.edu.cn

Received 26 June 2020; Revised 27 September 2020; Accepted 14 October 2020; Published 31 October 2020

Academic Editor: Zhile Yang

Copyright © 2020 Xuhui Zhu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Ensemble pruning has been widely applied to improve the capacity of multiple learner system. Both diversity and classification accuracy of learners are considered as two key factors for achieving an ensemble with competitive classification ability. Considering that extreme learning machine (ELM) is characterized by excellent training rate and generalization capability, it is employed as the base classifier. For a multiple ELM system, when we increase its constituents' diversity, the mean accuracy of the whole members must be decreased. Therefore, a compromise between them can ensure that the ELMs remain good diversity and high precision, but finding the compromise brings a heavy computational burden. It is hard to look for the exact result via the searching of intelligent algorithms or pruning of diversity measures. On the basis, we propose a hybrid ensemble pruning approach employing coevolution binary glowworm swarm optimization and reduce-error (HEPCBR). Considering the good performance of reduce-error (RE) in selecting ELMs with high diversity and precision, we try to employ RE to choose the satisfactory ELMs from the generated ELMs. In addition, the constituents are further selected via the proposed coevolution binary glowworm swarm optimization, which are utilized to construct the promising ensemble. Experimental results indicate that, compared to other frequently used methods, the proposed HEPCBR achieves significantly superior performance in classification.

1. Introduction

Ensemble learning is widely used to improve classification ability in the areas such as image recognition [1, 2], intelligent detection [3, 4], and data mining [5, 6]. Its main idea is to aggregate the predictions of multiple learners so that a better predictive results can be attained [7, 8]. Considering that its remarkable enhancement in classification in contrast to single ones [9, 10], it has been widely used to several distinct applications, e.g., image processing [11, 12], medical diagnosis [13, 14], age prediction [15, 16], gene expression data classification [17, 18], and intrusion detection [19], but there is no such thing as a free lunch. When we employ massive learners for attaining an ensemble with good generalization ability, it needs to consume a large number of computing resources in the application of practical problems [20]. To reduce the consumption of computing resources, ensemble pruning has been studied to cope with it [9]. It

aims to aggregate a fraction of learners for increasing prediction accuracy via consuming less resource [10]. It is widely accepted that the same results are achieved by aggregating multiple identical learners, and the classification capacity cannot be enhanced [20]. Namely, the learners should be more diverse on data samples, so as to acquire the ensemble with higher predictive ability.

Nevertheless, combining diverse learners may result in a better or worse performance, because the ensemble composed of the members with great diversity and low precision may win low classification capacity [21]. Hence, the performance of ensemble is dramatically influenced by the two fundamental factors, i.e., diversity and precision of learners [22, 23]. As a consequence, we should select the learners with large diversity and high precision, and it can be converted to a combinatory optimization question [24]. Assume that there are n learners in an ensemble, and its size of nonempty subsets is very large [9, 25]. Hence, most algorithms cannot

do an exhaustive search in such a large subset [9]. Many ensemble pruning methods are proposed for coping with the selection of learners, and the existing works are presented as below.

As a rule, we can classify these techniques into several categories [10, 20]. Firstly, some scholars attempt to adopt some diversity measures to evaluate the learners, and those ones who satisfy some stability conditions can be chosen to build the ensemble via different pruning strategies [26, 27], e.g., kappa [28], reduce-error [9], complementarity [26], contribution [29], margin-based criteria [30], margin and diversity-based measure [31], and relevancy and complementary measures [32]. Secondly, the pruning of classifiers is a combinatory optimization question. Some researchers try to directly employ heuristic search algorithms to solve it [33, 34], for instance, a well-known pruning method, named GASEN [33], and GSOEP [34]. Thirdly, some clustering techniques are used to select the learners with good diversity for integration, and those ones are a series of representative learners [8]. Widely used techniques include k -means [35] and deterministic annealing [36]. Finally, many scholars study other different pruning methods to acquire the selection of learners, for example, frequent pattern [7], a randomized greedy selective strategy and ballot [37], greedy randomized dynamic pruning [38], confidence interval based on double-fault measure [39], graph coloring way [40], cost-sensitive rotation forest [17], induction of decision tree [41], and simple coalitional games [20].

As to the analysis above, it is easy to find that those existing works via diversity measures only remove a part of learners with bad performance, and there still exist learners with low precision, which may impact on the performance of the ensemble. To tackle the aforementioned issue, we combine diversity measures and heuristic algorithms to filter out redundant learners. The integration of them can seek for the optimal combination of learners. Where diversity measure is used to preprune a collection of learners with poor comprehensive capability [9]. They notably reduce its computational requirements. Then, we employ heuristic algorithms to select the learners with good comprehensive performance from the remaining learners to build the ensemble [33].

Considering the advantages of ELM [42, 43], we use it as the base learner in this work [39]. Reduce-error (RE) can perform better than other diversity measures in filtering out learners with bad performance [9]. Hence, RE can select some complementary learners with high classification capability, memorably downsize ELMs, and reduce the computing requirements. Therefore, RE denotes an appropriate preselection measure. In addition, glowworm swarm optimization (GSO) shows the superiorities [44–48] of strong robustness, easy to realize, and good capability of global search. Hence, it provides an efficient search strategy for further pruning redundant ELMs.

Therefore, we propose a novel hybrid pruning method called HEPCBR utilizing the integration of the proposed CBGSO and RE. Existing pruning methods generally find the subensemble with best capability employing GSO or RE separately, which cannot exactly find it, so the combination

of CBGSO and RE is employed to search for it. HEPCBR is an integrated method for dealing with ensemble pruning of multiple ELMs, which performs well in terms of searching for the optimal combination of ELMs. We remove some redundant ELMs using RE for reducing the size of ELMs and alleviating the computational complexity, and the presented CBGSO is also adopted to further seek for the optimal subensemble of ELMs. Therefore, the combination of CBGSO and RE can cope with the selection of ELMs.

This work's contributions have been presented as follows:

- (1) An ensemble pruning algorithm, named HEPCBR, which utilizes the integration of RE and the proposed CBGSO
- (2) We modify the basic GSO, named CBGSO, which has good convergence accuracy and high evolution velocity
- (3) RE takes full advantage of the diversity among ELMs and filters out a fraction of ELMs with poor comprehensive performance
- (4) Experimental results show the proposed approach can obtain a significant enhancement in classification capacity

2. CBGSO

The presented GSO was inspired via glowworms' luminous behavior [44, 45]. In GSO's searching process, an initial population composed of many glowworms is generated randomly, and then each glowworm moves forwards the one with larger luciferin. In addition, many individuals can come together the locations of the ones with high luciferin values [49]. Therefore, the optimal one has been gained. For an optimization question in a discrete space, the basic GSO cannot be directly used because of its fixed step.

To deal with the above question in a discrete space, we propose the CBGSO by improving the searching processes of the basic GSO. In CBGSO, its fixed step is modified, so as to make the algorithm efficiently seek for global optimum solution in the binary space. Crossover operation and collaborative operation are introduced into the basic GSO, which can create some new individuals and maintain the population diversity, so as to avoid getting in the local optimum and improve the convergence rate. Escaping operation is used in this work, which make those individuals who equal to the locally optimal solution have a chance to escape from the current area and avoid falling into a local optimum. The above operations are introduced into the basic GSO, so as to make it efficiently find the optimum solution. CBGSO has been displayed in the following part.

2.1. Improvement of Moving Way. Given a binary combinatorial optimization problem, GSO cannot directly do a search using a fixed step, and we need to modify its movement. To make sure that GSO can move in binary space, we try to update the position of a glowworm with a

certain probability [34, 48]. Its updating process can be formulated as

$$x_{ik}(t+1) = \begin{cases} r, & \text{if } \text{rand} \leq \frac{p}{3}, \\ x_{ik}(t), & \text{if } \frac{p}{3} < \text{rand} \leq \frac{2p}{3}, \\ x_{ij}(t), & \text{if } \text{rand} > \frac{p}{3}, \end{cases} \quad (1)$$

where $X_i(t) = (x_{i1}, x_{i2}, \dots, x_{in})$ is the current glowworm; $x_{ik}(t)$ is the k -th dimension of $X_i(t)$ at the t -th iteration; r is random integer 0 or 1; p is a predefined probability; and rand is a random number.

2.2. Crossover Operation. To enhance the searching rate of GSO, a crossover operation is introduced in GSO, which is inspired by GA [33]. In the population of glowworms, two selected glowworms perform a crossover, and two new offspring individuals can be created by exchanging part of elements of their parent individuals. That is, it has a chance to create new offspring individuals that perform better than before, and those individuals are not maintained in the population. If all the glowworms in the population perform the crossover, then the population performs diversely with a large probability. After that, those glowworms with bad performance will be eliminated. Namely, if the new generated glowworms can achieve better results, then the current individuals would be replaced. The crossover operation is described as

$$\begin{cases} X'_i(t) = (x_{i1}, x_{i2}, \dots, x_{jk}, \dots, x_{in}) \\ X'_j(t) = (x_{j1}, x_{j2}, \dots, x_{ik}, \dots, x_{jn}) \end{cases}, \quad \text{if } r_k < 0.5, \\ \begin{cases} X_i(t) = X'_i(t), & \text{if } Y'_i(t) > Y_i(t) \text{ and } Y'_i(t) > Y'_j(t), \\ X_i(t) = X'_j(t), & \text{if } Y'_j(t) > Y_i(t) \text{ and } Y'_j(t) > Y'_i(t), \end{cases} \end{cases} \quad (2)$$

where $X'_i(t), X'_j(t)$ represent the new generated glowworms who are created using the crossover operation; X_i, X_j express two different glowworms, respectively; and r_k declares a random number, $r_k \in (0, 1)$, $1 \leq k \leq n$.

