



ARTICLE

Ensemble Classifier Design Based on Perturbation Binary Salp Swarm Algorithm for Classification

Xuhui Zhu^{1,3}, Pingfan Xia^{1,3}, Qizhi He^{2,4,*}, Zhiwei Ni^{1,3} and Liping Ni^{1,3}

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

²School of Statistics, Zhejiang Gongshang University, Hangzhou, 310018, China

³Key Laboratory of Process Optimization and Intelligent Decision-Making, Ministry of Education, Hefei, 230009, China

⁴Collaborative Innovation Center for Statistical Data Engineering Technology and Application, Zhejiang Gongshang University, Hangzhou, 310018, China

*Corresponding Author: Qizhi He. Email: happyhefei2000@163.com

Received: 03 April 2022 Accepted: 17 May 2022

ABSTRACT

Multiple classifier system exhibits strong classification capacity compared with single classifiers, but they require significant computational resources. Selective ensemble system aims to attain equivalent or better classification accuracy with fewer classifiers. However, current methods fail to identify precise solutions for constructing an ensemble classifier. In this study, we propose an ensemble classifier design technique based on the perturbation binary salp swarm algorithm (ECDPB). Considering that extreme learning machines (ELMs) have rapid learning rates and good generalization ability, they can serve as the basic classifier for creating multiple candidates while using fewer computational resources. Meanwhile, we introduce a combined diversity measure by taking the complementarity and accuracy of ELMs into account; it is used to identify the ELMs that have good diversity and low error. In addition, we propose an ECDPB with powerful optimizing ability; it is employed to find the optimal subset of ELMs. The selected ELMs can then be used to form an ensemble classifier. Experiments on 10 benchmark datasets have been conducted, and the results demonstrate that the proposed ECDPB delivers superior classification capacity when compared with alternative methods.

KEYWORDS

Ensemble classifier; salp swarm algorithm; diversity measure; multiple classifiers system; extreme learning machine

1 Introduction

Multiple classifier system (MCS) [1] is a popular field in machine learning. Compared with single classifiers, it provides potential enhancement in classification capability by integrating multiple base members [2]. MCS has been applied to handle variant tasks, including intrusion detection [3], expression recognition [4], image processing [5], and imbalanced learning [6]. In addition, some scholars attempt to solve the classification or regression problems in healthcare [7], well-being index [8], cancer detection [9], and population forecasting [10]. MCS mainly contains two steps: the first step is that an initial pool of classifiers with good diversity should be created, and the second one is



that the results of each classifier should be aggregated to produce the final decisions. Unfortunately, choosing an MCS with strong ability in actual applications involves tradeoffs [11] because it requires significant computing resources as the data size and the number of members increase [12]. Selective ensemble methods [13,14] aim at employing fewer classifiers while maintaining the classification capability of their ensemble to reduce the computing complexity. Considering their advantages, many scholars have studied how selective ensemble methods can be used to design well-performing ensemble classifiers [15].

To build an ensemble classifier with good classification ability, numerous methods have been developed. Different diversity measures, such as double-fault [13], Kappa [12], and the margin measure [16], can be used to assess each learner and to extract the members that satisfy certain conditions using different pruning methods. Finally, the constituents selected are integrated into an ensemble classifier. Meanwhile, many heuristic algorithms, such as genetic algorithm [17], glowworm swarm optimization [18], and artificial fish swarm algorithm [19], have been used to search for the final ensemble. Some scholars also utilize different means, such as graph coloring [20], simple coalitional games [15], greedy pruning [21], and clustering techniques [22,23], to identify the well-performing classifiers to form an ensemble. Moreover, the fusion of heuristic algorithms and diversity measures is a new approach to developing a selective ensemble of classifiers [16,19]. Diversity measures such as double-fault [19] and the margin measure [16] are used to select the initial classifiers, and heuristic algorithms, such as AFSA [16], are employed to select the final ensemble. The combination performs well when it comes to selecting classifiers with good performance within an ensemble [16,19].

In addition, the selective ensemble methods have a direct effect on the final ensemble classifier, and it is also critical to select the appropriate base classifier, which can be used to create multiple classifiers with good diversity [13,20]. Meanwhile, the base classifier should also have high computational efficiency to reduce the training times of multiple classifiers. Compared with traditional methods [24], extreme learning machines (ELMs) [25,26] are effective and simple learning methods. The random generation of the ELM's hidden nodes increases its learning speed [27,28]. Taking the advantages of ELM into consideration (high learning efficiency, conceptual simplicity, and good generalization capacity) [29,30], ELMs can be used as the base classifier for designing ensemble classifiers.

From this discussion, we find that fusion approaches work well when designing an ensemble classifier with good capability. However, the diversity measures used typically consider only the diversity or the precision of classifiers. This makes it difficult to capture the members with good diversity and low error to form the ensemble. If we only consider diversity, then the members with good diversity and low accuracy may be selected to construct the final ensemble, which may result in its low classification capability. If we only consider accuracy, then members with poor diversity and high precision may be used, which may lead to only slight improvement compared with single classifiers [16]. Additionally, the adopted heuristic algorithms cannot reach the expected convergence performance because of their shortcomings, such as the difficulty of implementation, a tendency to run into local optimums, and low global search capability. Since the salp swarm algorithm (SSA) [31–33] is simply constructed, has fewer parameters, and is more easily implemented, we employ a perturbation binary salp swarm algorithm (PBSSA) to enhance its searching ability. In this article, we propose a novel method called ensemble classifier design technique based on the perturbation binary salp swarm algorithm (ECDPB), which employs a combined diversity measure composed of complementarity and accuracy to yield a segment of candidates. Based on the preselected classifiers, we adopt the proposed PBSSA to search for better-performing classifiers in order to determine the final ensemble classifier.

The contributions of this work can be listed as follows:

- (1) We propose the ECDPB to design a well-performing ensemble classifier, by using a combined diversity measure and PBSSA.
- (2) The presented a combined diversity measure that simultaneously considers complementarity and accuracy in evaluating ELMs, and the proposed PBSSA delivers improved searching capacity compared to other algorithms.
- (3) Results of 10 benchmark datasets demonstrate that the proposed technique is superior to other related approaches.

The rest of this work is organized as below. In [Section 2](#), we introduce the combined diversity measure. [Section 3](#) proposes the ECDPB in detail. Comparison experiments are presented in [Section 4](#). Conclusions and future research directions are discussed in [Section 5](#).

2 Combined Diversity Measures

As we know, the diversity and the accuracy of ELMs are two fundamental determining factors for the classification capacity of an ensemble. However, there is no widely accepted measure of diversity. In addition, the method of designing a standard to extract the ELMs with good diversity and classification capacity remains an unresolved question [12,13]. We provide a novel combined diversity measure, which balances the diversity and the accuracy of ELMs. Therefore, the ELMs selected by the measure can be used to form an ensemble with high classification capacity in contrast to most current measures, which assess ELMs separately in terms of their complementarities or accuracies [34]. Instead, we evaluate the ELMs by simultaneously considering both these factors.

An ensemble with good predictive capability requires ELMs that complement each other [12,34]. In order to measure the ELM's complementarity with others, a complementarity measure is introduced, which can be used to select those ELMs that can correct the predictive errors obtained by other ELMs on the training samples. Meanwhile, the measure can make the selected ELMs perform diversely, which is an important prerequisite for attaining better classification capacity. If the ELMs provide identical output, it is then impossible to attain an improvement in classification by aggregating them [13].

Consider a series of samples, $X = \{(x_i, y_i), i = 1, 2, \dots, N\}$. Each sample is denoted by a feature vector x_i with its label y_i . For a set of ELMs $H = \{h_1, h_2, \dots, h_M\}$, the complementarity measure, Com_i of ELM h_i , can be calculated as follows:

$$Com_i = \frac{1}{M \times N} \sum_{j=1}^M \sum_{k=1}^N I(h_i(x_k) = y_k \cap h_j(x_k) \neq y_k) \quad (1)$$

where $I(\bullet)$ expresses an indicator function ($I(true) = 1$ and $I(false) = 0$), $h_i(x_k)$ denotes the predictive result of ELM h_i on the sample x_k .

Although the complementarity measure shows which ELM has higher complementarity with others, using it to select the ELMs can be problematic. The selection of ELMs using the measure can increase the diversity of ELMs, however, it may lead to a higher or lower error rate for an ensemble.

