Hindawi
Computational and Mathematical Methods in Medicine
Volume 2021, Article ID 5588385, 13 pages
https://doi.org/10.1155/2021/5588385

Research Article

Feature Selection on Elite Hybrid Binary Cuckoo Search in Binary Label Classification

Maoxian Zhao and Yue Qin

College of Mathematics and Systems Science, Shandong University of Science and Technology, Qingdao, Shandong 266590, China

Correspondence should be addressed to Yue Qin; 201982150013@sdust.edu.cn

Received 19 February 2021; Accepted 22 April 2021; Published 12 May 2021

Academic Editor: Waqas Haider Bangyal

For the low optimization accuracy of the cuckoo search algorithm, a new search algorithm, the Elite Hybrid Binary Cuckoo Search (EHBCS) algorithm, is improved by feature weighting and elite strategy. The EHBCS algorithm has been designed for feature selection on a series of binary classification datasets, including low-dimensional and high-dimensional samples by SVM classifier. The experimental results show that the EHBCS algorithm achieves better classification performances compared with binary genetic algorithm and binary particle swarm optimization algorithm. Besides, we explain its superiority in terms of standard deviation, sensitivity, specificity, precision, and F-measure.

1. Introduction

Feature selection attempts to find the most discriminative subset of features to bring reasonable recognition rates for some classifiers. Given a problem with d features, we have 2^d possible solutions, making an exhaustive search impracticable for high-dimensional feature spaces. In addition, the high-dimensional data also contains a large number of irrelevant and noise-polluted features, and there is often information redundancy between features. These factors will affect the learning effect of the learning algorithm and significantly increase the algorithm's computational complexity. Therefore, feature selection has become a research hot spot.

As a key technology of pattern recognition and machine learning, feature selection is an effective method to deal with high-dimensional data. Feature selection models can be divided into three categories [1]: filter [2], embedding [3], and wrapper [4]. Filter methods define the relevant features without prior classification of the data. The embedding method refers to the process of embedding the feature selection algorithm into the classification algorithm and conducting feature selection and training at the same time. Wrapper methods on the other hand incorporate classification algorithms to search for and select relevant features. The wrapper methods generally outperform filter methods in terms of classification accuracy [5]. Recent studies have shown that feature selection can better solve many practical problems, including classification and medical problems [6–9].

Another vital part of the feature selection process is the search strategy: selecting the feature subset that meets the optimal evaluation criteria, which is usually a combinatorial optimization problem. In recent years, metaheuristic algorithms based on biological behavior and physical systems in nature are proposed to solve the optimization problems [10]. Metaheuristic optimization algorithm, also known as natural heuristic algorithm, studies the evolutionary behavior of species and simulates it into computer science algorithms, including genetic algorithm [11], particle swarm optimization algorithm [12], bat algorithm [13, 14], and cuckoo algorithm [15]. The metaheuristic optimization algorithm has achieved good results in feature selection. For example, Liu et al. [16] combined genetic algorithm and simulated annealing algorithm to select feature subsets. The experiment result expresses the hybrid algorithm has high reliability and strong convergence. On the contrary, Siedlecki and Sklansky [17] combined genetic algorithm and feature selection to achieve a certain effect, but it exposed the problem of premature convergence of genetic algorithm. Kennedy and Eberhart [18]

proposed the binary particle swarm optimization algorithm called BPSO, which modified the traditional particle swarm optimization algorithm and solves the binary optimization problems. Besides, Firpi and Goodman [19] applied BPSO to feature selection problems.

The success of metaheuristic methods lies in the efficiency of search strategies and its ability to find solutions to combinatorial optimization problems. Metaheuristics take the information gathered during the search to guide the search process, and therefore, they are considered independent of the problems. The cuckoo search algorithm is a novel heuristic optimization approach introduced by Yang and Deb in 2009 [15]. The algorithm simulates cuckoo birds' parasitic breeding habits and is a random algorithm with strong global search ability. The cuckoo search algorithm has been efficiently employed in many fields, such as intelligent optimization and calculation. Cuckoo search is superior to other algorithms in continuous optimization problems including spring design and welding beam in engineering design applications [20]. This algorithm is especially suitable for large-scale problems [21]. Valian et al. have applied it in training the neural network [22] and spike neural model [23]. The experiment proved that CS has better search capability than other algorithms like particle swarm optimization algorithm, genetic algorithm, and artificial bee colony algorithm [21, 24, 25]. Therefore, CS is a metaheuristic algorithm used in combinatorial optimization problems to obtain higher performance.

The CS can only solve optimization problems in the continuous solution space. To solve combinatorial optimization problems in discrete solution space, Gherboudj et al. [26] proposed a binary version of the cuckoo search algorithm, namely, BCS algorithm. Pereira and Rodrigues [27] applied BCS algorithm to feature selection. Bhattacharjee and Sarmah [28] improved BCS by using the balance combination of local random walk and global exploration random walk so that BCS algorithm can better balance locality and globality. Sudha and Selvarajan [29] presented a feature selection approach based on an enhanced cuckoo algorithm and applied it to breast X-ray images. It can supply valuable information for clinicopathologists. Aziz and Hassanien [30] proposed a new improved cuckoo algorithm combined with the theoretical knowledge of rough set and finally applied it to feature selection.