2.3. Collaborative Operation. To avoid the remaining local optima, a collaborative operation is employed. Each glowworm in GSO can keep strong links with those glowworms inside its dynamic local-decision domain but has no link with that outside the dynamic local-decision domain. So, each glowworm should keep link with those individuals outside the dynamic local-decision domain. In this work, the population of glowworms can be partitioned into several subgroups. Each subgroup can keep strong links with other subpopulations. The optimal glowworm in one subgroup performs a crossover with the individual in another one, so as to gain new individuals with better performance via the exchange of their elements.

Assume that the population can be partitioned into S subgroups, $X_i^{s_1}(t) = (x_{i1}^{s_1}, x_{i2}^{s_1}, \dots, x_{in}^{s_1})$ is the optimal glowworm in the s_1 -th subgroup, $X_j^{s_2}(t) = (x_{j1}^{s_2}, x_{j2}^{s_2}, \dots, x_{jn}^{s_2})$ is the optimal glowworm in the s_2 -th one, and $1 \leq s_1 \neq s_2 \leq S$. The collaborative operation can be expressed as follows:

$$\begin{cases} X_i^{s_1 s_2}(t) = (x_{i1}^{s_1}, x_{i2}^{s_1}, \dots, x_{i(p'-1)}^{s_1}, x_{jp}^{s_2}, \dots, x_{in}^{s_2}), \\ X_j^{s_1 s_2}(t) = (x_{j1}^{s_2}, x_{j2}^{s_2}, \dots, x_{j(p'-1)}^{s_2}, x_{ip}^{s_1}, \dots, x_{in}^{s_1}), \\ \begin{cases} X_i^{s_1}(t) = X_i^{s_1 s_2}(t), & \text{if } Y_i^{s_1 s_2}(t) > Y_i^{s_1}(t) \text{ and } Y_i^{s_1 s_2}(t) > Y_j^{s_1 s_2}(t), \\ X_i^{s_1}(t) = X_j^{s_1 s_2}(t), & \text{if } Y_j^{s_1 s_2}(t) > Y_i^{s_1}(t) \text{ and } Y_j^{s_1 s_2}(t) > Y_i^{s_1 s_2}(t), \end{cases} \end{cases} \quad (3)$$

where $X_i^{s_1 s_2}(t), X_j^{s_1 s_2}(t)$ are the new generated glowworms, respectively; $Y_i^{s_1}(t), Y_i^{s_1 s_2}(t), Y_j^{s_1 s_2}(t)$ are the fitness values of $X_i^{s_1}(t), X_i^{s_1 s_2}(t), X_j^{s_1 s_2}(t)$, respectively.

2.4. Escaping Operation. To increase the probability of escaping from the local optima, escaping operation [50] is adopted. If the glowworm wins the same with the locally optimal solution, several random elements of the glowworm will be changed, so as to move forward other position for escaping from the local optima.

3. HEPCBR

In this section, HEPCBR is proposed using the combination of CBGSO and RE. First, the ELMs with poor comprehensive performance are pruned using RE for downsizing ELMs and notably reducing the computational overheads of the selection of ELMs; second, the subensemble of ELMs are chosen from the remaining ELMs using the proposed CBGSO. The proposed HEPCBR is described as follows.

3.1. Initial Pool. To maintain the good ability in classification by grouping multiple learners, these members should be diverse. When we try to gain the diverse learners, the weak one can be employed [40]. ELM is a weak learner [39] with fast training speed. We produce multiple diverse ELMs with a high efficiency. Therefore, ELM can be utilized to generate M ELMs via the bootstrap sampling method in bagging [51], and the initial pool of ELMs is constructed.

Huang et al. [42, 43] first published the first version of ELM for dealing with classification and regression problems [52]. When the hidden node of ELM is a certain value, the hidden weights and biases are generated randomly, so as to gain the unique solutions with a high efficiency.

For a training set $Z_{\text{train}} = \{(z_i, l_i), i = 1, 2, \dots, N_{\text{train}}\}$, then the model of ELM is expressed as follows:

$$\sum_{i=1}^L \beta_i g_i(z_j) = \sum_{i=1}^L \beta_i g(\omega_i z_j + b_i) = l_j, \quad (4)$$

where $z_i = [z_{i1}, z_{i2}, \dots, z_{in}]^T \in R^n, l_i = [l_{i1}, l_{i2}, \dots, l_{im}]^T \in R^m, 1 \leq i, j \leq N_{\text{train}}; L$ declares the hidden nodes' size; $g(z)$ exhibits an activate function; ω_i denotes the weight

matrix; β_i discloses the weight matrix of the output layer; and b_i reveals a threshold. Equation (4) is recharacterized in another way:

$$H^* \beta = L^*, \quad (5)$$

where $H^* = \begin{bmatrix} g(\omega_1 z_1 + b_1) & \cdots & g(\omega_L z_1 + b_L) \\ \vdots & \ddots & \vdots \\ g(\omega_1 z_N + b_1) & \cdots & g(\omega_L z_N + b_L) \end{bmatrix}_{N \times L}$, $L^* = [l_1^T \cdots l_N^T]_{N \times m}^T$, and $\beta = [\beta_1^T \cdots \beta_L^T]_{L \times m}^T$, and H^* manifests the hidden output matrix. Then, we can get

$$\beta = H^+ L^*. \quad (6)$$

where H^+ intimates the Moore–Penrose generalized inverse of H^* .

3.2. Preselection of ELMs. The pruning of learners is an NP-complete problem. To make an ensemble perform at its best, we should balance the accuracy and the diversity of ELMs, which will bring heavy computational burden. Therefore, it is almost impossible to directly perform the search for the optimal subensemble. To efficiently find the optimal subensemble, we need to downsize the redundant ELMs and reduce the computational burden of the selection of ELMs. Finally, the final ensemble can be obtained using heuristic algorithms with a high searching efficiency.

Reduce-error (RE) [9, 26, 28] selects the ELMs with good diversity and high precision via evaluating its contribution to the ensemble, and it should be utilized for removing some poorly performing ELMs. We introduce its basic concept in the following part.

For a training set $Z_{\text{train}} = \{(z_i, l_i), i = 1, 2, \dots, N_{\text{train}}\}$. Its samples are composed of a vector z_i and a label $l_i \in \{1, 2, \dots, c\}$. The predictive results are attained by classifying the samples in the training set Z_{train} , and $h(x)$ shows an ELM that can achieve the predictive label l_i in terms of z_i . The predictive result $E_H(z) \equiv \{h_m(z)\}_{m=1}^M$ of the original ensemble is obtained using the unweighted majority voting method, which is shown as follows [9, 26, 28]:

$$E_H(z) = \arg \max_{l \in C} \sum_{i=1}^M I(h_i(z) = l), \quad l \in C, \quad (7)$$

where $I(\cdot)$ represents an indicator function ($I(\text{true}) = 1$ and $I(\text{false}) = 0$); $h_m(z)$ expresses the class label classified by the i -th ELM; and $C = \{1, 2, \dots, c\}$ represents the set of sample labels.

The ELMs who can reduce the error rate of the candidate subensemble in the pool of unselected ELMs can be used to construct the final ensemble [9, 26, 28]. The first selected ELM is that with the highest classification ability on the training set. Assume u ELMs have been selected to constitute the ensemble S_u , and then the $u + 1$ th ELM is packaged into the ensemble S_{u+1} :

$$S_{u+1} = \arg \max_k \left(\sum_{(z,l) \in Z_{\text{train}}} I(E_{S_u \cup h_k}(z) = l) \right), \quad (8)$$

where $h_k \in F \setminus S_u$, $h_k \in F \cap h_k \notin S_u$.

In this section, we use RE to reduce a part of redundant ELMs for a goal of significantly alleviating the computation complexity. The ELM with the best classification accuracy is selected according to the prepruning method utilizing RE, which is the first ELM of the ensemble, and it grows by joining new ELMs in terms of equation (8). We attempt to determine the size of the preselection through the use of RE. Suppose that we retain the first M' ELMs acquired by RE to construct the prepruned ensemble, and then the M' members are used to be further selected.

Theorem 1. *The ensemble composed of the first M' ELMs extracted from the initial pool using RE, and its ensemble accuracy can be improved.*

Proof. M training subsets are generated employing a bootstrap sampling way, and multiple ELMs can be independently trained on the different training subsets. Hence, M ELMs are achieved $H = \{h_1, h_2, \dots, h_m\}$. The process of preselection based on RE is as follows.