Selecting an ELM with a high complementarity measure may correct the predictive errors on some samples, however, it may produce flawed results in the remaining samples, particularly when the ELM has high complementarity with others, but its classification error is high. Therefore, the ensemble may generate incorrect decisions, which can result in the low precision of the final ensemble. Hence, we

should take into consideration not only the ELM's complementarity but also its accuracy. We define accuracy in this work, in terms of the classification error of the ELM. The accuracy measure of ELM h_i can be calculated as follows:

$$Acc_i = \frac{1}{N} \sum_{k=1}^N I(h_i(x_k) = y_k) \quad (2)$$

To choose an ensemble with high predictive ability, the ELMs in the ensemble should possess high levels of diversity and accuracy. When we attempt to increase the complementarity measures of the ELMs, the mean accuracies of all the ELMs may be reduced. However, when we try to increase the precision of the ELMs, their complementarity may be reduced. There is a trade-off between the complementarity measure and the accuracy measure of the ELMs. In order to find the optimal combination, we assign a set of weights for the complementarity measure and the accuracy measure, which we use to create the combined diversity measure. The combined diversity measure CDM_i of ELM h_i can be formulated as follows:

$$CDM_i = \alpha \times Com_i + \beta \times Acc_i \quad (3)$$

where α and β indicate the weights of the complementarity measure and the accuracy measure of the ELM h_i , respectively, and where $\alpha + \beta = 1$.

The combined diversity measure (CDM) can be used to assess the ELMs to reduce the probability of selecting ELMs with good diversity and low accuracy. Instead, those ELMs with higher complementarity measures and accuracy measures can be selected to form a final ensemble with good classification capacity. We can rank the ELMs using the combined diversity measure in Eq. (3). According to the rankings of the ELMs, we start the ensemble with the ELM that has the maximal measure value, and then select additional ELMs individually to construct the ensemble [34]. In the following analysis of the combined diversity measure, the weights (α, β) are set as (0.4, 0.6).

Fig. 1 presents the error trend curves gained by the combined diversity measures under different combinations of weights (α, β) with the sizes of 100, and 200 on different datasets. In Fig. 1, we can see that, as the sizes of the ELMs increase, the error curves under the weight combinations of (0.1, 0.9), (0.2, 0.8) and (0.4, 0.6) display lower error rates than the weight combinations of (0.6, 0.4), (0.8, 0.2) and (0.9, 0.1). Under the weight combinations of (0.1, 0.9), (0.2, 0.8) and (0.4, 0.6), the weight of the accuracy measure is greater than that of the complementarity measure, leading to the selection of ELMs with lower errors at the beginning. Thereby, they can perform better as the complementary ELMs are added into the ensemble. Under the weight combinations of (0.6, 0.4), (0.8, 0.2) and (0.9, 0.1), and the ELMs with high complementarity measures are selected to form the ensemble. However, they perform poorly because of low accuracy measures. We can clearly observe from Fig. 1 that, among the weight combinations of (0.1, 0.9), (0.2, 0.8) and (0.4, 0.6), the weight combination (0.4, 0.6) has significantly lower errors in classification. The ELMs selected by the combined weights (0.1, 0.9) and (0.2, 0.8) have higher precision, however, they lack complementarity.

According to the above analyses, it is easy to identify the tradeoff between the complementarity and the error of ELMs. The combined diversity measure with weights (0.4, 0.6) performs the best. Fig. 1 shows that the errors generated by this combined measure decline first and then rise as the ELMs' sizes increase, which shows that we use the combined measure to extract the well-performing ELMs. As such, the weights (α, β) of the combined diversity measure are set at (0.4, 0.6).

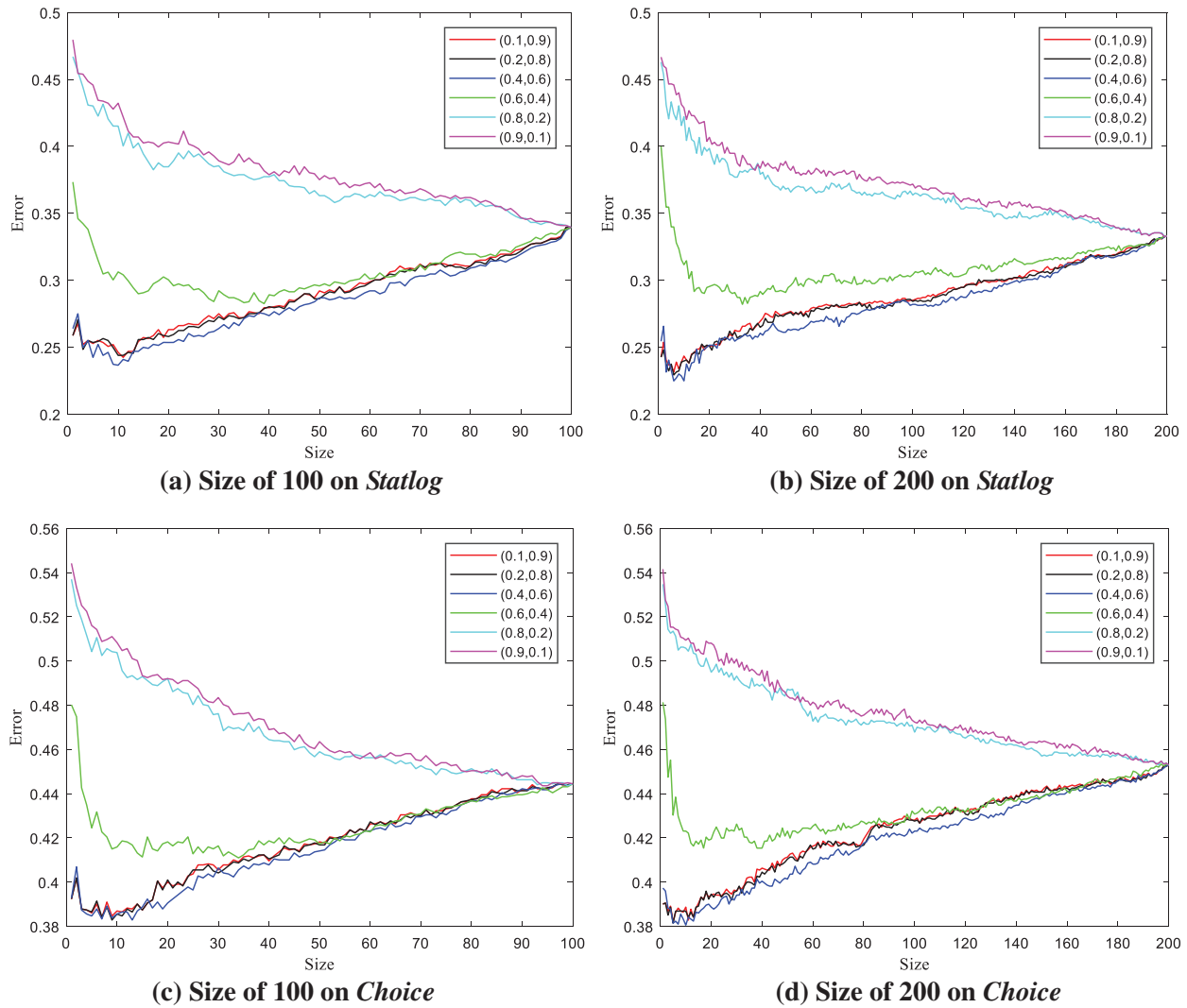


Figure 1: The error curves with different combinations of weights

To evaluate the performance of the combined diversity measure, we test its classification errors in comparison with the double-fault measure [13], kappa measure [12], complementarity measure [12], and margin measure [16]. Fig. 2 displays the error trend curves of these diversity measures. We compare the classification errors of the combined diversity measure with its weights of (0.4, 0.6) with other diversity measures. The combined diversity measure can achieve lower classification errors than others, because it evaluates ELMs by considering their accuracy and complementarity simultaneously, outperforming the alternative ones that only consider either one of them. In conclusion, the analysis demonstrates that the combined diversity measure produces good classification results.