The cuckoo search algorithm uses Lévy flight random walk to search space in the iteration. The cuckoo search cannot effectively search around the cuckoo's nest due to the Lévy flight with sharp 90-degree turns. Therefore, it suffers from low optimization accuracy [31]. In order to improve the cuckoo search algorithm, this paper proposes an Elite Hybrid Binary Cuckoo Search algorithm, and the novelty of the paper is two-fold:

(1) EHBCS adopts feature weighting and elite strategy in the binary cuckoo search algorithm. Feature weighting based on Relief algorithm is to estimate the feature weight and its importance according to the ability of each feature to distinguish different class instances. Elite strategy and genetic algorithm with the selection and crossover operators are embedded into the cuckoo algorithm so that the well-positioned nests can be inherited to the next generation

(2) EHBCS is applied to a set of binary label datasets, including low-dimensional and high-dimensional samples such that only the best features are retained in the subset. Experimental results demonstrate that EHBCS achieves a better classification performance to minimize the number of selected features, simultaneously maximizing the classification accuracy by SVM compared with binary genetic algorithm and binary particle swarm optimization

The main contributions of this paper are summarized as follows: (1) It is the first time to combine the feature weighting and elite strategy with BCS algorithm. (2) It specifically improves the low optimization accuracy of the BCS algorithm. (3) It may provide a useful revelation to high-dimensional data researches such as text processing, medical research, and gene analysis.

The structure of this paper is as follows: Section 2 provides details of the classical version of the Cuckoo Search and Binary Cuckoo Search algorithms; Section 3 presents the Elite Hybrid Binary Cuckoo Search (EHBCS) algorithm; Section 4 discusses the experimental methodology and in particular the dataset and evaluation measures; numerical experiment is also carried out to evaluate the prediction performance of our method in Section 5. The results demonstrate that the proposed method is efficient for high-dimensional datasets; finally, the conclusions of our work are given in Section 6.

2. Cuckoo Search Algorithm

2.1. Cuckoo Search (CS) Algorithm. The parasite behavior of cuckoos is extremely intriguing. These birds can lay down their eggs in host nests and mimic external characteristics of host eggs such as color and spots. If this strategy is unsuccessful, the host can throw the cuckoo's eggs away or simply abandon its nest, making a new one in another place. Based on this context, Yang and Deb [15] have developed a novel evolutionary optimization algorithm named cuckoo search (CS), and they have summarized CS using three rules, as follows:

(1) Each cuckoo chooses a nest to lays eggs randomly

(2) The number of available host nests is fixed, and nests with high-quality eggs will be passed on the next generations

(3) If a host bird discovered the cuckoo egg, it can throw the egg away or abandon the nest and build a completely new nest

For optimization problems, each nest represents a possible solution to the problems, and a nest can contain one or more eggs depending on the size of the problems. Firstly, the algorithm randomly initializes each nest, and then, the

algorithm carries out an iterative process. During each iteration, each nest is updated by Lévy flight with random walk, and the formula is shown in Equations (1) and (2):

$$x_i^{t+1} = x_i^t + \alpha \oplus \text{le}'\text{vy}(\lambda). \quad (1)$$

The updating formula of each dimension is expressed as

$$x_{ij}^{t+1} = x_{ij}^t + \alpha \times \text{le}'\text{vy}_j(\lambda), \quad (2)$$

where x_i^t denotes ith nest and x_{ij}^t stands for the jth eggs at nest i for the t generation. α is step size, and the product $\oplus$ means entrywise multiplications. In most case, we can use $\alpha = 1$. The Lévy flights Lévy (λ) employ a random step length, and Lévy $_j(\lambda)$ is its jth component.

In the 1930s, Lévy proposed Lévy's distribution, believing that the relationship between the continuous jump path of Lévy's flight and time t follows Lévy's distribution. Later, many scholars have studied Lévy's distribution and used it to explain random phenomena in nature, such as Brownian motion and random walk. Yang [15] studied and obtained the probability density function of Lévy distribution in power form by simplifying and Fourier transform:

$$\text{le}'\text{vy} \sim u = t^{-\lambda}(1 < \lambda \leq 3), \quad (3)$$

where λ is the power coefficient. Equation (2) is a probability distribution with a heavy tail. Although it can essentially describe the random walk process of cuckoo birds, it has not been further described in a more concise and easy to program mathematical language to achieve the CS algorithm. So Yang adopted the Mantegna algorithm to simulate Lévy jump path:

$$s = \frac{u}{|v|^{1/\beta}}, \quad (4)$$

where s is the Lévy flight Lévy $_j(\lambda)$, the relation of parameters β in equation (2) is $\lambda = 1 + \beta$ and content $0 < \beta \leq 2$. The parameter is $\beta = 1.5$, and μ and v are random number and satisfy Equations (5) and (6):

$$\begin{cases} \mu \sim N\left(0, \sigma_\mu^2\right) \\ v \sim N\left(0, \sigma_v^2\right) \end{cases}, \quad (5)$$

$$\begin{cases} \sigma_\mu = \left\{ \dfrac{\Gamma(1+\beta) \cdot \sin(\beta\pi/2)}{\Gamma(1+\beta/2) \cdot \beta \cdot 2^{(\beta-1)/2}} \right\}^{1/\beta}, \\ \sigma_v = 1. \end{cases} \quad (6)$$

Let step $= \alpha \times \text{levy}(\lambda)_j = \alpha \times s$ then step is the path that cuckoo bird experiences each time in solution space when it randomly searches for the new nest location x_{ij}^{t+1} from the old nest location x_{ij}^t according to Equation (2). In the finally step of each iteration, the nest with the worst quality is substituted with probability $p_a \in [0,1]$. Algorithm 1 shows the pseudo-code for the classical version of CS.