First, the ELM h'_1 in H who does the best in classification is selected to add into the new subensemble $H' = \{h'_1\}$, and its accuracy is en_1 . Second, the ELM h'_2 is extracted from the unselected ELMs $H \setminus H'$, so as to attain that new subensemble $H' = \{h'_1, h'_2\}$ wins the highest precision according to equation (8), and its ensemble accuracy is en_2 . Finally, h'_3, h'_4, \dots, h'_m in the unselected ELMs $H \setminus H'$ can be added into the new subensemble H' one by one according to equation (8) in a similar fashion, and their ensemble accuracies en_3, en_4, \dots, en_m are achieved. Then, the new sequence of ELMs $H' = \{h'_1, h'_2, \dots, h'_m\}$ and its sequence of ensemble accuracies en_1, en_2, \dots, en_m are attained.

Let $M' = \arg \max (en_1, en_2, \dots, en_1, \dots, en_m)$, namely, $en_{M'}$ is the best.^{*i*} In addition, $h'_1, h'_2, \dots, h'_{M'}$ should be retained, because the ensemble composed of $h'_1, h'_2, \dots, h'_{M'}$ can achieve the highest accuracy. In a word, the first M' ELMs are utilized to form the prepruned ensemble using RE, and then its ensemble performance can be improved.

To verify the above theorem, the average ensemble accuracies are achieved by ordered bagging in terms of RE with 50, 150, and 250 ELMs on different datasets (Bupa, Vehicle, and CMC), respectively, as shown in Figure 1. We can observe from Figure 1 that the curves of the accuracies achieved by RE first rise up and then move down, with the ELMs' size growing. It can attain a peak when the number of ELMs is moderated. After that, as the size of ELMs increases, its precision gradually decreases. Namely, the ensemble ability can be enhanced after preselection based on RE. At the beginning, the ELMs have low diversity, so their accuracies go up with the size of ELMs rising, whereas as the number gets larger, the poor-performing ELMs have been added into the subensemble, which leads to the decline of the precision. From Figure 1, we can know that the ordered ensemble using RE achieves a maximal value before the ELMs' size reaches a certain value. Yang et al. [21] found that the heuristic algorithms can perform better in the selection of learners than other techniques when the size of learners is less than 25. As a

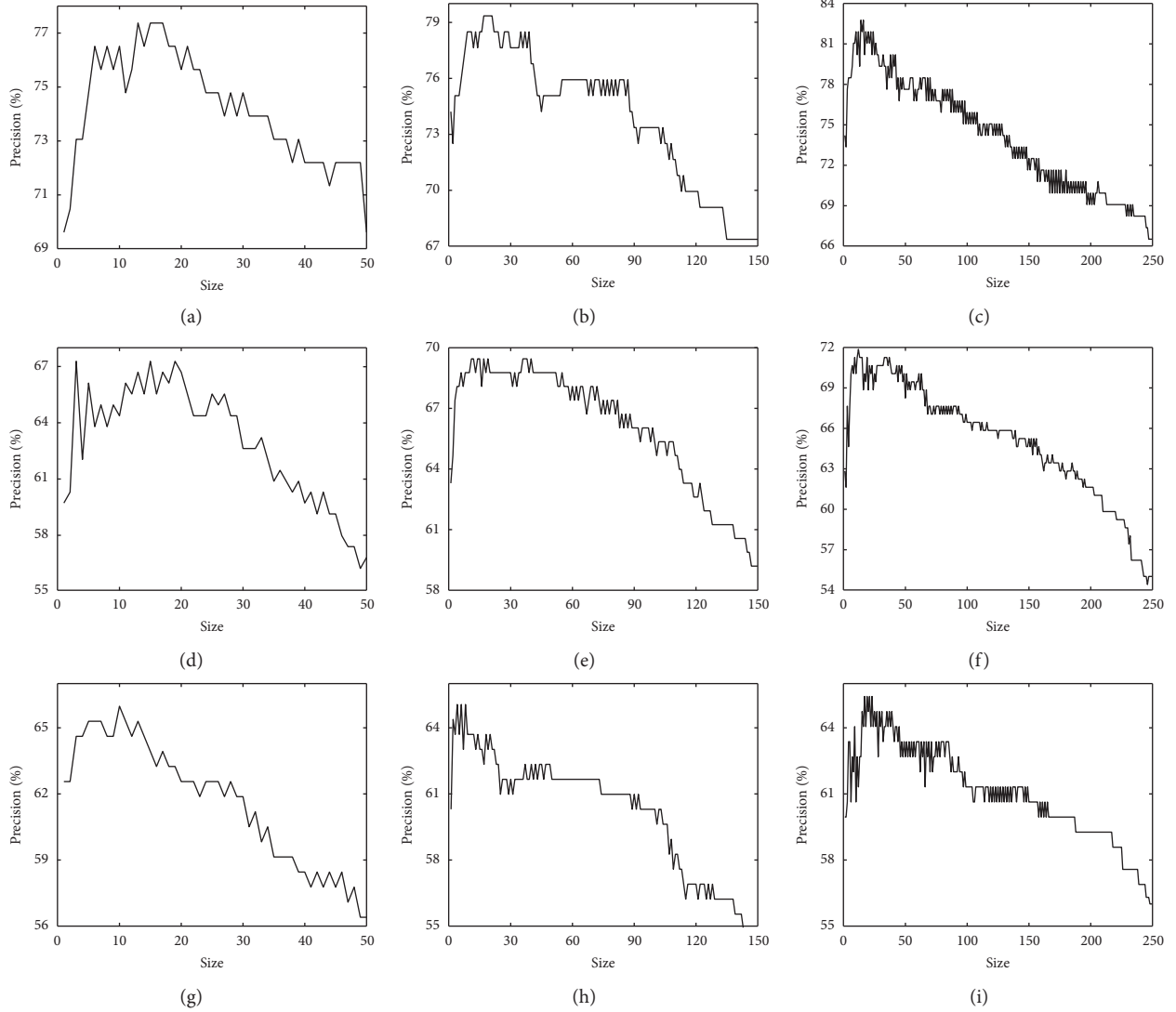


FIGURE 1: Mean precision of ordered ensemble using reduce-error on the different datasets. On Bupa: (a) size of 50, (b) size of 150, and (c) size of 250. On Vehicle: (d) size of 50, (e) size of 150, and (f) size of 250. On CMC: (g) size of 50, (h) size of 150, and (i) size of 250.

consequence, we select the first 25 well-performing ELMs, which are used for postselection via CBGSO.

3.3. Postselection of ELMs. Ensemble pruning is a combinatorial optimization problem. To calculate its fitness value, we take the classification accuracy f as its fitness function [33, 34], where f shows the classification precision between the predictive outputs and the actual labels, $f = (1/m) \sum_{j=1}^m \text{Acc}(\hat{y}_j, y_j)$, where if $\hat{y}_j = y_j$, then $\text{Acc}(\hat{y}_j, y_j) = 1$; otherwise, $\text{Acc}(\hat{y}_j, y_j) = 0$, m expresses the size of testing samples, \hat{y}_j, y_j represent the predictive result and its actual label on the j -th testing samples, respectively. In addition, $H' = \{h'_1, h'_2, \dots, h'_{25}\}$ shows the selected 25 ELMs gained by RE. In this work, each glowworm is characterized as a bit sequence. In the bit sequence, 1 discloses the ELM is selected, and 0 points to the ELM is not selected. So, we can cope with the pruning question of well employing CBGSO.

3.4. The Pseudocode of HEPCBR. The pseudocode of HEPCBR has been exhibited as follows (Algorithm 1).

4. Experiments

To evaluate the classification ability of HEPCBR, we select 25 UCI machine learning benchmark datasets to implement the massive experiments, and these datasets are presented in Table 1. Specifically, each experiment is repeated 30 times using different random seeds, and then the final classification accuracies are attained. The parameters of CBGSO are given in the following part [44, 45]: the $\rho = 0.4$, the $\gamma = 0.08$, the $n_t = 5$, the $p = 0.45$, the $S = 5$, and the $t_{\max} = 400$. We randomly divide the dataset into five equal parts, three of them as training, one of them as validation, and the other one as testing, where the ELMs were trained on the training datasets, the pruning process of ELMs is implemented on the validation datasets, and the final results are achieved on the testing datasets.

```

Inputs: HEPCBR's parameters.
Outputs: the subensemble  $X_{\text{opt}}$  and its precision  $f_{\text{opt}}$ .
(1) A pool composed of  $M$  ELMs is built,  $H \leftarrow \{h_1, h_2, \dots, h_M\}$ .
(2)  $H' \leftarrow \emptyset$ .
(3) The optimal ELM  $h'_1$  with the best classification ability is chosen.
(4)  $H' \leftarrow \{h'_1\}$ .
(5)  $H \leftarrow H \setminus \{h'_1\}$ .
(6) for  $i \leftarrow 2$  to  $M'$  do
(7) The ELM  $h'_i$  is chosen via equation (8).
(8)  $H' \leftarrow H' \cup \{h'_i\}$ .
(9)  $H \leftarrow H \setminus \{h'_i\}$ .
(10) end for
(11)  $N$  individuals can be produced randomly, and evaluate their fitness values  $f$ .
(12)  $X_{\text{opt}} \leftarrow \text{max fitness}(X_1, \dots, X_N)$ ,  $f_{\text{opt}} \leftarrow \text{max}\{f_1, \dots, f_N\}$ .
(13)  $t \leftarrow 1$ .
(14) while  $t \leq t_{\text{max}}$  do
(15) for  $s \leftarrow 1$  to  $S$  do
(16) for  $i \leftarrow 1$  to  $N/S$  do
(17) if  $f_i = f_{\text{opt}}$  then
(18) Perform escaping operation to generate a new glowworm.
(19) end if
(20) Find the objective individual  $X_j$ .
(21) Update the position of  $X_i$  via equation (1).
(22) Update parameters of  $X_i$ .
(23) Perform crossover operation to produce new individuals.
(24) end for
(25) Perform collaborative operation produce new individuals.
(26) end for
(27)  $X_{\text{opt}} \leftarrow \text{max fitness}(X_1, \dots, X_N)$ ,  $f_{\text{opt}} \leftarrow \text{max}\{f_1, \dots, f_N\}$ .
(28) end while
(29) return  $X_{\text{opt}}$  and  $f_{\text{opt}}$ 

```

ALGORITHM 1: HEPCBR.