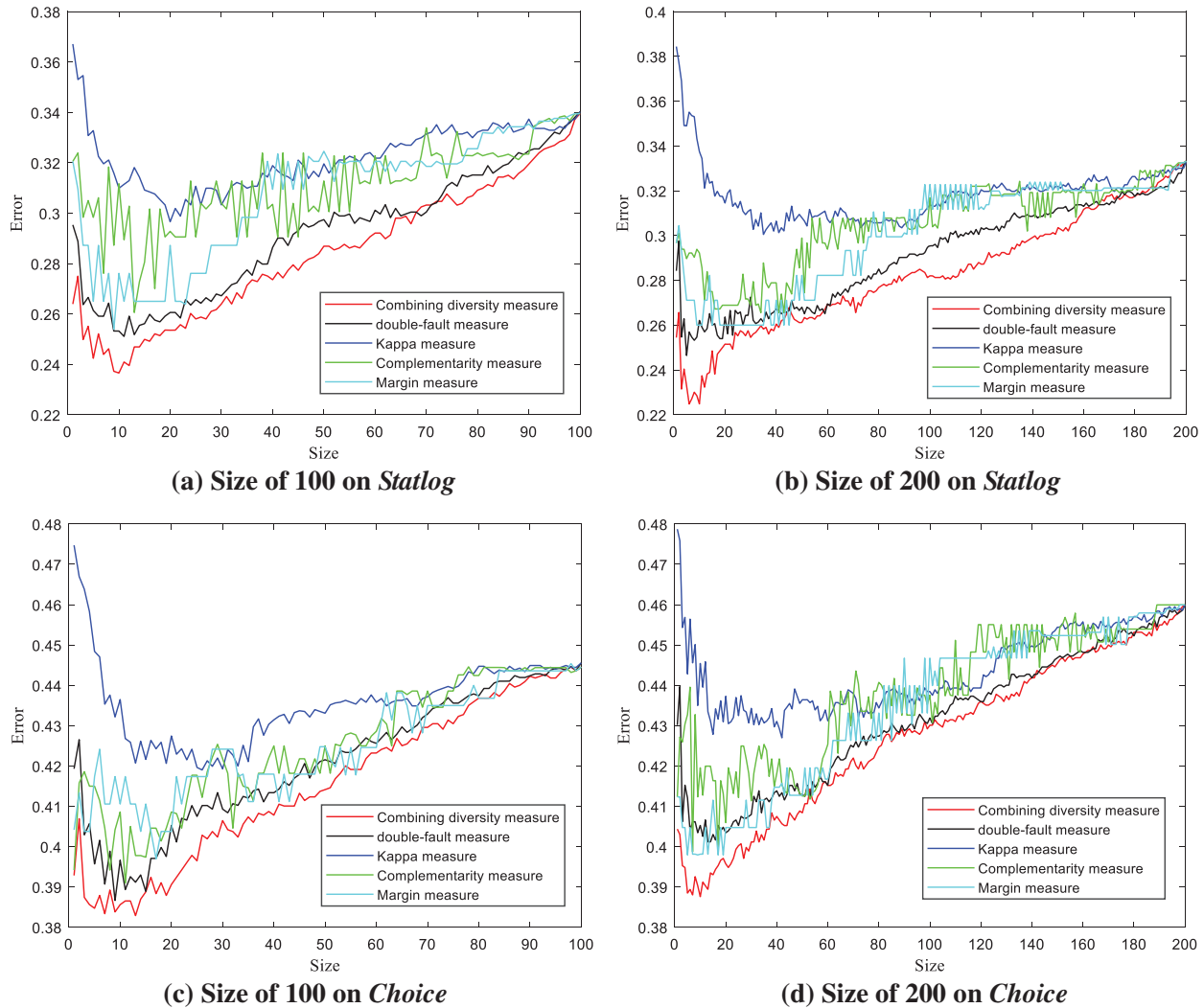


Figure 2: The error curves achieved by different diversity measures

3 Ensemble Classifier Design via the Perturbation Binary Salp Swarm Algorithm

In this section, we propose the ECDPB. First, we present the process used to generate the base ELMs. Second, we present the proposed PBSSA. Finally, we propose the ECDPB using the combined diversity measure and PBSSA. We also show how to employ the ECDPB to identify the final subset of ELMs to design a well-performing ensemble classifier. In addition, the pseudo-code of the ECDPB is presented.

3.1 Generating Multiple Extreme Learning Machines

In this section, we design an ensemble classifier with good capacity and employ ECDPB to select the ELMs for the ensemble. That is, we extract the ELMs with good complementarity and low error for inclusion in the final ensemble classifier. Before this, we need to generate a pool of ELMs with good diversity [20] in order to select a collection of well-performing ELMs. ELMs have fast learning

rates and strong generalization ability; we make complete use of these advantages in creating multiple classifiers that work quickly. To increase the diversity of the generated ELMs, we utilize bootstrap extraction [35] to select a part of the samples in the training set, which can be used as the input of ELM; this produces the trained ELM. We utilize bootstrap extraction [35] with M iterations to extract the training set, generating M subsets. Then, multiple ELMs can be obtained by training the model of the ELM on each subset. Therefore, multiple ELMs with good diversity are created.

3.2 Perturbation Binary Salp Swarm Algorithm

In this section, we join the PBSSA and the combined diversity measure to select a fraction of well-performing ELMs in order to design an ensemble classifier. We first introduce the SSA. Mirjalili et al. [31] first proposed the SSA, which was inspired by the salps' swarming behavior in the ocean. Salps consume marine phytoplankton; they move to other places by inhaling or exhaling seawater. And they usually gather to form a chain called a salp chain. The chain behavior can be adopted to forage food. In the SSA, the salp chain is composed of two kinds of salps: the leader and the follower. The updating process of the leader salp can be formulated as shown in Eqs. (4) and (5).

$$x_j^l = \begin{cases} F_j + c_1 ((ub_j - lb_j) c_2 + lb_j), & c_3 \geq 0.5 \\ F_j - c_1 ((ub_j - lb_j) c_2 + lb_j), & c_3 < 0.5 \end{cases} \quad (4)$$

$$c_1 = 2e^{-\left(\frac{4t}{T_{\max}}\right)^m} \quad (5)$$

where x_j^l shows the j th dimension of the leader; F_j indicates the j th dimension of food source; ub_j and lb_j represent the upper and lower bound of the j th dimension space, respectively; c_2 and c_3 denote random numbers between 0 and 1, respectively; m represents the power factor; t denotes the current iteration; T_{\max} denotes the maximal iteration; and c_1 expresses the parameter that balances the exploring and developing capabilities of SSA.

The followers' position can be updated utilizing the Eq. (9), denoting their movement.

$$x_j^i = \frac{1}{2} (x_j^i + x_j^{i-1}) \quad (6)$$

where x_j^i ($i \geq 2$) declares the i th follower (because the first salp is the leader). After the population initialization of salps, the salps can seek the optimal solution in terms of Eqs. (4) and (6).

The selection of ELMs is a combinatorial optimization problem, which the basic SSA cannot directly solve. Hence, we need to improve the motion pattern of the salps, such that they can search for the optimal solution in the discrete solution space. The enhancements of SSA are discussed next.

3.2.1 Improvement of the Search Process

The motion pattern of salps in the SSA cannot match the movements of individuals in a discrete solution space; this means that the motion of the leader and the follower should be changed. Considering the characteristics of the problem to be solved and that the motion pattern should be simple and effective, we change the original updating process of the leader and the follower in SSA. The updating processes of the leader and the follower are presented in the following equations.

For leaders, if $c_3 \geq 0.5$, then we update their movement according to Eq. (7), otherwise, we use Eq. (8).

$$x_j^l = \begin{cases} 1, & \text{if } F_j + c_1 c_2 > 0 \\ 0, & \text{if } F_j + c_1 c_2 \leq 0 \end{cases} \quad (7)$$

$$x_j^1 = \begin{cases} 1, & \text{if } F_j - c_1 c_2 > 0 \\ 0, & \text{if } F_j - c_1 c_2 \leq 0 \end{cases} \quad (8)$$

For followers, we update their movement using Eq. (9).

$$x_j^i = \begin{cases} 1, & \text{if } \frac{1}{2} (x_j^i + x_j^{i-1}) > 0.5 \\ r, & \text{if } \frac{1}{2} (x_j^i + x_j^{i-1}) = 0.5 \\ 0, & \text{if } \frac{1}{2} (x_j^i + x_j^{i-1}) < 0.5 \end{cases} \quad (9)$$

where r points to a random number, which is either 0 or 1.

3.2.2 Perturbation Mechanism

In the SSA, the individuals usually gather to construct the salp chain, and the population may lack diversity. Hence, we attempt to improve the solution diversity in order to enhance the searching efficiency. A perturbation mechanism [36] is introduced to the SSA, which can optimize the tradeoff between exploitation and exploration in the searching process. A predefined perturbation factor *perfactor* is introduced, and the perturbation mechanism is presented as follows:

$$\text{perfactor} = \frac{1}{\pi} \arctan(t/T_{\max}) + \frac{1}{2} e^{-coh} \quad (10)$$

$$coh = \frac{1}{n} \sum_{i=1}^n \left\| \frac{x^i - \bar{x}}{\bar{x}} \right\| \quad (11)$$

$$x^i = x^i + [\text{perfactor} \times \text{step}], \text{ if } \text{rand} < r_p \quad (12)$$

where *coh* signifies the cohesion index of the salp population; x^i denotes the i th individual; \bar{x} shows the center position of all of the salps; r_p indicates a predefined perturbation probability; and *step* denotes a perturbation step. When *coh* is smaller, it suggests that the cohesion index is larger and the individuals flock together, which requires a large perturbation; otherwise, it indicates that the cohesion index is smaller and the individuals are dispersed, which may require only a small perturbation.