2.2. Binary Cuckoo Search (BCS) Algorithm. In traditional CS, the position of the solution is updated in the continuous search space. Unlike the above CS, the BCS search space for feature selection is modeled as a binary d-bit string, where d is the number of features. BCS represents each nest as a binary vector, where each 1 corresponds to a selected feature and 0 otherwise. This means each nest represents a possible solution, and each nest represents a feature.

$$x_{ij} = \begin{cases} 1 & x_i \text{ selects the } j\text{th feature} \\ 0 & \text{otherwise} \end{cases}. \quad (7)$$

The original cuckoo algorithm introduces mapping functions to extend the cuckoo algorithm to discrete binary regions as follows [25]:

$$\text{sig}(\text{step}) = \frac{1}{1 + e^{-\gamma \text{step}}}, \gamma = 1, \quad (8)$$

$$x_{ij}^t = \begin{cases} 1 & \text{rand}\,() \leq \text{sig}(\text{step}) \\ 0 & \text{otherwise} \end{cases}, \quad (9)$$

in which rand $() \sim U(0,1)$ and x_{ij}^t denotes the new egg's value at iteration t.

3. Elite Hybrid Binary Cuckoo Search (EHBCS) Algorithm

3.1. Feature Weighting Based on Relief Algorithm. The core idea of feature weighting based on Relief is to estimate the feature weight and its importance according to the ability of each feature to distinguish different class instances [32]. Given a two-class dataset D, C containing n cases is a class label set, $x = (x_1, x_2, \cdots, x_d)$ is a case in D, and x is a real-valued vector with dimension d. Relief performs the following iterative learning: randomly select a case x, then find the nearest case $\text{NH}(x)$ of the same class and the nearest case $\text{NM}(x)$ of the different class, and then update the weight using the following rules:

$$w_j = w_j + \frac{1}{T}\left|x_j - \text{NM}(x)_j\right| - \frac{1}{T}\left|x_j - \text{NH}(x)_j\right|, \quad (10)$$

where w_j represents the weight of the jth feature and T represents the maximum number of iterations. $|x_j - y_j|$ is used to calculate the difference between the jth dimensional eigenvalues of two instances, that is, the absolute value vector of the feature difference vector.

A variant that considers k neighbors has been developed from the nearest neighbor Relief, whose weight value update formula is

$$w_j = w_j + \sum_{z \in \text{KNN}(x,l), l \in c} \left|x_j - z_j\right| / T - \sum_{z \in \text{KNN}(x,c)} \left|x_j - z_j\right| / T, \quad (11)$$

Objective function f(x) $x = (x_1, \cdots, x_d)^T$
Generate initial population of n host nest x_i,(i=1,2,$\cdots$,n)
while ($t < MaxGeneration$) *or (stop criterion)***do**
Get a cuckoo randomly by Levy flights evaluate its quality/fitness F_i
Choose a nest among n (say, j);
if ($F_i < F_j$) **then**
replace j by new solution;
end
A fraction (p_a) of worse nests are abandoned and new ones are built
Keep the best solution (or nests with quality solutions)
Rank the solutions and find the current best
end
Postprocess results and visualization

Algorithm 1: Classical version of CS adapted from [15].

Input: binary label dataset D with n cases and d dimensions, Maxiter T
Output: weight vector w
for*each* $j(j = 1, \cdots, d)$**do**
$w_j = 0$;
end
while$(t < T)$**do**
Randomly select an case x from the dataset D and calculate the distance between k nearest cases of the same kind $NH(x)$ and k nearest cases of the different kind $NM(x)$;
for*each* $j(j = 1, \cdots, d)$**do**
w_j generated by formula (11);
$t = t + 1$;
end
end

Algorithm 2: Relief algorithm.

where $\mathrm{KNN}(x; c)$ is the set of k nearest neighbors of x in X_c by Euclidean distance. Process is shown in Algorithm 2.

3.2. Selection and Crossover Operator. The selection operator is to inherit the individuals with high fitness in the current population to the next generation according to selection probability. Generally, individuals with high fitness will have more chances to inherit to the next generation. This paper uses the roulette model to select individuals. The calculation formula is as follows:

$$p(x_i) = \frac{f(x_i)}{\sum_{j=1}^{n} f(x_j)}, \tag{12}$$

$$q_i = \sum_{j=1}^{i} p(x_j), \tag{13}$$

where $p(x_i)$ is the selection probability, q_i is the cumulative probability, $f(x_i)$ is the individual x_i fitness function value, and n is the number of the group. Select operator process is in Algorithm 3.

Crossover is to cross the selected a pair of individuals according to probability, such as single-point crossover or multipoint crossover. In this paper, the single-point crossover is adopted, that is, the random number is generated within the range of individual coding bits as the crossover point, and then, the coding exchange of the two bodies from this point to the end is carried out, so that the crossover process can be completed.

3.3. Weight-Based Elite Hybrid Binary Cuckoo Search (EHBCS) Algorithm. In the CS algorithm, the Lévy flight is used to explore the search space using a straight flight path with a sudden 90-degree turn, and Figure 1 simulates Lévy's flight path. In addition, the CS algorithm is highly dependent on random walk search, which can be easily moved from one area to another without carefully exploring each nest. Therefore, the CS algorithm has weak local search ability and low optimization accuracy [31]. In order to cover the mentioned weakness of the CS, elite strategy and genetic algorithm operators are embedded into the cuckoo algorithm, such as selection and crossover operators, so that the well-positioned nests can be inherited to the next generation. The so-called elitist strategy is to preserve the nest in a good location so as not to miss the optimal nest during the algorithm iterations by Lévy flight. According to certain rules, the selection operator is to inherit the individuals with high fitness in the current population to the next generation. Generally, individuals with high fitness will have more chances to inherit