4.1. Experimental Results. The proposed HEPCBR achieves the final ensemble using the combination of RE and the proposed CBGSO, and we attempt to verify that whether HEPCBR can perform better than any one of them. So, the predictive results of the proposed HEPCBR are compared to both RE and CBGSO in different sizes of ELMs, as shown in Tables 2 and 3. It is easy to see from Tables 2 and 3 that HEPCBR, via the integration of RE and CBGSO, wins a higher precision than any one of them (CBGSO or RE) on the most datasets using less size of ELMs. We can also observe from Tables 2 and 3 that, as the pool size increases, the prediction results achieved by HEPCBR and RE are getting better, but those attained by CBGSO are getting worse. We diagnose the causes, and we find that the computation complexity of the selection of ELMs exponentially increases as the size gets larger. For CBGSO, it is difficult to do an exhaustive search to find the exact subensemble. As a result, the preselection of ELMs can significantly lower the calculation complexity of the selection of ELMs. RE can select some well-performing ELMs, and the postselection of CBGSO can make the ensemble gain better results. Moreover, the classification ability of HEPCBR can attain a very small improvement as the initial size exceeds 150. Hence, 150 ELMs are used in this work.

4.2. Comparison with Other Methods. To test the classification ability of HEPCBR further, massive experiments were implemented by comparing it with other techniques: bagging [51], kappa [28], AGOB [26], POBE [27], DREP [22], DEELM [39], GASEN [33], GSOEP [34], MOAG [30], RREP [10], DMEP [40], EPSCG [20], DASEP [24], MDOEP [31], RCOA [32], and PEAD [41]. Bagging extracts the training samples with equal probability, and it can construct an initial pool composed of multiple learners with a good diversity. Kappa pruning achieves the ensemble by grouping the learners with their measures under a certain value. AGOB, POBE, and MOAG find the ordering of those learners in the ensemble works well. DREP first selects learner with the best classification ability and gradually increases the size of the ensemble which can make the ensemble achieve better precisions. DEELM chooses the learners with their measures falling into an interval. GASEN can assign a random weight to each learner. It adopts GA to evolve their parameters, and the learners whose weights exceed a predefined value can be utilized for combination. GSOEP employs GSO to directly filter out the redundant learners. RREP attained the ensemble by utilizing the retained learners and the removed ones gained by RE. DMEP employs GA to optimize five pairwise diversity matrices for a combining diversity measure, and select a part of learners in the light of the graph

TABLE 1: UCI datasets.

Datasets	Instances	Attributes	Classes
Heart	270	13	2
Cleveland	303	13	5
Column	310	6	2
Ecoli	336	8	8
Bupa	345	6	2
Wholesale	440	7	2
Forest	523	27	4
Balance	625	4	3
Diabetes	768	8	2
Vehicle	846	18	4
Tic-tac-toe	958	9	2
German	1000	20	2
QSAR	1055	41	2
Diabetic-r	1151	19	2
Phishing	1353	9	3
CMC	1473	9	3
Yeast	1484	9	10
Wineq-r	1599	11	6
Car	1728	6	4
CTG	2126	22	3
Segment	2310	19	7
Abalone	4177	8	3
Spambase	4601	57	2
Wineq-w	4898	11	7
Waveform	5000	21	3

TABLE 2: Precisions of HEPCBR in comparison with RE and CB.

Datasets	50						100						150					
	RE	n	CB	n	HE	n	RE	n	CB	n	HE	n	RE	n	CB	n	HE	n
Heart	74.29	25	76.83	19	78.07	12	77.59	25	74.38	44	80.49	13	81.35	25	75.31	71	83.12	13
Cleveland	56.82	25	56.26	25	59.34	10	58.62	25	54.44	47	60.25	10	57.04	25	53.09	75	57.95	15
Column	86.43	25	87.59	20	90.54	11	88.22	25	86.03	46	90.51	14	90.60	25	86.70	72	92.38	14
Ecoli	87.63	25	88.23	23	88.43	12	88.38	25	88.25	49	88.92	13	89.27	25	88.13	75	89.88	12
Bupa	74.41	25	74.97	22	77.91	13	78.88	25	78.18	44	81.05	14	79.22	25	75.98	70	81.72	14
Wholesale	89.33	25	90.84	18	91.37	13	91.82	25	89.48	43	93.19	13	90.98	25	88.76	71	91.68	14
Forest	92.47	25	93.01	23	93.96	11	92.52	25	91.16	46	93.44	14	94.64	25	91.50	69	94.94	15
Balance	92.72	25	93.12	21	93.04	12	91.84	25	91.36	50	92.48	13	91.84	25	91.76	73	92.80	11
Diabetes	69.41	25	72.27	19	72.55	10	71.78	25	70.67	45	73.62	11	74.33	25	69.68	68	75.17	14
Vehicle	65.85	25	67.26	19	69.95	13	69.43	25	68.50	42	71.25	16	69.77	25	66.14	65	71.59	15
Tic-tac-toe	93.95	25	94.88	22	95.62	13	94.94	25	95.00	43	96.25	13	96.29	25	93.53	76	96.66	15
German	78.60	25	78.10	21	81.20	11	80.90	25	77.50	44	82.80	13	80.90	25	74.20	73	82.20	12
QSAR	87.39	25	86.89	21	89.96	12	89.08	25	87.05	47	91.55	11	90.35	25	84.52	69	92.60	13
Diabetic-r	77.02	25	77.59	18	81.01	13	78.96	25	77.18	46	81.47	14	80.50	25	75.84	71	83.10	12
Phishing	86.39	25	86.53	24	88.54	11	87.60	25	85.82	48	88.78	11	88.53	25	85.54	76	89.35	13
CMC	61.14	25	62.10	20	63.79	12	62.33	25	61.29	41	64.30	12	63.40	25	59.86	63	64.75	14
Yeast	58.74	25	60.53	20	60.56	12	62.57	25	61.98	46	63.99	13	62.23	25	61.39	71	63.23	12
Wineq-r	57.30	25	61.56	19	60.34	11	61.86	25	59.13	41	63.95	13	64.07	25	58.74	67	66.26	13
Car	92.64	25	94.27	19	93.80	10	94.72	25	92.71	46	95.18	12	94.54	25	92.06	72	95.30	11
CTG	81.70	25	84.39	16	85.28	9	85.09	25	80.88	42	87.82	9	85.14	25	79.54	77	87.54	10
Segment	86.12	25	87.97	19	88.34	11	88.27	25	86.88	43	89.44	14	89.19	25	85.97	69	90.44	14
Abalone	57.11	25	59.99	19	58.58	11	60.60	25	57.21	41	61.64	12	58.74	25	59.97	71	59.96	12
Spambase	80.21	25	81.66	17	82.89	11	81.82	25	79.75	40	84.27	12	83.37	25	79.04	61	85.17	12
Wineq-w	47.32	25	51.20	15	50.12	10	51.05	25	48.48	41	53.14	13	51.18	25	48.60	69	53.76	12
Waveform	88.24	25	89.38	20	89.71	13	89.28	25	88.35	44	90.17	13	90.12	25	88.56	70	90.94	14

HE: HEPCBR; RE: reduce-error; CB: CBGSO.

coloring theory. EPSCG evaluates the learners' contribution to the ensemble in terms of the Banzhaf index of power. DASEP finds new measures by simultaneously considering

diversity and accuracy of learners, and acquires a good result in the selection of learners. MDOEP attains the final ensemble by the aggregation using margin and diversity-based

measure. RCOA ranks all the learners based on the relevancy and complementary measures in a descending order, and the final ensemble is achieved. PEAD selects a collection of learners extracted from a pool using the induction of decision tree, which are used to form the final ensemble.

We gain the classification precisions of the proposed HEPCBR, and other methods with 150 ELMs in the pool are displayed in Tables 4 and 5. “+/-/-,” respectively, indicates that HEPCBR achieves higher/neutral/lower precisions than other methods. It is clearly shown in Tables 4 and 5 that the proposed HEPCBR outperforms other state-of-the-art techniques on the most classification tasks, which shows its enhancements in classification. Table 6 displays the sizes of the final ensemble attained by all the pruning approaches. As shown in Table 6, HEPCBR utilizes less the size of ELMs to build the ensemble, but it uses more ELMs than DMEP. Additionally, HEPCBR gains better results than DMEP. As a whole, the proposed HEPCBR has acquired the expected results in classification.