In the early stage of the SSA's search, the individuals are scattered in the solution space, and the cohesion index is low, which produces a small perturbation. In the later stage of the search, most of the individuals have gathered, and the cohesion index is high, generating a large perturbation. This process ensures that the population performs diversely.

3.2.3 Gauss Mutation Operation

To avoid having the salp individuals caught in local optimum, we introduce a Gauss mutation operation [37,38], such that, some of the individuals break away from the local optimum with a large probability. After each iteration, we select 20% of the individuals with the worst performance to carry out the Gauss mutation operation. The operation can be described as follows:

$$x^i = x^i + [r_g \times \text{step}] \quad (13)$$

where r_g denotes a random number from the Gaussian distribution $N(1, 1)$.

3.3 Selection Process of Extreme Learning Machines

In this section, we utilize the ECDPB to select the ELMs for constructing the ensemble classifier. This process involves four main steps. First, the combined diversity measure of each ELM is calculated using Eq. (3). Second, we rank all the ELMs in terms of their combined measures to obtain their new ordering. Third, we retain the first M' ($M' \leq M$) ELMs with the largest combined diversity measures; M' is a parameter of the ECDPB, and we analyze it in Section 4.3. Finally, we adopt the PBSSA to choose well-performing ELMs from the remaining M' options. Thereby, the proposed ECDPB can extract the ELMs with good diversity and low error to constitute an ensemble classifier with good classification capacity.

The selection of ELMs is accomplished using the ECDPB, and the classification error of the ensemble classifier is taken as the objective function in this study. The ensemble process of multiple ELMs is calculated as follows:

$$E_H(x) = \arg \max \sum_{i=1}^m I(h_i(x) = y), y \in C \quad (14)$$

where h_i shows the i th ELM; (x, y) denotes a sample; and C denotes the class label set. In addition, we present the pseudo-code of the ECDPB as follows:

Algorithm ECDPB

Inputs: Training set and test set.

Outputs: An ensemble classifier, and its error.

- 1: Multiple ELMs are created utilizing bootstrap extraction for constructing $H \leftarrow \{h_1, h_2, \dots, h_M\}$.
- 2: The combined diversity measure, CDM_i , of each ELM, h_i , can be calculated utilizing Eq. (3).
- 3: The new sequence $CDM' = (CDM'_1, CDM'_2, \dots, CDM'_M)$ is attained by ranking that measure in descending order.
- 4: The ELMs' sequence h'_1, h'_2, \dots, h'_m corresponds to the new sequence, CDM' .
- 5: $H' \leftarrow \emptyset$.
- 6: **for** $j \leftarrow 1$ to M' **do**
- 7: $H' \leftarrow H' \cup \{h'_j\}$.
- 8: **end for**
- 9: The population initialization of salps is achieved.
- 10: $t \leftarrow 1$.
- 11: **while** $t \leq T_{\max}$ **do**
- 12: Calculate the classification error of each salp.
- 13: Rank the salp individuals in terms of their classification errors.
- 14: Choose the half of the salps with the lowest errors as the leaders and the rest as followers.
- 15: **for** $i \leftarrow 1$ to N **do**
- 16: **if** $i \leq n/2$ **do**
- 17: Execute leader behavior via Eq. (7) or Eq. (8).
- 18: **else**
- 19: Execute follower behavior using Eq. (9).
- 20: **end if**
- 21: **if** $rand < r_p$

(Continued)

Algorithm (Continued)

```

22:   Introduce perturbation mechanism.
23:   end if
24:   if  $i > 4n/5$ 
25:     Perform the Gauss mutation operation.
26:   end if
27: end for
28: end while
29: return outputs.

```

3.4 Complexity Analysis of ECDPB

In this section, we analyze the time complexity of the proposed ECDPB. Assume that the number of PBSSA's population is N , the maximal number of iterations is T_{\max} , the number of ELMs in the initial pool is M , and the dimension of salp is M' . Firstly, we use combined diversity measure to select M' ELMs, and its complexity is $O(M \log_2 M)$. Secondly, the time complexity of salp population initialization is $O(N \times M')$, and the searching process's complexity is $O(N^2)$. Finally, the overall complexity of ECDPB is $O(T_{\max} \times N^2)$ after T_{\max} iterations.

4 Experiments

To evaluate the ECDPB's utility, we employ 10 benchmark tasks as the empirical dataset, as listed in Table 1. In this work, to reduce the randomness of experimental results, each experiment was repeated 30 times. The experiments were carried out in Matlab 2020a on a computer running 64-bit Windows 10 with 3.6 GHz processor i7-9700K and 32 GB memory. Each dataset is randomly divided into five equal parts. Three are used for training; one is used for verification; and one is used for testing.

Table 1: Benchmark datasets

| Datasets | Instances | Attributes | Classes |
|------------|-----------|------------|---------|
| Sonar | 208 | 60 | 2 |
| Statlog | 270 | 13 | 2 |
| Energy | 768 | 8 | 5 |
| Vowel | 871 | 3 | 6 |
| Flare | 1066 | 11 | 6 |
| Website | 1353 | 9 | 3 |
| Choice | 1473 | 9 | 3 |
| Navigation | 5456 | 24 | 4 |
| Satimage | 6435 | 36 | 7 |
| Marketing | 6876 | 13 | 9 |

4.1 Experimental Results

The ECDPB is composed of two parts: the CMD and the PBSSA. We attempt to determine whether the ECDPB (the fusion of the CMD and the PBSSA) can attain lower errors than the CMD

and the PBSSA. Tables 2 and 3 illustrate the classification errors produced by the ECDPB and those obtained by the CDM and the PBSSA. The Wilcoxon test [20] is applied in this study to verify whether the differences between the ECDPB and others are significant (its significance level is usually set to 0.05). When the p -value is lower than 0.05, we should reject the null hypothesis, and the difference in results between the two methods can be regarded as significant. As can be seen in Tables 2 and 3, the errors and standard deviations of the ECDPB are lower than the CDM and PBSSA, and it indicates that the ECDPB achieves obvious improvements in classification over the two alternatives. Additionally, the results in Tables 2 and 3 include p -values that are under 0.05. The results in Tables 2 and 3 indicate that the ECDPB utilizes fewer ELMs than the CMD and PBSSA and produces fewer errors. Meanwhile, the ECDPB employs fewer ELMs in constructing an ensemble classifier. Therefore, we can conclude that the ECDPB is an effective technique for designing an ensemble classifier with greater classification ability.

Table 2: Classification errors and standard deviation of the ECDPB in comparison with the CDM and PBSSA with 100 ELMs

| Datasets | ECDPB | Size | CDM | Size | ECDPB vs. CDM | PBSSA | Size | ECDPB vs. PBSSA |
|------------|------------------------|------|-----------------|------|-------------------|-----------------|------|-------------------|
| Sonar | 0.0689 ± 0.0320 | 16 | 0.1181 ± 0.0374 | 35 | 1.73e − 06 | 0.1392 ± 0.0387 | 54 | 1.73e − 06 |
| Statlog | 0.2290 ± 0.0361 | 16 | 0.2823 ± 0.0400 | 35 | 1.71e − 06 | 0.2920 ± 0.0384 | 66 | 1.72e − 06 |
| Energy | 0.1684 ± 0.0262 | 15 | 0.1914 ± 0.0251 | 35 | 2.52e − 06 | 0.1922 ± 0.0321 | 67 | 8.25e − 06 |
| Vowel | 0.1538 ± 0.0272 | 15 | 0.1920 ± 0.0292 | 35 | 1.73e − 06 | 0.2083 ± 0.0307 | 50 | 1.73e − 06 |
| Flare | 0.2218 ± 0.0309 | 15 | 0.2456 ± 0.0347 | 35 | 1.69e − 06 | 0.2490 ± 0.0347 | 53 | 1.68e − 06 |
| Website | 0.1085 ± 0.0288 | 14 | 0.1316 ± 0.0320 | 35 | 2.49e − 06 | 0.1392 ± 0.0283 | 54 | 1.69e − 06 |
| Choice | 0.3791 ± 0.0348 | 15 | 0.4171 ± 0.0337 | 35 | 1.68e − 06 | 0.4252 ± 0.0338 | 66 | 1.69e − 06 |
| Navigation | 0.1653 ± 0.0169 | 16 | 0.1843 ± 0.0182 | 35 | 1.73e − 06 | 0.1862 ± 0.0178 | 60 | 1.73e − 06 |
| Satimage | 0.3770 ± 0.0162 | 18 | 0.3936 ± 0.0172 | 35 | 1.73e − 06 | 0.3987 ± 0.0185 | 63 | 1.73e − 06 |
| Marketing | 0.6237 ± 0.0215 | 16 | 0.6414 ± 0.0212 | 35 | 1.73e − 06 | 0.6449 ± 0.0201 | 62 | 1.73e − 06 |

Note: Bold indicates that the results gained by ECDPB perform better than others.