Input: population with n nests, number of dimensions (features) d, crossover rate p_c, fitness function $f(x)$
Output: New population *newnest* after elite selection and crossover
for*each* $i(i = 1, \cdots, n)$**do**
$p(x_i)=f(x_i)/\sum_{j=1}^{n} f(x_j)$, $q_i = \sum_{j=1}^{i} p(x_j)$;
for*each* $k(k = 1, \cdots, n)$**do**
Generate a random number r_k from [0,1];
if $(r_k \le q_1)$ **then**
Select the x_1;
else
if$(q_{i-1} < r_k \le q_i (2 \le i \le n))$**then**
Select the x_i;
end
end
end
end
Train the classifier to evaluate accuracy of x_i;
Calculate the fitness function value and store it in $fitness_i$;
$[Maxfitness, index] = \max(fitness_i)(i = 1, \cdots, n)$;
$Bestnest = x_{index}$;
$Bestfitness = Maxfitness$;
Two nests in the population are paired at random for each pair except *bestnest*, such as x_i and $x_j (i, j \ne index)$
if $(rand(0, 1) < p_c)$ **then**
Generate a random integer r in (1,d) with one-point crossover between two individuals $x_i = x_{i1}, x_{i2}, \cdots, x_{ir-1}, x_{jr}, \cdots, x_{jd}$;
$x_j = x_{j1}, x_{j2}, \cdots, x_{jr-1}, x_{ir}, \cdots, x_{id}$;
end
The crossed nests and Bestness form a new population *newnest* as output

Algorithm 3: Elite selection and crossover operators.

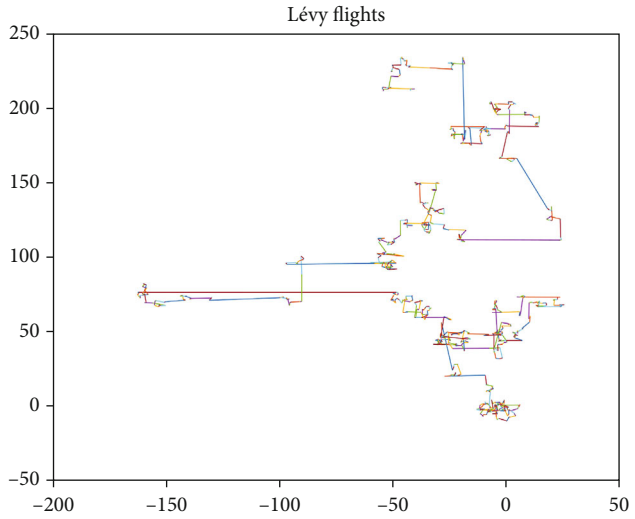


Figure 1: Lévy flight path.

to the next generation. The crossover operator usually inputs two individuals as candidate solutions with a certain probability and generates neighborhood solutions by exchanging part of the chromosomes of two individuals.

The CS algorithm is suitable for continuous domain problems, and the feature selection is a binary discrete problem. This paper proposes an Elite Hybrid Binary Cuckoo Search (EHBCS) algorithm considering these facts. The EHBCS algorithm weights the features firstly according to the Relief algorithm mentioned in part III-A, so that the features with larger weights have greater opportunities to be selected. Then, in each iteration of the EHBCS algorithm, the optimal nest does not carry out Lévy flight or crossover to avoid damaging the optimal nest position. The nest generated by Lévy flight is operated by selection and crossover operators.

Since the existing BCS algorithm does not consider the influence brought by the Sig(step) function, the coefficient in the Sig(step) function is changed to the feature weight in this paper so that features with significant feature weight have a greater chance to be selected and the improved algorithm can finish the iterative process faster. The BCS mapping function is modified as follows:

When step ≥ 0

$$\text{sig}(\text{step}) = \frac{1}{1 + e^{-\gamma \text{step}}}, \gamma = w_j, \tag{14}$$

$$x_{ij}^t = \begin{cases} 1 & \text{rand}\,() \le \text{sig}(\text{step}) \\ 0 & \text{otherwise} \end{cases}. \tag{15}$$

When step < 0

$$\text{sig}(\text{step}) = 1 - \frac{1}{1 + e^{-\gamma \text{step}}}, \gamma = w_j, \tag{16}$$

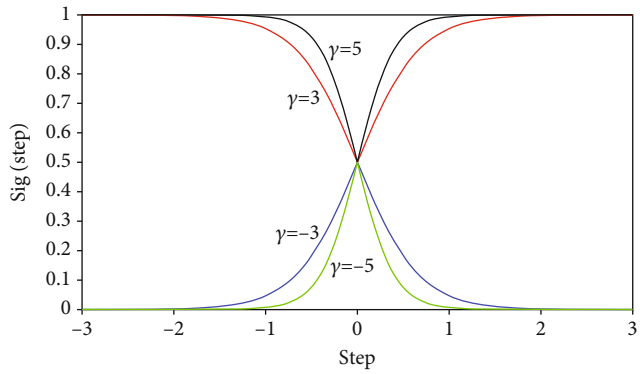


Figure 2: Parameter comparison.

$$x_{ij}^{t} = \begin{cases} 1 & \text{rand}\,() \le \text{sig(step)} \\ 0 & \text{otherwise} \end{cases}. \tag{17}$$

The function of sig(step) does not represent the probability of change, and it represents the probability of a certain change being 1. Let $\gamma = -5, -3, 3, 5$. The corresponding function graph is shown in Figure 2. It can be seen from the figure that the greater the parameter of the same abscissa, the greater the corresponding value. That is, the greater the feature weight, the greater the probability of being selected.

It should be emphasized that the weights calculated by the Relief algorithm may have negative weights, and the negative weight indicates that the distance of the similar neighbor samples is larger than that of the nonsimilar neighbor samples. Therefore, it is considered that this feature is unfavorable to classification, and the probability of selecting this feature in the corresponding feature selection is low.

Because the purpose of nest discovery and crossover operation is to make the population various, this paper adopts crossover operation instead of discovery operation. In the late iteration of the algorithm, the elite strategy proposed in this paper ensures the convergence. The elite selection and crossover operators as well as the pseudo-code of the algorithm presented in this paper are as follows: Algorithm 3 and Algorithm 4.