In this section, we try to estimate the difference significance between the predictive accuracies gained by HEPCBR and that attained by other comparative methods, and we used the Wilcoxon rank sum test [40] with its level of 0.05. When the p value of two techniques is lower than 0.05, it expresses that there exists significant difference between them. Table 7 exhibits the p values acquired in the test between HEPCBR and other comparative approaches. Table 7 indicates that the p values have been below 0.05. It manifests the significance and the effectiveness of HEPCBR.

The execution time of the proposed HEPCBR and other comparative techniques is reported in Table 8. The observations from Table 8 flag up that HEPCBR uses much less time than GASEN, GSOEP, and DMEP, but utilizes more time than the retaining comparative methods. In HEPCBR, CBGSO is employed to extract the final ELMs from the remaining ones after pruning, which needs to calculate the classification accuracy of each candidate subensemble at each iteration. During the search of CBGSO, each candidate subensemble is expressed by a glowworm, and each glowworm needs to calculate its fitness value at least once per iteration. Therefore, it requires much more time. Those techniques with less running time achieve the selection of ELMs without repetitive iteration process and assessment. Nevertheless, they cannot attain good results in classification. HEPCBR cost less time than the mentioned three methods, it is the reason that those approaches directly selected the ELMs with good comprehensive performance without preselection, and it might not be enough to seek for the exact solution due to the enormous amount candidate subensembles. Moreover, the preselection process can notably alleviate the computational burden of the selection of ELMs. It is concluded that HEPCBR consumes much more time, but the remarkable improvements in classification are obtained.

The aforementioned comparative methods are based on bagging, which are attained using diversity measures with different strategies or heuristic algorithms. In addition, we further compare the proposed HEPCBR with the more state-of-the-art approaches by considering other factors, which

are not the same category with the aforementioned comparative methods: C-RoF [17], CS-D-ELM [18], KPCA-RoF [4], and DyPReVNsGraspEnS [38]. C-RoF introduces misclassification, test, and rejection costs for overcoming that some existing works neglect the classification costs, and it achieves a good result. CS-D-ELM extends the D-ELM employing misclassification cost of the learner and embeds rejection cost into it to increase the classification stability. KPCA-RoF integrates the KPCA and RoF for linearly inseparable dataset classification problems, and it enhances the classification capacity of RoF on the nonlinear fractional datasets. DyPReVNsGraspEnS realizes random multistart search, and it can avoid falling into local optimal solutions and possesses a high probability to exactly find the ensemble with better performance. The classification results achieved by different approaches are presented in Table 9. We can see from Table 9 that the accuracies gained by HEPCBR are higher than other four state-of-the-art methods. Then, we verify the significance between HEPCBR and other four approaches using the Wilcoxon rank sum test, which is displayed in Table 10. From Table 10, all the p values are less than 0.05. So, we can conclude that the proposed HEPCBR has powerful advantages over other four techniques in classification, and it is significant and effective.

4.3. Parameter Analysis. In HEPCBR, we employ CBGSO to select the ELMs with good diversity and high accuracy from the remaining ones after prepruning. To make the proposed CBGSO perform well, we try to analyze its parameters. In addition, to assess its convergence speed and precision, the comparative analyses have been presented as follows: MGSO [48], MBGSO [47], MDGSO [34], IBFS [50], MBFS [53], MFS [54], and GA [33]. We implemented the experiments on two datasets, and we adopt different methods in search of the optimal subensemble from the remaining ELMs using RE with the pool size of 150.

It is clearly observed in Figure 2 that, as a whole, CBGSO attains better convergence accuracy and higher evolution velocity than other seven comparative algorithms, i.e., CBGSO performs better than others in terms of the speed and the precision of convergence. It is also easy to find that the accuracies acquired by CBGSO rise up and stay stable. The results achieved by CBGSO just can attain a slight gain in classification, when the number of iterations is above 400. So, it is 400. In addition, when CBGSO is utilized to search for the final ensemble in HEPCBR, it costs less time than other seven heuristic algorithms, which is displayed in Table 11. Table 11 demonstrates that CBGSO performs higher efficiency of searching than other heuristic algorithms. Hence, CBGSO can work well when it comes to searching speed and efficiency. The accuracies mount and level off, as the population size increases, which is rendered in Figure 3(a). We can see from Figure 3(a) that, when the population size of glowworms is above 25, it will greatly increase the computation complexity with less improvements in classification. Hence, the size is 25. Figure 3(b) analyzes the affection of the initial range on the performance of CBGSO. We can observe from Figure 3(b) that it performs

TABLE 5: Classification precisions won by other methods with 150 ELMs.

Datasets	MOAG	RREP	DMEP	EPSCG	DASEP	MDOEP	RCOA	PEAD
Heart	71.63	81.40	78.51	76.12	78.85	77.52	77.38	76.54
Cleveland	53.26	56.24	56.60	54.81	55.84	55.76	56.92	57.88
Column	83.43	91.43	88.76	86.94	86.27	85.82	86.39	85.98
Ecoli	87.59	89.60	89.83	87.14	89.96	86.97	88.46	89.22
Bupa	70.45	79.61	77.30	75.15	76.64	77.97	76.11	75.26
Wholesale	87.69	92.05	91.24	87.48	89.82	87.76	88.52	87.50
Forest	90.50	92.80	92.83	90.45	94.94	94.94	94.82	93.75
Balance	91.96	93.00	93.32	91.72	93.20	91.40	91.68	92.44
Diabetes	69.37	75.89	74.56	72.52	72.66	71.38	71.62	70.27
Vehicle	60.32	71.08	67.57	66.16	66.59	67.63	68.86	65.27
Tic-tac-toe	92.38	96.66	95.41	96.66	94.29	96.66	95.17	94.92
German	75.85	81.50	81.15	77.30	78.60	78.70	78.00	77.15
QSAR	84.48	90.29	88.97	86.58	87.50	88.36	87.33	86.52
Diabetic-r	72.46	81.48	78.79	77.24	77.80	79.99	78.08	76.34
Phishing	85.74	87.91	89.06	85.11	88.03	86.52	86.77	87.65
CMC	55.87	63.31	61.44	58.95	59.15	58.12	58.06	57.60
Yeast	59.46	62.11	62.32	58.60	60.99	57.99	58.98	59.63
Wineq-r	57.74	63.80	62.80	57.08	60.86	59.68	59.09	57.82
Car	92.42	94.65	94.71	92.13	93.96	91.47	93.79	93.13
CTG	81.21	87.54	86.90	82.54	85.21	87.54	86.39	87.54
Segment	85.13	89.03	87.49	84.92	86.30	86.10	85.63	84.13
Abalone	58.19	60.33	60.13	57.33	58.93	59.20	60.50	59.96
Spambase	76.31	84.15	82.24	78.83	79.45	81.34	79.10	79.91
Wineq-w	47.89	51.57	51.65	48.92	49.94	49.48	50.40	49.52
Waveform	87.43	90.26	89.36	87.28	88.17	89.34	88.67	88.29
+/-	0/0/25	2/2/20	1/0/24	0/1/24	1/1/23	0/3/22	1/0/24	0/1/24

TABLE 6: Sizes of the selected ELMs via other methods with 150 ELMs.

Datasets	HE	Ba	Ka	AG	PO	DR	DE	GA	GS	MO	RR	DM	EP	DA	MD	MR	PE
Heart	13	150	30	17	30	75	77	56	71	30	18	5	31	29	30	26	37
Cleveland	15	150	30	21	29	75	61	37	76	23	17	2	15	26	30	25	22
Column	14	150	30	23	30	75	81	44	71	35	21	4	17	13	30	32	47
Ecoli	12	150	30	20	29	75	76	52	79	26	16	2	19	15	30	25	32
Bupa	14	150	30	25	32	75	82	71	70	26	24	3	19	23	30	24	52
Wholesale	14	150	30	20	31	75	81	54	65	33	17	3	12	19	30	29	32
Forest	15	150	30	18	30	75	83	69	71	31	18	9	27	30	30	26	23
Balance	11	150	30	20	30	75	77	61	72	8	16	2	17	24	30	23	22
Diabetes	14	150	30	20	29	75	76	28	69	35	16	6	50	14	30	24	27
Vehicle	15	150	30	22	31	75	82	67	72	32	19	8	69	30	30	23	41
Tic-tac-toe	15	150	30	15	31	75	82	73	69	31	17	8	36	29	30	19	39
German	12	150	30	15	30	75	79	39	73	36	16	5	31	17	30	18	29
QSAR	13	150	30	19	30	75	81	67	72	27	21	9	13	15	30	28	22
Diabetic-r	12	150	30	22	32	75	81	50	62	34	17	7	15	14	30	29	47
Phishing	13	150	30	19	31	75	78	29	73	24	17	4	21	34	30	30	35
CMC	14	150	30	26	34	75	81	58	60	24	18	8	38	13	30	23	23
Yeast	12	150	30	20	31	75	79	36	74	31	18	6	11	18	30	26	23
Wineq-r	13	150	30	20	32	75	80	29	69	35	18	4	17	33	30	23	24
Car	11	150	30	18	30	75	79	40	73	26	16	5	38	21	30	26	26
CTG	10	150	30	17	25	75	67	31	55	32	16	4	51	17	30	23	23
Segment	14	150	30	23	32	75	82	66	70	30	20	9	45	19	30	24	23
Abalone	12	150	30	26	32	75	79	14	65	28	21	5	16	31	30	23	28
Spambase	12	150	30	16	28	75	72	44	71	33	18	6	98	16	30	19	24
Wineq-w	12	150	30	26	32	75	71	27	68	29	20	4	40	17	30	24	32
Waveform	14	150	30	24	31	75	81	71	61	36	21	5	29	24	30	25	29

HE: HEPCBR; Ba: bagging; Ka: kappa; AG: AGOB; PO: POBE; DR: DREP; DE: DEELM; GA: GASEN; GS: GSOEP; MO: MOAG; RR: RREP; DM: DMEP; EP: EPSCG; DA: DASEP; MD: MDOEP; RC: RCOA; PE: PEAD.