Table 3: Classification errors and standard deviation of the ECDPB in comparison with the CDM and PBSSA with 200 ELMs

| Datasets | ECDPB | Size | CDM | Size | ECDPB vs. CDM | PBSSA | Size | ECDPB vs. PBSSA |
|------------|------------------------|------|-----------------|------|-------------------|-----------------|------|-------------------|
| Sonar | 0.0542 ± 0.0296 | 15 | 0.0958 ± 0.0345 | 35 | 1.73e − 06 | 0.1495 ± 0.0431 | 113 | 1.73e − 06 |
| Statlog | 0.2123 ± 0.0347 | 17 | 0.2559 ± 0.0401 | 35 | 1.69e − 06 | 0.2954 ± 0.0409 | 109 | 1.71e − 06 |
| Energy | 0.1690 ± 0.0213 | 13 | 0.1833 ± 0.0204 | 35 | 8.23e − 06 | 0.1896 ± 0.0234 | 117 | 5.53e − 06 |
| Vowel | 0.1441 ± 0.0282 | 17 | 0.1717 ± 0.0288 | 35 | 1.73e − 06 | 0.2120 ± 0.0370 | 110 | 1.73e − 06 |
| Flare | 0.2078 ± 0.0299 | 15 | 0.2303 ± 0.0332 | 35 | 2.51e − 06 | 0.2559 ± 0.0347 | 105 | 1.70e − 06 |
| Website | 0.1030 ± 0.0270 | 14 | 0.1207 ± 0.0295 | 35 | 3.56e − 06 | 0.1416 ± 0.0299 | 115 | 1.70e − 06 |
| Choice | 0.3632 ± 0.0338 | 16 | 0.4033 ± 0.0377 | 35 | 1.69e − 06 | 0.4265 ± 0.0391 | 112 | 1.69e − 06 |
| Navigation | 0.1568 ± 0.0175 | 19 | 0.1743 ± 0.0189 | 35 | 1.73e − 06 | 0.1910 ± 0.0196 | 126 | 1.73e − 06 |
| Satimage | 0.3662 ± 0.0195 | 16 | 0.3798 ± 0.0186 | 35 | 1.73e − 06 | 0.3988 ± 0.0205 | 110 | 1.73e − 06 |
| Marketing | 0.6265 ± 0.0210 | 16 | 0.6442 ± 0.0239 | 35 | 1.72e − 06 | 0.6555 ± 0.0228 | 109 | 1.73e − 06 |

Note: Bold indicates the results gained by ECDPB perform better than others.

It also can be seen from Tables 2 and 3 that, overall, over 75 percent of the ELMs can be eliminated employing the ECDPB, meaning that we can utilize fewer ELMs to design a better-performing ensemble classifier and save significant computational resources in practical applications. Regarding the selection of ELMs, as the size of an ELM increases, its computational complexity increases

exponentially [12,13]. In comparison with the CDM and PBSSA using 100 and 200 ELMs, the ECDPB has the capability to produce fewer errors with ELMs of different sizes. In addition, the errors can be reduced as the number of ELMs increases. When we employ 200 ELMs to design an ensemble classifier, the ECDPB achieves only a slight enhancement in classification. As a result, the number of ELMs should be set at 100.

To verify the utility of the ECDPB, we compare the proposed method with the following techniques: GASEN [17], DMEP [20], MDOEP [39], IBAFSEN [19], PEAD [40], RCOA [34], and IDAFMEP [16]. The primary goal of these techniques is to enhance the classification capacity in comparison with a bagging ensemble. The implementations of these approaches are based on their corresponding literatures. In contrast to IBAFSEN and IDAFMEP, the proposed ECDPB yields the combined diversity measure and PBSSA by optimizing them overall, and the combined diversity measure takes the diversity and accuracy of classifiers into consideration. IBAFSEN and IDAFMEP use a double-fault or margin measure and AFSA to select the final ensemble by optimizing them separately. Double-fault and margin measures consider only either diversity or accuracy at a time. In addition, the proposed PBSSA has more powerful optimization ability compared to AFSA, which is shown in Section 4.2.

We present the comparisons of classification errors and standard deviations between the proposed method and the alternatives with 100 ELMs in Table 4. The results in Table 4 reveal that the proposed ECDPB produces lower errors and standard deviations than the other methods, though it does produce higher errors than IDAFMEP on *Website* datasets. Moreover, the proposed technique does not obtain the highest errors on any of the 10 datasets. Overall, the ECDPB yields lower errors than its competitors with good stability. Meanwhile, IDAFMEP can also obtain lower errors than other methods. The sizes of the ELMs used by different approaches to design an ensemble classifier have been displayed in Table 5. We find that the ECDPB uses smaller ELMs than most methods for designing an ensemble classifier, however, it still uses larger ELMs than DMEP and IDAFMEP. Although the proposed ECDPB does not use the smallest ELMs, it does produce lower classification errors.

In addition, we use the Wilcoxon test [20] to verify the significance of differences between the proposed ECDPB and its competitors for each of the datasets, as shown in Table 6, where “+/-” indicates the number of datasets that there exist significant differences between the proposed ECDPB and other methods or not. The results in Table 6 suggest that most of the values are lower than 0.05, which by and large implies that there are significant differences between the proposed method and the comparison methods. Therefore, we can conclude that the proposed ECDPB provides superior classification compared with other techniques, and it is an effective approach for designing a better-performing ensemble classifier. Meanwhile, the proposed ECDPB can deliver better classification capacity because it employs the combined diversity measure to evaluate each ELM by simultaneously considering diversity and accuracy. It can choose a collection of better-performing ELMs than alternative measures that consider only one of the two factors. Furthermore, the proposed ECDPB also uses PBSSA with powerful optimizing ability to search for a more competitive subset of ELMs, which can provide an ensemble classifier with superior performance. Therefore, the proposed ECDPB is a reasonable and effective technique for designing an ensemble classifier.

Table 4: Classification errors and standard deviation gained by ECDPB and other techniques with 100 ELMs

| Datasets | ECDPB | GASEN | DMEP | MDOEP | IBAFSEN | PEAD | RCOA | IDAFMEP |
|------------|------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|------------------------|
| Sonar | 0.0689 ± 0.0320 | 0.1166 ± 0.0383 | 0.1066 ± 0.0473 | 0.1054 ± 0.0484 | 0.0858 ± 0.0270 | 0.1079 ± 0.0562 | 0.1030 ± 0.0405 | 0.0715 ± 0.0244 |
| Statlog | 0.2290 ± 0.0361 | 0.2551 ± 0.0364 | 0.2465 ± 0.0512 | 0.2514 ± 0.0544 | 0.2379 ± 0.0436 | 0.2460 ± 0.0408 | 0.2569 ± 0.0483 | 0.2305 ± 0.0452 |
| Energy | 0.1684 ± 0.0262 | 0.1787 ± 0.0223 | 0.1900 ± 0.0476 | 0.1822 ± 0.0393 | 0.1715 ± 0.0233 | 0.1854 ± 0.0364 | 0.1814 ± 0.0322 | 0.1711 ± 0.0227 |
| Vowel | 0.1538 ± 0.0272 | 0.1856 ± 0.0293 | 0.1802 ± 0.0394 | 0.1824 ± 0.0482 | 0.1655 ± 0.0261 | 0.1736 ± 0.0377 | 0.1954 ± 0.0276 | 0.1622 ± 0.0232 |
| Flare | 0.2218 ± 0.0309 | 0.2450 ± 0.0430 | 0.2393 ± 0.0426 | 0.2408 ± 0.0368 | 0.2357 ± 0.0434 | 0.2465 ± 0.0470 | 0.2469 ± 0.0409 | 0.2241 ± 0.0429 |
| Website | 0.1085 ± 0.0288 | 0.1214 ± 0.0299 | 0.1269 ± 0.0243 | 0.1309 ± 0.0217 | 0.1208 ± 0.0269 | 0.1363 ± 0.0297 | 0.1160 ± 0.0313 | 0.1062 ± 0.0261 |
| Choice | 0.3791 ± 0.0348 | 0.3927 ± 0.0354 | 0.3806 ± 0.0546 | 0.4099 ± 0.0548 | 0.3997 ± 0.0379 | 0.3949 ± 0.0335 | 0.3963 ± 0.0369 | 0.3864 ± 0.0375 |
| Navigation | 0.1653 ± 0.0169 | 0.1809 ± 0.0156 | 0.1831 ± 0.0217 | 0.1832 ± 0.0191 | 0.1715 ± 0.0177 | 0.1797 ± 0.0249 | 0.1840 ± 0.0163 | 0.1696 ± 0.0198 |
| Satimage | 0.3770 ± 0.0162 | 0.3904 ± 0.0163 | 0.3841 ± 0.0242 | 0.3935 ± 0.0249 | 0.3662 ± 0.0189 | 0.3960 ± 0.0148 | 0.3916 ± 0.0236 | 0.3872 ± 0.0176 |
| Marketing | 0.6237 ± 0.0215 | 0.6434 ± 0.0200 | 0.6440 ± 0.0263 | 0.6533 ± 0.0254 | 0.6423 ± 0.0285 | 0.6431 ± 0.0246 | 0.6427 ± 0.0220 | 0.6355 ± 0.0303 |