4. Experimental Methodology

4.1. Datasets. Eight datasets were extracted from the UCI Machine Learning Repository [33–35]. In order to make a more comprehensive comparison between the proposed algorithm and other algorithms, four low-dimensional feature datasets and four high-dimensional feature datasets are selected. Each dataset has two classes, and Table 1 provides the datasets' names, the total number of features, total number of cases, and classification accuracy before feature selection.

4.2. Performance Evaluation Measures. Generalization ability is the ability of a model to predict new data accurately after training on the training datasets. Cross-validation is a method to evaluate model generalization ability, which is widely used in data mining and machine learning [36]. In cross-validation, the dataset is usually divided into two parts: the training set, which is used to build a prediction model, and the other is test set, which is used to test the model's generalization ability. Cross-validation was performed, and the value of k was set to $k = 5$ for datasets with cases below 100 and to $k = 10$ for datasets with cases above 100. The evaluation indicators used include Accuracy, Sensitivity, Precision, and F-measure [37].

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{18}$$

$$\text{sensitivity} = \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{19}$$

$$\text{specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}}, \tag{20}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{21}$$

$$F - \text{measure} = \frac{2 * (\text{precision} * \text{recall})}{\text{precision} + \text{recall}}, \tag{22}$$

whereTPis the total number of positive cases and correctly identified as positive,TNis the total number of negative cases and correctly identified as negative,FPis the total number of negative cases and wrongly identified positive cases, andFNis the total number of positive cases and wrongly identified negative cases.

For the overall classification performance of each algorithm, we calculate the average value of all tests as follows:

$$\text{acc} = \frac{1}{k}\sum_{i=1}^{k} \text{accuracy}_i, \tag{23}$$

$$\text{SE} = \frac{1}{k}\sum_{i=1}^{k} \text{sensitivity}_i, \tag{24}$$

$$\text{SP} = \frac{1}{k}\sum_{i=1}^{k} \text{specificity}_i, \tag{25}$$

$$\text{Pre} = \frac{1}{k}\sum_{i=1}^{k} \text{precision}_i, \tag{26}$$

$$F1 = \frac{1}{k}\sum_{i=1}^{k} F - \text{measure}_i, \tag{27}$$

where k is the total number folds.

4.3. Evaluating Classification Performance. The support vector machine (SVM) classifier was adopted to evaluate the accuracy of feature subset classification. SVM is a supervised machine learning algorithm introduced by Boser et al. [38], in which data is mapped as the points in an n-dimensional feature space (n = number of features). The final output of SVM is an optimal hyperplane that classifies new cases.

SVM highly depends on kernel functions, so the experiments with different kernel functions are fundamental. The kernel function is a similarity function, which determines

Input: labelled dataset D, Maxiter T, CS parameters value, number of nests n, number of dimensions (features) d
Output: $Bestfitness$, $Bestnest$
for*each* $i(i = 1, \cdots, n)$**do**
  x_i randomly generate a binary 0-1 string;
Train the classifier to evaluate accuracy of x_i;
Calculate the fitness function value and store it in $fitness_i$;
end
$[Maxfitness, index] = \max(fitness_i)(i = 1,\cdots,n)$;
$Bestness = x_{index}$;
$Bestfitness = Maxfitness$;
while$(t < T)$ or *(stop criterion)***do**
  for*each* $i(i = 1, \cdots, n-1)$**do**
    for*each* $j(i = 1, \cdots, d)$**do**
newnest generated by formula (14)–(17) and store it in $new_i(i = 1,\cdots,n-1)$;
  end
end
A new population *nest* with n members of $new_i(i = 1,\cdots,n-1)$ and $Bestnest$;
Train the classifier to evaluate accuracy of $nest_i(i = 1,\cdots,n)$;
Calculate the fitness function value and store it in fit_i;
for*each* $i(i = 1, \cdots, n)$**do**
if $(fit_i > fitness_i)$ **then**
  $fitness_i = fit_i$;
  $x_i = nest_i$;
  end
end
Generate new population *newnest* after elite selection and crossover;
Train the classifier to evaluate accuracy of x_i;
Calculate the fitness function value and store it in $fitness_i$;
$[Maxfitness, index] = max(fitness_i)(i = 1,\cdots,n)$;
$Bestness = x_{index}$;
$Bestfitness = Maxfitness$;
end

Algorithm 4: EHBCS algorithm.

Table 1: Datasets.

Datasets	Features	Cases	Accuracy
Cervical Cancer Behavior Risk	19	72	0.865
Breast Cancer Wisconsin (diagnostic)	30	569	0.627
Breast Cancer Wisconsin (prognosis)	33	198	0.763
Sonar	60	208	0.702
Colon Tumor	2000	62	0.853
Medulloblastomas	5893	34	0.648
Central Nervous System	7129	60	0.600
Relation Leukemia	7129	72	0.919