TABLE 7: The p values of the Wilcoxon rank sum test.

Comparison	p value	Significant or not
HEPCBR vs bagging	0.00123	Yes
HEPCBR vs kappa	0.00123	Yes
HEPCBR vs AGOB	0.00123	Yes
HEPCBR vs POBE	0.00123	Yes
HEPCBR vs DREP	0.00123	Yes
HEPCBR vs DEELM	0.00123	Yes
HEPCBR vs GASEN	0.00123	Yes
HEPCBR vs GSOEP	0.00123	Yes
HEPCBR vs MOAG	0.00123	Yes
HEPCBR vs RREP	0.02327	Yes
HEPCBR vs DMEP	0.00286	Yes
HEPCBR vs EPSCG	0.00182	Yes
HEPCBR vs DASEP	0.00267	Yes
HEPCBR vs MDOEP	0.00401	Yes
HEPCBR vs RCOA	0.00157	Yes
HEPCBR vs PEAD	0.00270	Yes

TABLE 8: Execution time of different techniques.

Datasets	Execution time (s)																
	HE	Bag	Ka	AG	PO	DR	DF	GA	GSO	MO	RRE	DivP	SC	SD	MD	MR	PE
Heart	6.32	0.47	0.50	4.09	0.49	1.27	0.51	16.22	16.71	0.50	4.23	18.12	0.50	0.65	0.56	1.39	0.49
Cleveland	25.29	0.42	0.44	8.09	0.45	3.48	0.45	39.80	39.34	0.59	8.19	42.48	0.59	0.82	0.55	1.67	0.42
Column	6.73	0.51	0.54	4.70	0.53	1.42	0.55	18.04	18.34	0.55	4.78	23.52	0.55	0.70	0.63	1.47	0.51
Ecoli	34.64	0.45	0.47	12.53	0.48	5.09	0.49	57.57	57.97	0.72	12.59	65.55	0.72	1.04	0.74	1.94	0.54
Bupa	7.72	0.48	0.51	5.17	0.49	1.53	0.52	20.07	19.47	0.53	5.82	22.43	0.53	0.70	0.60	1.67	0.57
Wholesale	9.56	0.69	0.73	6.59	0.69	1.92	0.74	25.26	24.82	0.74	6.67	26.17	0.74	0.95	0.87	1.83	0.81
Forest	24.43	0.81	0.84	7.73	0.82	3.67	0.85	40.52	39.36	0.98	8.21	45.87	0.97	1.19	0.99	2.23	0.85
Balance	21.46	0.87	0.89	7.31	0.90	3.54	0.90	38.75	38.76	1.01	7.51	47.92	1.01	1.22	1.05	2.17	1.01
Diabetes	10.54	0.69	0.73	6.76	0.71	1.98	0.74	25.85	25.51	0.75	6.83	33.71	0.75	0.97	0.85	1.84	0.71
Vehicle	37.70	1.11	1.16	12.50	1.23	5.94	1.17	64.70	63.03	1.39	12.97	69.21	1.38	1.75	1.33	3.70	1.48
Tic-tac-toe	12.48	1.50	1.55	9.35	1.60	3.16	1.56	32.17	32.37	1.58	9.51	43.53	1.57	1.85	1.75	3.58	2.27
German	6.65	1.27	1.29	5.29	1.29	2.10	1.30	19.06	19.03	1.30	5.82	25.28	1.30	1.45	1.37	2.02	1.07
QSAR	7.22	1.65	1.68	5.95	1.65	2.58	1.69	20.03	20.56	1.69	6.05	26.67	1.69	1.86	1.79	2.36	1.38
Diabetic-r	7.59	1.15	1.19	5.94	1.16	2.21	1.20	21.38	21.79	1.20	5.94	23.17	1.20	1.38	1.28	2.27	1.17
Phishing	23.25	2.29	2.32	9.94	2.35	5.48	2.33	49.93	47.89	2.47	10.21	59.35	2.47	2.72	2.50	2.91	1.66
CMC	26.57	2.22	2.26	10.83	2.33	5.84	2.27	52.99	53.62	2.42	10.98	67.29	2.42	2.70	2.39	2.93	1.27
Yeast	59.84	1.06	1.09	19.42	1.16	8.11	1.10	87.86	88.76	1.47	20.39	98.21	1.47	1.95	1.36	3.04	0.95
Wineq-r	45.94	2.00	2.04	16.53	2.20	7.80	2.05	77.23	76.34	2.33	16.91	86.12	2.33	2.74	2.24	3.86	1.80
Car	36.04	3.08	3.12	14.12	3.38	7.53	3.13	64.04	62.53	3.34	14.53	69.18	3.33	3.66	3.39	4.45	2.86
CTG	36.07	1.53	1.57	12.42	1.58	6.03	1.58	64.75	66.94	1.78	12.67	65.62	1.77	2.11	1.82	3.26	1.45
Segment	81.08	3.20	3.25	27.39	3.19	12.77	3.26	115.91	115.46	3.76	28.15	135.40	3.75	4.42	3.74	7.42	4.10
Abalone	47.83	3.81	3.87	18.40	3.91	9.87	3.88	90.25	88.63	4.15	18.49	99.82	4.14	4.60	4.11	7.48	5.03
Spambase	19.30	5.49	5.58	17.89	5.55	8.21	5.59	57.69	57.97	5.61	18.11	68.33	5.61	6.05	5.85	7.87	5.73
Wineq-w	85.03	4.91	4.99	31.50	4.96	15.52	5.00	140.39	138.23	5.52	32.32	165.14	5.51	6.26	5.33	7.99	4.60
Waveform	58.34	6.36	6.44	23.66	6.61	13.63	6.45	110.49	107.94	6.76	25.01	131.15	6.76	7.30	6.85	9.07	6.26

HE: HEPCBR; Ba: bagging; Ka: kappa; AG: AGOB; PO: POBE; DR: DREP; DE: DEELM; GA: GASEN; GS: GSOEP; MO: MOAG; RR: RREP; DM: DMEP; EP: EPSCG; DA: DASEP; MD: MDOEP; RC: RCOA; PE: PEAD.

TABLE 9: Comparison of classification performance with other state-of-the-art methods.

Datasets	HE	C-RoF	CS-D	KPCA	DyP
Heart	83.12	78.62	77.55	79.21	80.84
Cleveland	57.95	57.05	57.66	56.05	57.86
Column	92.38	89.07	88.48	88.73	89.43
Ecoli	89.88	90.36	89.82	89.65	90.00
Bupa	81.72	77.52	77.69	78.22	80.46
Wholesale	91.68	89.38	91.36	91.58	92.84
Forest	94.94	92.58	93.70	94.08	93.28

TABLE 9: Continued.

Datasets	HE	C-RoF	CS-D	KPCA	DyP
Balance	92.80	92.80	92.48	92.48	92.80
Diabetes	75.17	74.71	74.60	75.41	74.81
Vehicle	71.59	68.18	68.91	68.43	70.56
Tic-tac-toe	96.66	94.32	95.51	95.57	96.57
German	82.20	78.00	79.20	78.40	80.80
QSAR	92.60	89.94	89.32	90.32	90.81
Diabetic-r	83.10	75.21	77.64	76.61	80.76
Phishing	89.35	88.35	89.83	89.24	89.71
CMC	64.75	61.02	61.28	60.88	62.28
Yeast	63.23	61.65	62.73	61.65	62.41
Wineq-r	66.26	63.63	61.63	62.25	61.50
Car	95.30	94.44	94.91	93.85	94.37
CTG	87.54	84.18	85.87	84.74	86.53
Segment	90.44	87.30	87.91	88.01	89.72
Abalone	59.96	57.93	57.72	58.78	58.36
Spambase	85.17	80.54	81.44	81.22	83.28
Wineq-w	53.76	50.23	51.30	52.58	51.21
Waveform	90.94	88.45	88.70	88.68	88.84

HE: HEPCBR; CS-D: CS-D-ELM; KPCA: KPCARoF; DyP: DyPreVNsGraspEnS.