Table 5: Sizes of the ensembles produced by comparison techniques with 100 ELMs

| Datasets | ECDPB | GASEN | DMEP | MDOEP | IBAFSEN | PEAD | RCOA | IDAFMEP |
|------------|-------|-------|-----------|-------|---------|------|------|---------|
| Sonar | 16 | 42 | 7 | 30 | 23 | 29 | 21 | 13 |
| Statlog | 16 | 35 | 6 | 30 | 21 | 42 | 29 | 10 |
| Energy | 15 | 42 | 3 | 30 | 19 | 55 | 26 | 6 |
| Vowel | 15 | 38 | 9 | 30 | 18 | 28 | 29 | 12 |
| Flare | 15 | 39 | 5 | 30 | 21 | 32 | 25 | 13 |
| Website | 14 | 27 | 4 | 30 | 22 | 27 | 26 | 7 |
| Choice | 15 | 34 | 7 | 30 | 23 | 29 | 23 | 12 |
| Navigation | 16 | 45 | 11 | 30 | 23 | 26 | 23 | 13 |
| Satimage | 18 | 44 | 7 | 30 | 21 | 25 | 27 | 15 |
| Marketing | 16 | 23 | 9 | 30 | 23 | 31 | 23 | 11 |

Table 6: Results of the Wilcoxon test comparing the ECDPB with other methods

| Datasets | GASEN | DMEP | MDOEP | IBAFSEN | PEAD | RCOA | IDAFMEP |
|------------|-----------------|----------|----------|----------|----------|-----------------|-----------------|
| Sonar | 5.22e-06 | 3.59e-04 | 3.45e-04 | 2.26e-03 | 5.22e-06 | 2.35e-06 | 4.28e-02 |
| Statlog | 8.92e-05 | 6.64e-04 | 6.62e-05 | 5.32e-03 | 5.22e-06 | 5.75e-06 | 3.71e-02 |
| Energy | 5.26e-02 | 1.72e-03 | 9.27e-03 | 2.30e-02 | 7.71e-04 | 1.60e-04 | 4.20e-02 |
| Vowel | 3.41e-05 | 1.25e-04 | 1.48e-04 | 1.74e-04 | 3.52e-06 | 2.88e-06 | 1.11e-02 |
| Flare | 1.75e-02 | 6.73e-03 | 2.10e-03 | 1.06e-02 | 5.32e-03 | 2.11e-03 | 4.91e-02 |
| Website | 4.22e-02 | 1.01e-02 | 2.11e-03 | 2.45e-02 | 1.97e-05 | 5.10e-02 | 7.50e-01 |
| Choice | 1.71e-03 | 1.57e-02 | 1.45e-02 | 2.83e-02 | 9.31e-06 | 3.40e-05 | 4.71e-02 |
| Navigation | 2.11e-03 | 4.53e-04 | 4.20e-04 | 1.48e-02 | 5.79e-05 | 1.92e-06 | 1.78e-02 |
| Satimage | 1.32e-02 | 1.71e-03 | 9.84e-03 | 8.31e-03 | 1.25e-04 | 1.97e-05 | 4.28e-02 |
| Marketing | 2.42e-03 | 1.36e-04 | 2.61e-04 | 4.05e-02 | 1.02e-05 | 1.36e-04 | 4.53e-02 |
| +/- | 9/1 | 10/0 | 10/0 | 10/0 | 10/0 | 9/1 | 9/1 |

Note: Bold expresses there is not significant difference between the proposed ECDPB and other methods.

4.2 Parameters Sensitivity

In this study, we employ the proposed ECDPB to design an ensemble classifier, and its parameters have great effects on its performance. Hence, we need to explore the effects of the ECDPB's parameters: the maximal iterations T_{\max} , the retained size M' , population size N , the perturbation probability r_p , and the power factor m . Due to space constraints, we carry out the experiments on the *Energy* dataset utilizing 100 ELMs. The sensitivity analysis of the parameters is reported next.

The ECDPB uses the PBSSA proposed in Section 3.2 as its component. We need to analyze the influence of the number of iterations on the performance of the PBSSA, which is shown in Fig. 3. Fig. 3 indicates that the classification errors of the PBSSA initially decrease as the number of iterations increases, flattening out afterward. Moreover, we assess the search performance of the proposed PBSSA in comparison with other heuristic algorithms. The comparison algorithms include IDAFS [16], IBAFS [19], BAFS [41], IAFS [42], DGSO [43], BGSO [44], MPSO [45], BPSO [46], and BGA [17].

The trends in Fig. 3 suggest that the proposed PBAFSA produces fewer errors than other methods, and it has a more powerful optimizing ability and higher optimizing precision. The results also show that the different strategies we employ to modify the basic SSA are reasonable and effective. We explore the effect of the retained size of ELMs produced by the CDM on the results of the ECDPB, as shown in Fig. 4a. Fig. 4b demonstrates the impacts of population size on the classification errors. We show the relationship between the perturbation probability and performance of PBSSA in Fig. 4c. The effect of the power factor on the results of the PBSSA is exhibited in Fig. 4d. With these figures, we can identify the best choice for each parameter. In addition, similar observations on other datasets can be made. Therefore, we set $T_{\max} = 600$, $M' = 35$, $N = 35$, $r_p = 0.6$, $m = 3.5$ for the experiments in this study.

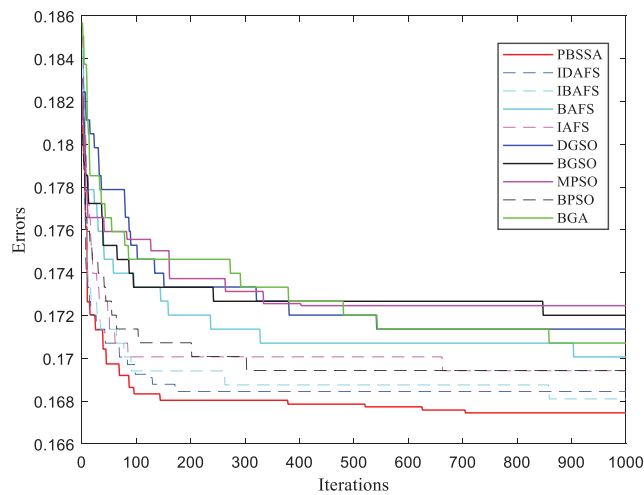
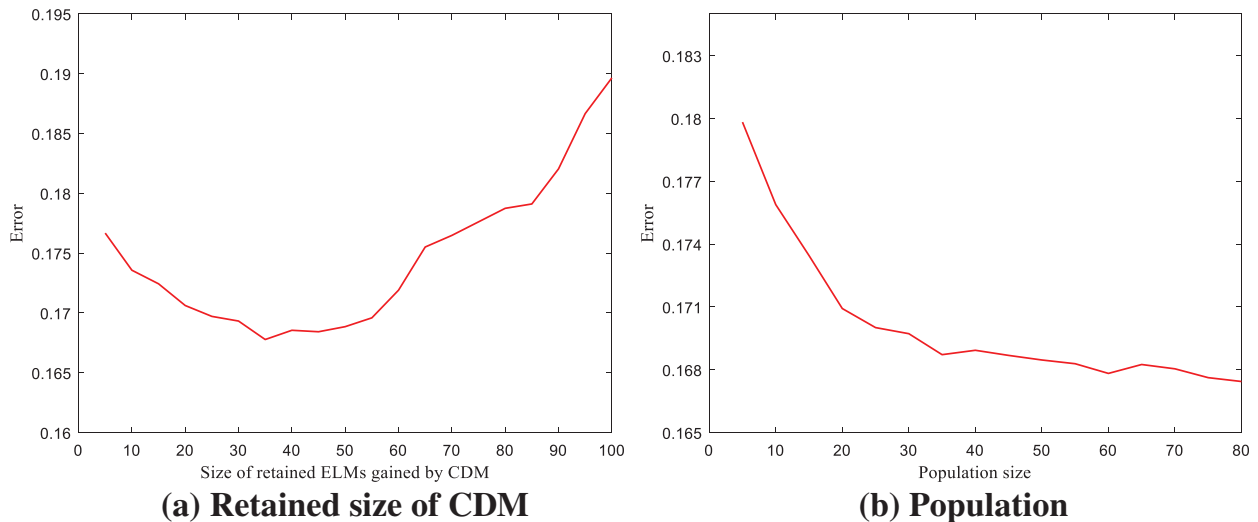