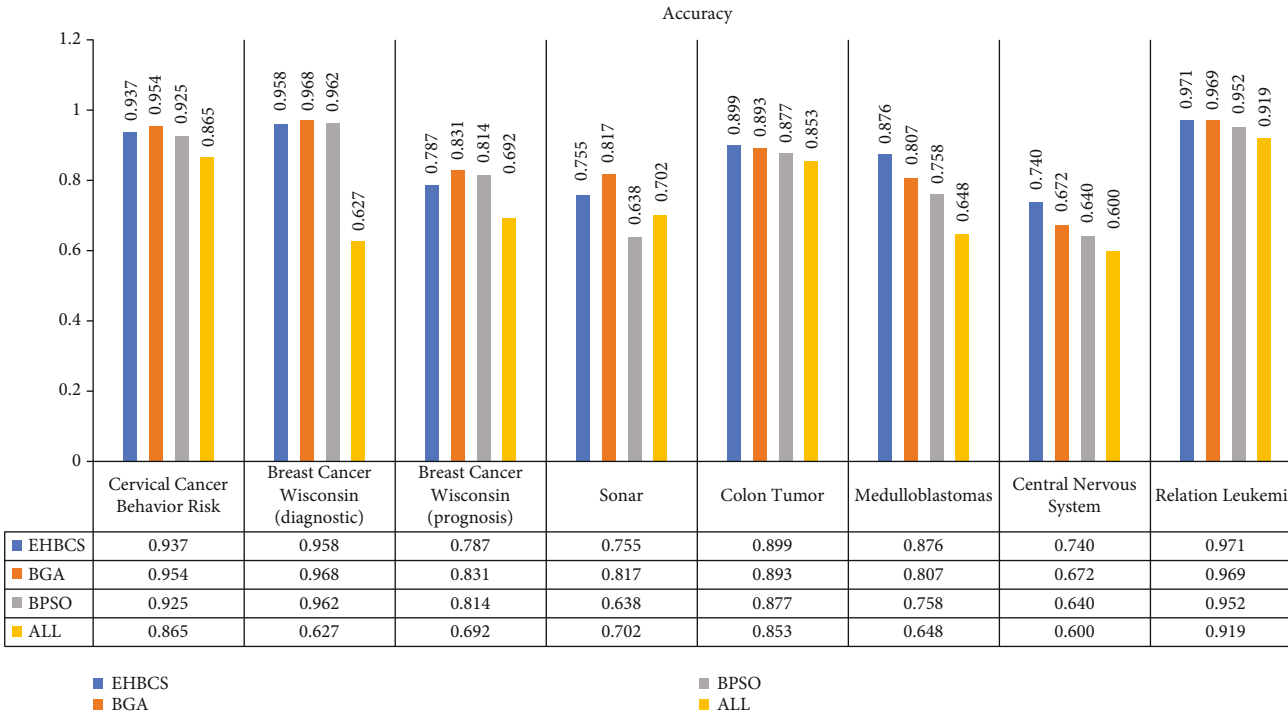
Table 2: Selected kernel functions.

Dataset	Kernel function
Cervical Cancer Behavior Risk	Radial basis function
Breast Cancer Wisconsin (diagnostic)	Radial basis function
Breast Cancer Wisconsin (prognosis)	Radial basis function
Sonar	Radial basis function
Colon Tumor	Linear function
Medulloblastomas	Linear function
Central Nervous System	Linear function
Relation Leukemia	Linear function

the similarity between any two inputs by calculating the distance between them. It is not difficult to determine the kernel function. Any function that satisfies the Mercer theorem can be used as a kernel function. There are various types of kernel functions such as linear kernel function, polynomial kernel function, radial basis kernel function, Sigmoid kernel function, and composite kernel function. Selecting the appropriate kernel function is relevant to the datasets and the problems. Therefore, it is often selected experimentally. Based on experiments, suitable kernel functions are selected

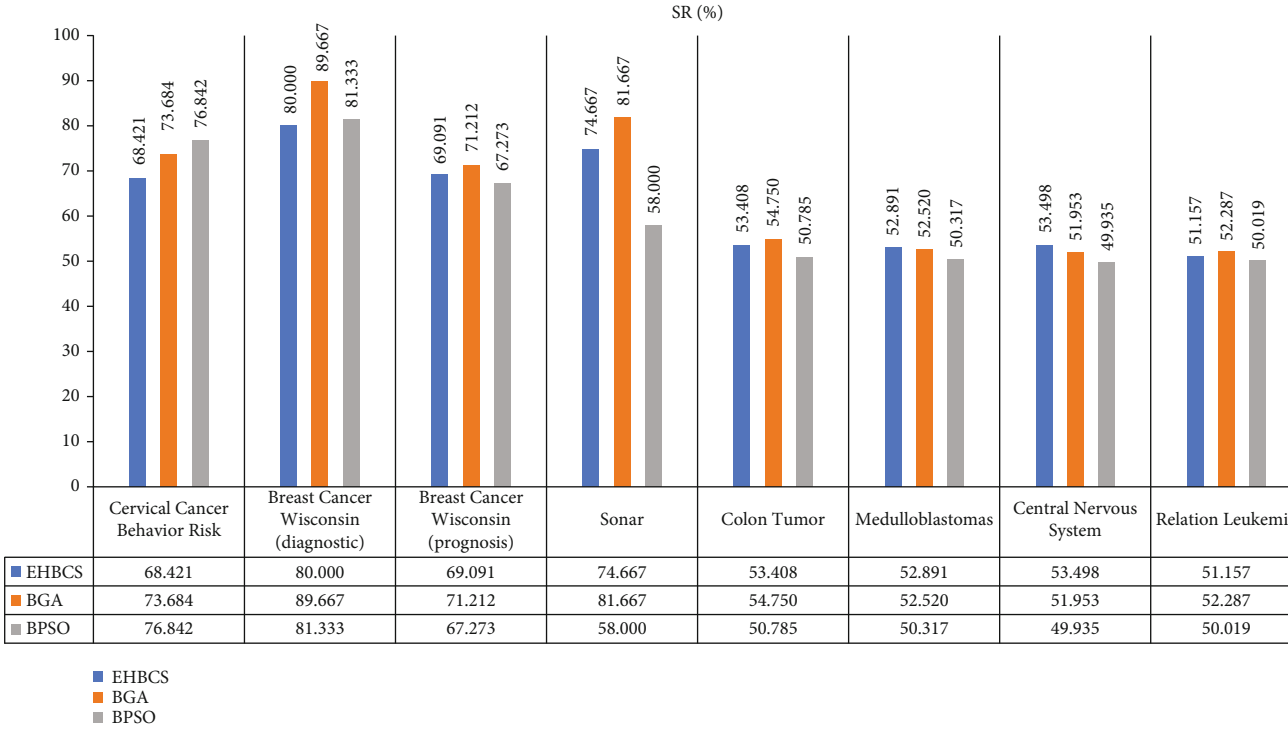
Table 3: Parameter setting.