TABLE 10: The p values of the Wilcoxon rank sum test.

Comparison	p value	Significant or not
HEPCBR vs C-RoF	0.00235	Yes
HEPCBR vs CS-D	0.00254	Yes
HEPCBR vs KPCA	0.00200	Yes
HEPCBR vs DyP	0.01716	Yes

CS-D: CS-D-ELM; KPCA: KPCA-RoF; DyP: DyPreVNsGraspEnS.

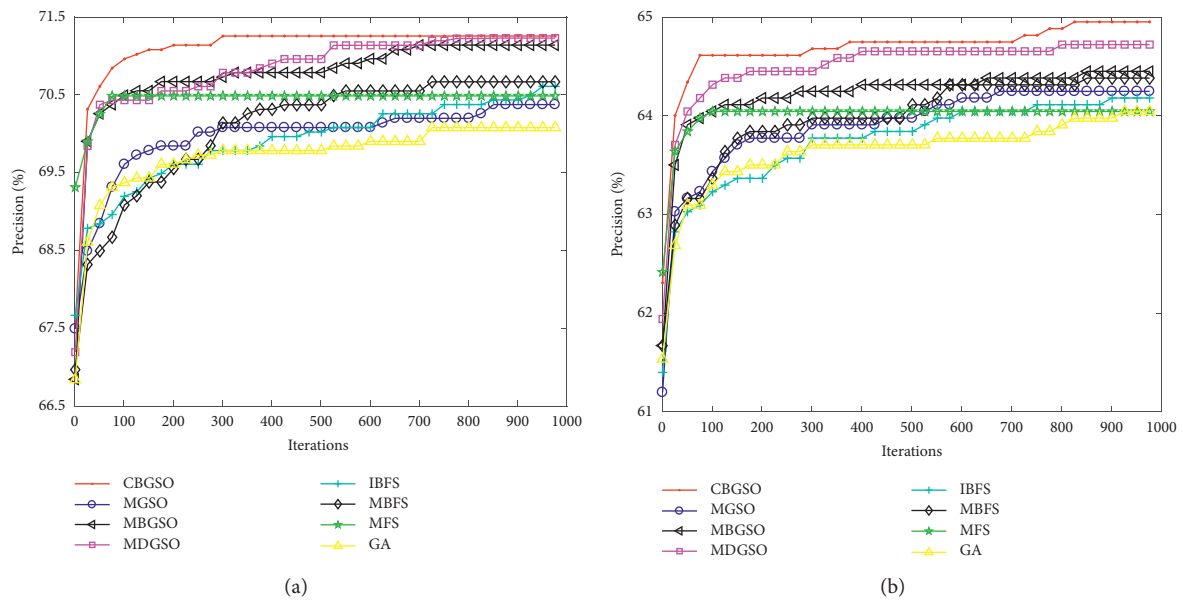


FIGURE 2: The affection of iteration on the convergence rate of algorithms. (a) Vehicle. (b) CMC.

TABLE 11: Comparison of execution time with different heuristic algorithms.

Methods	Vehicle (s)	CMC (s)
CBGSO	33.16	25.09
MGSO	36.41	25.89
MBGSO	36.64	25.62
MDGSO	36.85	26.18
IBFS	87.26	56.07
MBFS	57.70	40.03
MFS	93.15	60.66
GA	33.55	25.37

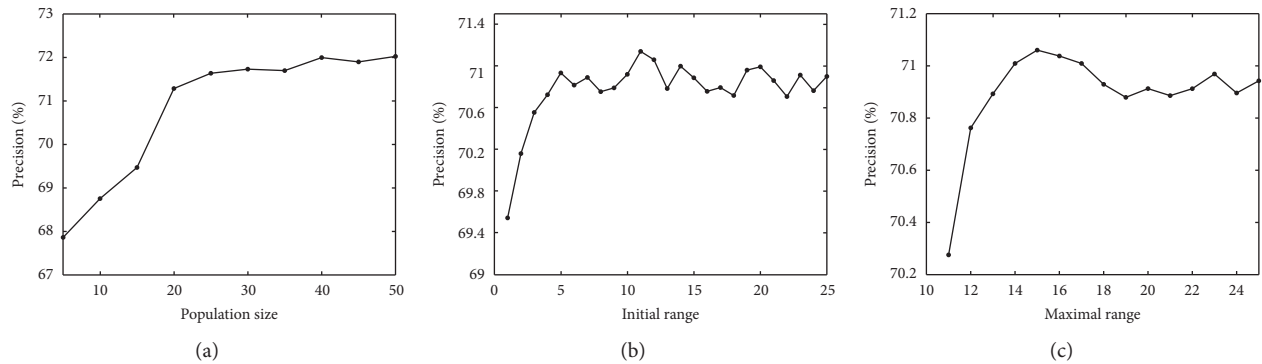


FIGURE 3: The affection of other parameters on the performance of CBGSO. (a) Population. (b) Initial range. (c) Maximal range.

at its best when the initial range is 11. Hence, the initial range is 11. Figure 3(c) reveals the affection of the maximal one on the performance of CBGSO. It is more than the initial one. The observations from Figure 3(c) disclose that the performance of CBGSO peaks at 15. So, we advise it is set as 15.

5. Conclusions

Ensemble pruning extracts a subset of ELMs extracted from a constructed pool of ELMs for achieving better predictive results and efficiency. To make an ensemble perform well, it must consist of accurate and diverse ELMs. Attaining that is a combinatory optimization question, and it has high complexity. To address this issue, HEPCBR is presented via the fusion of RE and CBGSO, the prepruning strategy using RE can avoid heavy computing burdens of the selection of ELMs, and then the ELMs with good comprehensive performance can be efficiently selected by CBGSO. Experimental study utilizing 25 UCI datasets has demonstrated that the proposed HEPCBR outperforms bagging and other algorithms, and it also indicates the effectiveness and significance. Furthermore, the presented CBGSO can look for the better results than other methods. The proposed HEPCBR brings forward a new study way for the pruning question of ELMs.

With respect to future research directions, we aim to design a new combining diversity measure, which can minimize the redundant ELMs. Therefore, it can provide these well-performing ELMs for further pruning.

Data Availability

The data about the UCI datasets used are from the website <https://archive.ics.uci.edu/ml/index>.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was funded by the Anhui Provincial Natural Science Foundation under grant no. 1908085QG298, the National Nature Science Foundation of China under grant nos. 91546108 and 61806068, the Fundamental Research Funds for the Central Universities under grant nos. JZ2019HGTA0053 and JZ2019HGBZ0128, the Anhui Provincial Science and Technology Major Projects under grant no. 201903a05020020, and the Open Research Fund Program of Key Laboratory of Process Optimization and Intelligent Decision-Making (Hefei University of Technology), Ministry of Education.

References

- [1] I. Rida, R. Herault, G. L. Marcialis, and G. Gasso, "Palmprint recognition with an efficient data driven ensemble classifier," *Pattern Recognition Letters*, vol. 126, pp. 21–30, 2019.
- [2] X. C. Yu, Z. Zhang, L. Wu et al., "Deep ensemble learning for human action recognition in still images," *Complexity*, vol. 2020, Article ID 9428612, 23 pages, 2020.