Figure 3: Effects of iteration on errors obtained by different algorithms



(a) Retained size of CDM

(b) Population

Figure 4: (Continued)

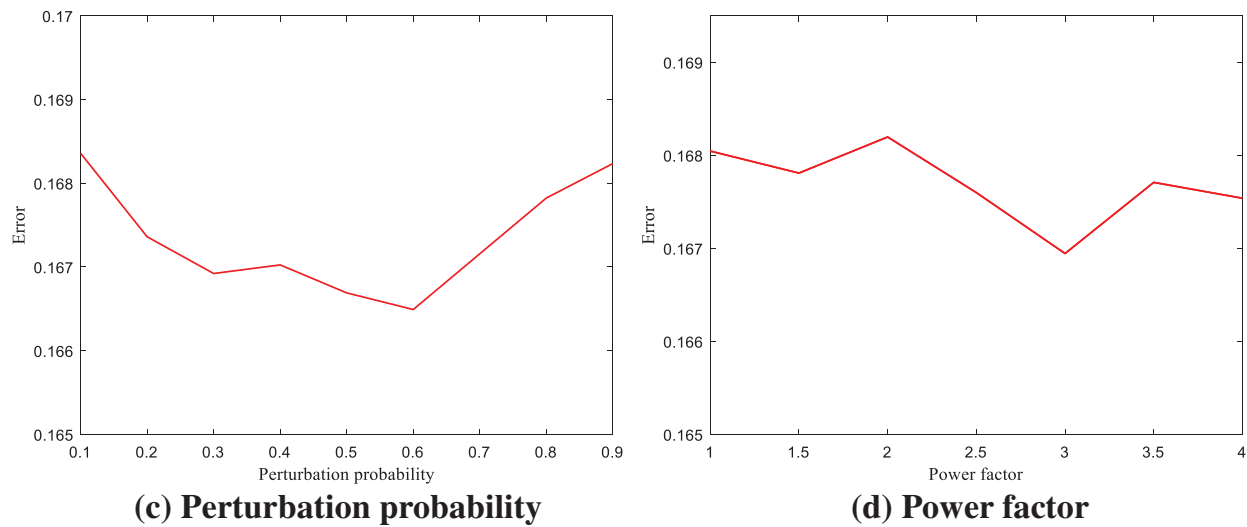


Figure 4: Effects of different parameters on errors of ECDPB

5 Conclusions

We proposed the ECDPB as a powerful means for designing a well-performing ensemble classifier. The proposed ECDPB unifies the combined diversity measure, the high-performance PBSSA, and the efficient ELM to form an ensemble classifier. The combined diversity measure can optimize the selection of the ELMs to account for the tradeoff between diversity and precision, and it enables those ELMs with good performance to be selected. The high-performance PBSSA selects the ELMs rapidly and accurately, which allows the ensemble classifier to attain good classification results while utilizing smaller ELMs. The efficient ELM can create a base classifier utilizing very few computing resources, and it is critical for minimizing resource consumption when generating multiple ELMs. The aforementioned methods allow for improvements in classification.

We have assessed the classification capacity of the ECDPB in 10 benchmark datasets, and the results imply that, compared with other algorithms, the proposed ECDPB can deliver superior performance in classification with smaller ELM sizes. In summary, the proposed ECDPB is an effective way to design an ensemble classifier. In the future, we attempt to apply the proposed technique in the stock market [47], compared with its feature extraction using the deep learning method, and the enhancements in prediction precisions of stock price indices can be made to provide support for stock investments.

Funding Statement: This work was supported in part by the Anhui Provincial Natural Science Foundation [1908085QG298, 1908085MG232], the National Nature Science Foundation of China [91546108, 61806068], the National Social Science Foundation of China [21BTJ002], the Anhui Provincial Science and Technology Major Projects Grant [201903a05020020], the Fundamental Research Funds for the Central Universities [JZ2019HGTA0053, JZ2019HGBZ0128], the Humanities and Social Science Fund of Ministry of Education of China [20YJA790021], the Major Project of Philosophy and Social Science Planning of Zhejiang Province [22YJRC07ZD], and the Open Research Fund Program of Key Laboratory of Process Optimization and Intelligent Decision-Making (Hefei University of Technology), Ministry of Education.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Cruz, R. M., Sabourin, R., Cavalcanti, G. D. (2018). Dynamic classifier selection: Recent advances and perspectives. *Information Fusion*, 41, 195–216. DOI 10.1016/j.inffus.2017.09.010.
2. Nguyen, T. T., Nguyen, M. P., Pham, X. C., Liew, A. W. C., Pedrycz, W. (2018). Combining heterogeneous classifiers via granular prototypes. *Applied Soft Computing*, 73, 795–815. DOI 10.1016/j.asoc.2018.09.021.
3. Vijayanand, R., Devaraj, D., Kannapiran, B. (2018). Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection. *Computers & Security*, 77, 304–314. DOI 10.1016/j.cose.2018.04.010.
4. Wen, G., Hou, Z., Li, H., Li, D., Jiang, L. et al. (2017). Ensemble of deep neural networks with probability-based fusion for facial expression recognition. *Cognitive Computation*, 9(5), 597–610. DOI 10.1007/s12559-017-9472-6.
5. Chen, S. S. C., Cui, H., Du, M. H., Fu, T. M., Sun, X. H. et al. (2019). Cantonese porcelain classification and image synthesis by ensemble learning and generative adversarial network. *Frontiers of Information Technology & Electronic Engineering*, 20(12), 1632–1643. DOI 10.1631/FITEE.1900399.
6. Zhu, Z., Wang, Z., Li, D., Du, W. (2019). Tree-based space partition and merging ensemble learning framework for imbalanced problems. *Information Sciences*, 503, 1–22. DOI 10.1016/j.ins.2019.06.033.
7. Seh, A. H., Al-Amri, J. F., Subahi, A. F., Agrawal, A., Pathak, N. et al. (2022). An analysis of integrating machine learning in healthcare for ensuring confidentiality of the electronic records. *Computer Modeling in Engineering & Sciences*, 130(3), 1387–1422. DOI 10.32604/cmescs.2022.018163.
8. He, Q., Tong, H., Liu, J. B. (2022). How does inequality affect the residents' subjective well-being: Inequality of opportunity and inequality of effort. *Frontiers in Psychology*, 13, 843854. DOI 10.3389/fpsyg.2022.843854.
9. Wang, Y., Wang, D., Geng, N., Wang, Y., Yin, Y. et al. (2019). Stacking-based ensemble learning of decision trees for interpretable prostate cancer detection. *Applied Soft Computing*, 77, 188–204. DOI 10.1016/j.asoc.2019.01.015.
10. Zou, Y., Zhang, S., Min, Y. (2019). Exploring urban population forecasting and spatial distribution modeling with artificial intelligence technology. *Computer Modeling in Engineering & Sciences*, 119(2), 295–310. DOI 10.32604/cmescs.2019.03873.
11. Kiziloz, H. E. (2021). Classifier ensemble methods in feature selection. *Neurocomputing*, 419, 97–107. DOI 10.1016/j.neucom.2020.07.113.
12. Martínez-Muñoz, G., Hernández-Lobato, D., Suárez, A. (2008). An analysis of ensemble pruning techniques based on ordered aggregation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2), 245–259. DOI 10.1109/TPAMI.2008.34.
13. Lu, H. J., An, C. L., Zheng, E. H., Lu, Y. (2014). Dissimilarity based ensemble of extreme learning machine for gene expression data classification. *Neurocomputing*, 128, 22–30. DOI 10.1016/j.neucom.2013.02.052.
14. Dai, Q., Zhang, T., Liu, N. (2015). A new reverse reduce-error ensemble pruning algorithm. *Applied Soft Computing*, 28, 237–249. DOI 10.1016/j.asoc.2014.10.045.
15. Ykhlef, H., Bouchaffra, D. (2017). An efficient ensemble pruning approach based on simple coalitional games. *Information Fusion*, 34, 28–42. DOI 10.1016/j.inffus.2016.06.003.
16. Zhu, X., Ni, Z., Ni, L., Jin, F., Cheng, M. et al. (2019). Improved discrete artificial fish swarm algorithm combined with margin distance minimization for ensemble pruning. *Computers & Industrial Engineering*, 128, 32–46. DOI 10.1016/j.cie.2018.12.021.
17. Zhou, Z. H., Wu, J., Tang, W. (2002). Ensembling neural networks: Many could be better than all. *Artificial Intelligence*, 137(1–2), 239–263. DOI 10.1016/S0004-3702(02)00190-X.