Algorithms	Parameters
EHBCS	$\alpha = 1, p_c = 0.5, p_r = 0.3, k = 3$
BGA	$p_c = 0.8, p_m = 0.1$
BPSO	$c_1 = 1, c_2 = 2, \omega = 0.9$



	Cervical Cancer Behavior Risk	Breast Cancer Wisconsin (diagnostic)	Breast Cancer Wisconsin (prognosis)	Sonar	Colon Tumor	Medulloblastomas	Central Nervous System	Relation Leukemia
EHBCS	0.937	0.958	0.787	0.755	0.899	0.876	0.740	0.971
BGA	0.954	0.968	0.831	0.817	0.893	0.807	0.672	0.969
BPSO	0.925	0.962	0.814	0.638	0.877	0.758	0.640	0.952
ALL	0.865	0.627	0.692	0.702	0.853	0.648	0.600	0.919

(a)



	Cervical Cancer Behavior Risk	Breast Cancer Wisconsin (diagnostic)	Breast Cancer Wisconsin (prognosis)	Sonar	Colon Tumor	Medulloblastomas	Central Nervous System	Relation Leukemia
EHBCS	68.421	80.000	69.091	74.667	53.408	52.891	53.498	51.157
BGA	73.684	89.667	71.212	81.667	54.750	52.520	51.953	52.287
BPSO	76.842	81.333	67.273	58.000	50.785	50.317	49.935	50.019

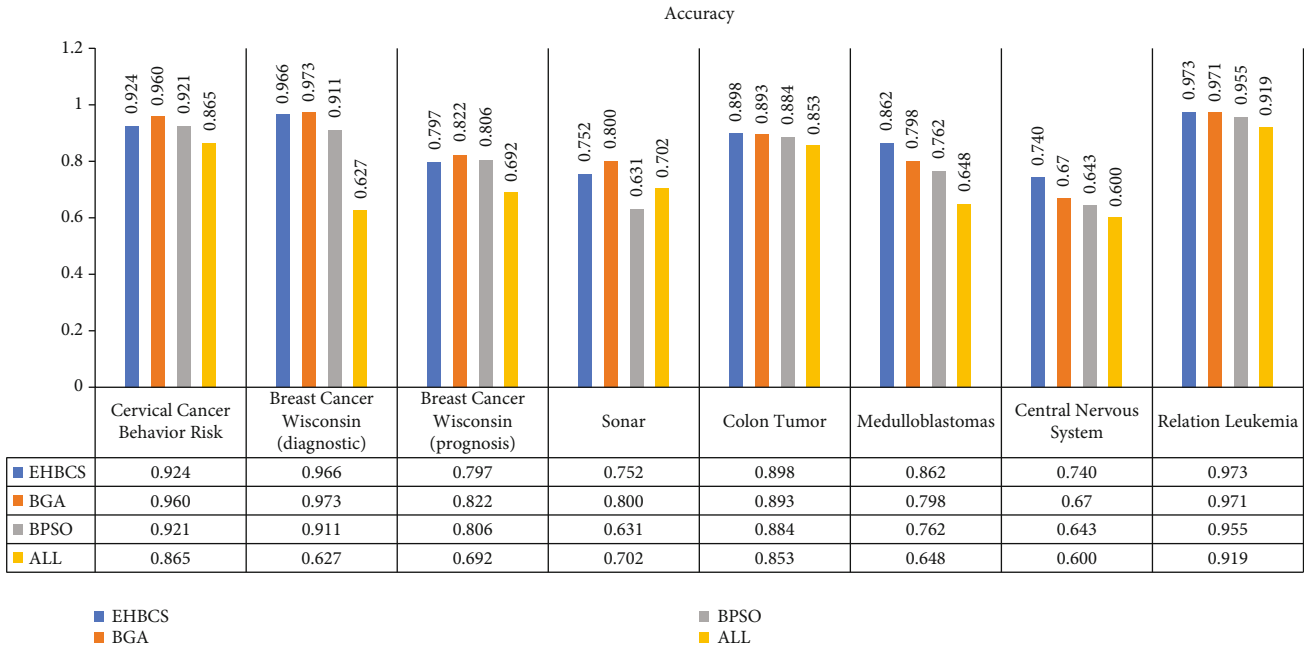
(b)

Figure 3: Accuracy and size reduction (%) with function f.

to evaluate the datasets. The selected kernel functions are presented in Table 2.