- [3] A. A. Aburomman and M. B. I. Reaz, "A survey of intrusion detection systems based on ensemble and hybrid classifiers," *Computers & Security*, vol. 65, pp. 135–152, 2017.
- [4] H. Lu, Y. Meng, K. Yan, and Z. Gao, "Kernel principal component analysis combining rotation forest method for linearly inseparable data," *Cognitive Systems Research*, vol. 53, pp. 111–122, 2019.
- [5] G. Song and Q. Dai, "A novel double deep ELMs ensemble system for time series forecasting," *Knowledge-Based Systems*, vol. 134, pp. 31–49, 2017.
- [6] Y. Yang, X. F. Liu, Q. W. Ye, and D. P. Tao, "Ensemble learning-based person re-identification with multiple feature representations," *Complexity*, vol. 2018, Article ID 5940181, 12 pages, 2018.
- [7] H. Zhou, X. Zhao, and X. Wang, "An effective ensemble pruning algorithm based on frequent patterns," *Knowledge-Based Systems*, vol. 56, pp. 79–85, 2014.
- [8] A. Onan, S. Korukoğlu, and H. Bulut, "A hybrid ensemble pruning approach based on consensus clustering and multi-objective evolutionary algorithm for sentiment classification," *Information Processing & Management*, vol. 53, no. 4, pp. 814–833, 2017.
- [9] G. Martínez-Muñoz, D. Hernández-Lobato, and A. Suárez, "An analysis of ensemble pruning techniques based on ordered aggregation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 245–259, 2009.
- [10] Q. Dai, T. Zhang, and N. Liu, "A new reverse reduce-error ensemble pruning algorithm," *Applied Soft Computing*, vol. 28, pp. 237–249, 2015.
- [11] M. Han, X. Yang, and E. Jiang, "An extreme learning machine based on cellular automata of edge detection for remote sensing images," *Neurocomputing*, vol. 198, pp. 27–34, 2016.
- [12] Y. J. Xu and X. J. Liu, "Interactive algorithms in complex image processing systems based on big data," *Complexity*, vol. 2020, Article ID 5929584, 9 pages, 2020.
- [13] P. Cunningham, J. Carney, and S. Jacob, "Stability problems with artificial neural networks and the ensemble solution," *Artificial Intelligence in Medicine*, vol. 20, no. 3, pp. 217–225, 2000.
- [14] Z.-H. Zhou, Y. Jiang, Y.-B. Yang, and S.-F. Chen, "Lung cancer cell identification based on artificial neural network ensembles," *Artificial Intelligence in Medicine*, vol. 24, no. 1, pp. 25–36, 2002.
- [15] G. Antipov, M. Baccouche, S.-A. Berrani, and J.-L. Dugelay, "Effective training of convolutional neural networks for face-based gender and age prediction," *Pattern Recognition*, vol. 72, pp. 15–26, 2017.
- [16] K. Li, J. Xing, W. Hu, and S. J. Maybank, "D2C: deep cumulatively and comparatively learning for human age estimation," *Pattern Recognition*, vol. 66, pp. 95–105, 2017.
- [17] H. Lu, L. Yang, K. Yan, Y. Xue, and Z. Gao, "A cost-sensitive rotation forest algorithm for gene expression data classification," *Neurocomputing*, vol. 228, pp. 270–276, 2017.
- [18] Y. Q. Lu, H. J. Lu, K. Yan, H. Xia, and C. An, "Applying cost-sensitive extreme learning machine and dissimilarity integration to gene expression data classification," *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 8056253, 9 pages, 2016.
- [19] X. Yang, D. Lo, X. Xia, and J. Sun, "TLEL: a two-layer ensemble learning approach for just-in-time defect prediction," *Information and Software Technology*, vol. 87, pp. 206–220, 2017.
- [20] H. Ykhlef and D. Bouchaffra, "An efficient ensemble pruning approach based on simple coalitional games," *Information Fusion*, vol. 34, pp. 28–42, 2017.
- [21] C. Yang, X. C. Yin, H. W. Hao et al., "Learner ensemble with diversity: effectiveness analysis and ensemble optimization," *Acta Automatica Sinica*, vol. 40, no. 4, pp. 660–674, 2014.
- [22] N. Li, Y. Yu, and Z. H. Zhou, "Diversity regularized ensemble pruning," *Machine Learning and Knowledge Discovery in Databases*, Springer, Berlin, Germany, pp. 330–345, 2012.
- [23] X. Zhu, Z. Ni, M. Cheng, F. Jin, J. Li, and G. Weckman, "Selective ensemble based on extreme learning machine and improved discrete artificial fish swarm algorithm for haze forecast," *Applied Intelligence*, vol. 48, no. 7, pp. 1757–1775, 2018.
- [24] Q. Dai, R. Ye, and Z. Liu, "Considering diversity and accuracy simultaneously for ensemble pruning," *Applied Soft Computing*, vol. 58, pp. 75–91, 2017.
- [25] I. Partalas, G. Tsoumakas, and I. Vlahavas, "Pruning an ensemble of classifiers via reinforcement learning," *Neurocomputing*, vol. 72, no. 7–9, pp. 1900–1909, 2009.
- [26] G. Martínez-Muñoz and A. Suárez, "Aggregation ordering in bagging," in *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications*, pp. 258–263, Innsbruck, Austria, February 2004.
- [27] G. Martínez-Muñoz and A. Suárez, "Pruning in ordered bagging ensembles," in *Proceedings of the Twenty-Third International Conference on Machine Learning*, pp. 609–616, Pittsburgh, PA, USA, June 2006.
- [28] D. D. Margineantu and T. G. Dietterich, "Pruning adaptive boosting," in *Proceedings of the Fourteenth International Conference on Machine Learning*, vol. 97, pp. 211–218, Nashville, TN, USA, July 1997.
- [29] Z. Lu, X. Wu, X. Zhu et al., "Ensemble pruning via individual contribution ordering," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 871–880, Washington, DC, USA, July 2010.
- [30] L. Guo and S. Boukir, "Margin-based ordered aggregation for ensemble pruning," *Pattern Recognition Letters*, vol. 34, no. 6, pp. 603–609, 2013.
- [31] H. Guo, H. Liu, R. Li, C. Wu, Y. Guo, and M. Xu, "Margin & diversity based ordering ensemble pruning," *Neurocomputing*, vol. 275, pp. 237–246, 2018.
- [32] X. Xia, T. Lin, and Z. Chen, "Maximum relevancy maximum complementary based ordered aggregation for ensemble pruning," *Applied Intelligence*, vol. 48, no. 9, pp. 2568–2579, 2018.
- [33] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: many could be better than all," *Artificial Intelligence*, vol. 137, no. 1–2, pp. 239–263, 2002.
- [34] Z. W. Ni, C. Zhang, and L. P. Ni, "Haze forecast method of selective ensemble based on glowworm swarm optimization algorithm," *Pattern Recognition and Artificial Intelligence*, vol. 29, no. 2, pp. 143–153, 2016.
- [35] S. Xu, K.-S. Chan, J. Gao et al., "An integrated K-means–Laplacian cluster ensemble approach for document datasets," *Neurocomputing*, vol. 214, pp. 495–507, 2016.
- [36] Chaomurilige, J. Yu, M. S. Yang et al., "Deterministic annealing Gustafson-Kessel fuzzy clustering algorithm," *Information Sciences*, vol. 417, pp. 435–453, 2017.
- [37] Q. Dai, "A novel ensemble pruning algorithm based on randomized greedy selective strategy and ballot," *Neurocomputing*, vol. 122, pp. 258–265, 2013.
- [38] R. Ye and Q. Dai, "A novel greedy randomized dynamic ensemble selection algorithm," *Neural Processing Letters*, vol. 47, no. 2, pp. 565–599, 2018.
- [39] H.-j. Lu, C.-l. An, E.-h. Zheng, and Y. Lu, "Dissimilarity based ensemble of extreme learning machine for gene expression

- data classification,” *Neurocomputing*, vol. 128, pp. 22–30, 2014.
- [40] G. D. C. Cavalcanti, L. S. Oliveira, T. J. M. Moura, and G. V. Carvalho, “Combining diversity measures for ensemble pruning,” *Pattern Recognition Letters*, vol. 74, pp. 38–45, 2016.
- [41] S. Ding, Z. Chen, S.-y. Zhao, and T. Lin, “Pruning the ensemble of ANN based on decision tree induction,” *Neural Processing Letters*, vol. 48, no. 1, pp. 53–70, 2018.
- [42] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, “Extreme learning machine: theory and applications,” *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.
- [43] Y. Zhang, J. Wu, Z. Cai, P. Zhang, and L. Chen, “Memetic extreme learning machine,” *Pattern Recognition*, vol. 58, pp. 135–148, 2016.
- [44] K. N. Krishnanand and D. Ghose, “Glowworm swarm optimization: a new method for optimising multi-modal functions,” *International Journal of Computational Intelligence Studies*, vol. 1, no. 1, pp. 93–119, 2009.
- [45] K. N. Krishnan and D. Ghose, “Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions,” *Swarm Intelligence*, vol. 3, no. 2, pp. 87–124, 2009.
- [46] P. Xia, Z. Ni, X. Zhu, J. Zhang, and Y. Jin, “A novel prediction method based on improved binary glowworm swarm optimization and multi-fractal dimension for P2P Lending investment risk,” *IEEE Access*, vol. 8, pp. 23232–23245, 2020.
- [47] M. Li, X. Wang, Y. Gong, Y. Liu, and C. Jiang, “Binary glowworm swarm optimization for unit commitment,” *Journal of Modern Power Systems and Clean Energy*, vol. 2, no. 4, pp. 357–365, 2014.
- [48] Y. Q. Zhou, Z. X. Huang, and H. X. Liu, “Discrete glowworm swarm optimization algorithm for TSP problem,” *Acta Electronica Sinic*, vol. 40, no. 6, pp. 1164–1170, 2012.
- [49] S. Ding, Y. An, X. Zhang, F. Wu, and Y. Xue, “Wavelet twin support vector machines based on glowworm swarm optimization,” *Neurocomputing*, vol. 225, pp. 157–163, 2017.
- [50] M. A. K. Azad, A. M. A. C. Rocha, and E. M. G. P. Fernandes, “Improved binary artificial fish swarm algorithm for the 0-1 multidimensional knapsack problems,” *Swarm and Evolutionary Computation*, vol. 14, pp. 66–75, 2014.
- [51] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [52] Y. Cai, X. Liu, Y. Zhang, and Z. Cai, “Hierarchical ensemble of extreme learning machine,” *Pattern Recognition Letters*, vol. 116, pp. 101–106, 2018.
- [53] P. K. Singhal, R. Naresh, and V. Sharma, “Binary fish swarm algorithm for profit-based unit commitment problem in competitive electricity market with ramp rate constraints,” *IET Generation, Transmission & Distribution*, vol. 9, no. 13, pp. 1697–1707, 2015.
- [54] X.-Y. Luan, Z.-P. Li, and T.-Z. Liu, “A novel attribute reduction algorithm based on rough set and improved artificial fish swarm algorithm,” *Neurocomputing*, vol. 174, pp. 522–529, 2016.