18. Ni, Z., Zhang, C., Ni, L. (2016). Haze forecast method of selective ensemble based on glowworm swarm optimization algorithm. *Pattern Recognition and Artificial Intelligence*, 29(2), 143–153.
19. Zhu, X., Ni, Z., Cheng, M., Jin, F., Li, J. et al. (2018). Selective ensemble based on extreme learning machine and improved discrete artificial fish swarm algorithm for haze forecast. *Applied Intelligence*, 48(7), 1757–1775. DOI 10.1007/s10489-017-1027-8.
20. Cavalcanti, G. D., Oliveira, L. S., Moura, T. J., Carvalho, G. V. (2016). Combining diversity measures for ensemble pruning. *Pattern Recognition Letters*, 74, 38–45. DOI 10.1016/j.patrec.2016.01.029.
21. Dai, Q., Ye, R., Liu, Z. (2017). Considering diversity and accuracy simultaneously for ensemble pruning. *Applied Soft Computing*, 58, 75–91. DOI 10.1016/j.asoc.2017.04.058.
22. Yu, J., Yang, M. S. (2017). Deterministic annealing gustafson-kessel fuzzy clustering algorithm. *Information Sciences*, 417, 435–453. DOI 10.1016/j.ins.2017.07.005.
23. Bai, L., Liang, J., Cao, F. (2020). A multiple k-means clustering ensemble algorithm to find nonlinearly separable clusters. *Information Fusion*, 61, 36–47. DOI 10.1016/j.inffus.2020.03.009.
24. He, Q., Xia, P., Li, B., Liu, J. B. (2021). Evaluating investors' recognition abilities for risk and profit in online loan markets using nonlinear models and financial Big data. *Journal of Function Spaces*, 2021, 5178970. DOI 10.1155/2021/5178970.
25. Huang, G. B., Zhu, Q. Y., Siew, C. K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1–3), 489–501. DOI 10.1016/j.neucom.2005.12.126.
26. Zhang, Y., Wu, J., Cai, Z., Zhang, P., Chen, L. (2016). Memetic extreme learning machine. *Pattern Recognition*, 58, 135–148. DOI 10.1016/j.patcog.2016.04.003.
27. Li, Y., Zeng, Y., Qing, Y., Huang, G. B. (2020). Learning local discriminative representations via extreme learning machine for machine fault diagnosis. *Neurocomputing*, 409, 275–285. DOI 10.1016/j.neucom.2020.05.021.
28. Cai, Y., Liu, X., Zhang, Y., Cai, Z. (2018). Hierarchical ensemble of extreme learning machine. *Pattern Recognition Letters*, 116, 101–106. DOI 10.1016/j.patrec.2018.06.015.
29. Yang, Y., Zhang, H., Yuan, D., Sun, D., Li, G. et al. (2019). Hierarchical extreme learning machine based image denoising network for visual Internet of Things. *Applied Soft Computing*, 74, 747–759. DOI 10.1016/j.asoc.2018.08.046.
30. Shukla, S., Raghuwanshi, B. S. (2019). Online sequential class-specific extreme learning machine for binary imbalanced learning. *Neural Networks*, 119, 235–248. DOI 10.1016/j.neunet.2019.08.018.
31. Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H. et al. (2017). Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, 114, 163–191. DOI 10.1016/j.advengsoft.2017.07.002.
32. Kansal, V., Dhillon, J. S. (2020). Emended salp swarm algorithm for multiobjective electric power dispatch problem. *Applied Soft Computing*, 90, 106172. DOI 10.1016/j.asoc.2020.10617.
33. Zhu, X., Xia, P., He, Q., Ni, Z., Ni, L. (2022). Coke price prediction approach based on dense GRU and opposition based learning salp swarm algorithm. *International Journal of Bio-Inspired Computation*, 2022. DOI 10.1504/IJBIC.2022.10047653.
34. Xia, X., Lin, T., Chen, Z. (2018). Maximum relevancy maximum complementary based ordered aggregation for ensemble pruning. *Applied Intelligence*, 48(9), 2568–2579. DOI 10.1007/s10489-017-1106-x.
35. Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2), 123–140. DOI 10.1007/BF00058655.
36. Xin, G., Wang, P., Jiao, Y. (2021). Multiscale quantum harmonic oscillator optimization algorithm with multiple quantum perturbations for numerical optimization. *Expert Systems with Applications*, 185, 115615. DOI 10.1016/j.eswa.2021.115615.
37. Spana, S., Du, L. (2022). Optimal information perturbation for traffic congestion mitigation: Gaussian process regression and optimization. *Transportation Research Part C: Emerging Technologies*, 138, 103647. DOI 10.1016/j.trc.2022.103647.

38. Gong, T., Fan, T., Pei, L., Cai, Z. (2017). Magnetic resonance imaging-clonal selection algorithm: An intelligent adaptive enhancement of brain image with an improved immune algorithm. *Engineering Applications of Artificial Intelligence*, 62, 405–411. DOI 10.1016/j.engappai.2016.10.004.
39. Guo, H., Liu, H., Li, R., Wu, C., Guo, Y. et al. (2018). Margin & diversity based ordering ensemble pruning. *Neurocomputing*, 275, 237–246. DOI 10.1016/j.neucom.2017.06.052.
40. Ding, S., Chen, Z., Zhao, S. Y., Lin, T. (2018). Pruning the ensemble of ANN based on decision tree induction. *Neural Processing Letters*, 48(1), 53–70. DOI 10.1007/s11063-017-9703-6.
41. Singhal, P. K., Naresh, R., Sharma, V. (2015). Binary fish swarm algorithm for profit-based unit commitment problem in competitive electricity market with ramp rate constraints. *IET Generation, Transmission & Distribution*, 9(13), 1697–1707. DOI 10.1049/iet-gtd.2015.0201.
42. Azad, M. A. K., Rocha, A. M. A., Fernandes, E. M. (2014). Improved binary artificial fish swarm algorithm for the 0–1 Multidimensional knapsack problems. *Swarm and Evolutionary Computation*, 14, 66–75. DOI 10.1016/j.swevo.2013.09.002.
43. Zhou, Y. Q., Huang, Z. X., Liu, H. X. (2012). Discrete glowworm swarm optimization algorithm for TSP problem. *Acta Electronica Sinic*, 40(6), 1164–1170.
44. Li, M., Wang, X., Gong, Y., Liu, Y., Jiang, C. (2014). Binary glowworm swarm optimization for unit commitment. *Journal of Modern Power Systems and Clean Energy*, 2(4), 357–365. DOI 10.1007/s40565-014-0084-9.
45. Jordehi, A. R. (2019). Binary particle swarm optimisation with quadratic transfer function: A new binary optimisation algorithm for optimal scheduling of appliances in smart homes. *Applied Soft Computing*, 78, 465–480. DOI 10.1016/j.asoc.2019.03.002.
46. Islam, M. J., Li, X., Mei, Y. (2017). A time-varying transfer function for balancing the exploration and exploitation ability of a binary PSO. *Applied Soft Computing*, 59, 182–196. DOI 10.1016/j.asoc.2017.04.050.
47. Zhang, X., Ding, Z., Hang, J., He, Q. (2022). How do stock price indices absorb the COVID-19 pandemic shocks? *The North American Journal of Economics and Finance*, 60, 101672. DOI 10.1016/j.najef.2022.101672.