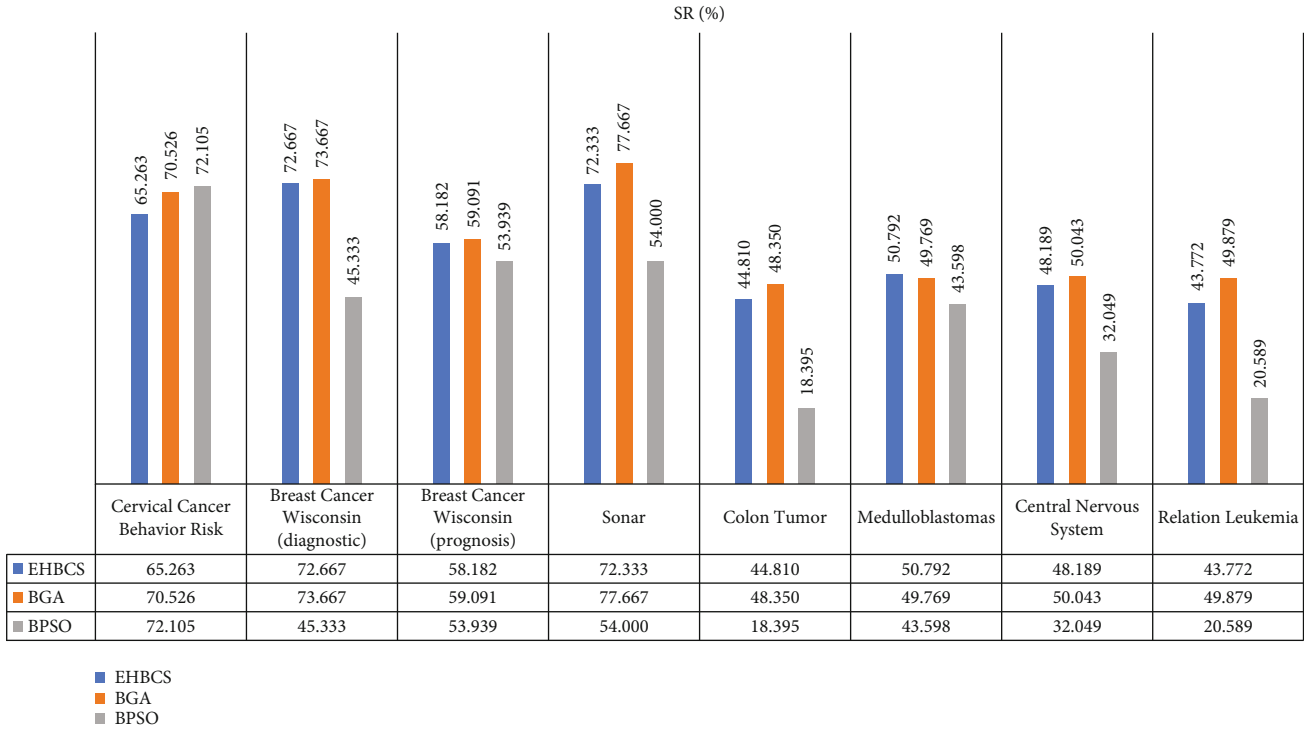
4.4. Fitness Function. The main objective of the feature selection task is to find a subset of features from the dataset so that the learning algorithm can use these selected features to achieve as high accuracy as possible.

In the classification problems, two feature subsets with different numbers likely have the same classification accuracy for the same dataset. Therefore, in the case of the same



	Cervical Cancer Behavior Risk	Breast Cancer Wisconsin (diagnostic)	Breast Cancer Wisconsin (prognosis)	Sonar	Colon Tumor	Medulloblastomas	Central Nervous System	Relation Leukemia
EHBCS	0.924	0.966	0.797	0.752	0.898	0.862	0.740	0.973
BGA	0.960	0.973	0.822	0.800	0.893	0.798	0.67	0.971
BPSO	0.921	0.911	0.806	0.631	0.884	0.762	0.643	0.955
ALL	0.865	0.627	0.692	0.702	0.853	0.648	0.600	0.919

(a)



	Cervical Cancer Behavior Risk	Breast Cancer Wisconsin (diagnostic)	Breast Cancer Wisconsin (prognosis)	Sonar	Colon Tumor	Medulloblastomas	Central Nervous System	Relation Leukemia
EHBCS	65.263	72.667	58.182	72.333	44.810	50.792	48.189	43.772
BGA	70.526	73.667	59.091	77.667	48.350	49.769	50.043	49.879
BPSO	72.105	45.333	53.939	54.000	18.395	43.598	32.049	20.589

(b)

Figure 4: Accuracy and size reduction (%) with function acc.

classification accuracy, if the metaheuristic algorithm finds the subset with more features earlier, the subset with fewer features will be ignored. In this paper, a new evaluation method is proposed as the fitness function to overcome this constraint, which considers the classification accuracy and takes the rate of feature reduction as an adjusting term.

Let d be the total number of features contained in the datasets, s be the number of features selected by metaheuristic optimization algorithms, β be the weight of rate of feature reduction, and 1-β be the weight of average accuracy. The value of the adaptation fitness function can be calculated as shown in (28). We set β=0.2.

$$f = \beta \cdot \frac{d - s}{d} + (1 - \beta) \cdot \text{acc}. \tag{28}$$

Table 4: Experimental results.

Fitness	Algorithm	Avgacc	Max	Min	Std	AvgN	SE	SP	Pre	F1	Dataset
f	EHBCS	0.937	0.920	0.948	0.011	6.000	0.840	0.966	0.895	0.854	Cervical Cancer Behavior Risk
	BGA	0.954	0.975	0.934	0.015	5.000	0.917	0.952	0.906	0.893	
	BPSO	0.925	0.946	0.896	0.014	4.400	0.700	0.978	0.967	0.775	
	Avg	0.937	0.947	0.926	0.013	5.133	0.819	0.965	0.908	0.841	
acc	EHBCS	0.924	0.932	0.906	0.011	6.600	0.785	0.960	0.821	0.793	
	BGA	0.960	0.973	0.945	0.009	5.600	0.883	0.981	0.943	0.896	
	BPSO	0.920	0.948	0.896	0.017	5.300	0.753	0.917	0.800	0.754	
	Avg	0.935	0.951	0.916	0.012	5.833	0.807	0.953	0.854	0.814	
f	EHBCS	0.957	0.961	0.953	0.003	6.000	0.947	0.960	0.936	0.940	Breast Cancer Wisconsin (diagnostic)
	BGA	0.968	0.975	0.960	0.005	3.100	0.970	0.955	0.926	0.947	
	BPSO	0.962	0.972	0.954	0.006	5.600	0.954	0.941	0.907	0.930	
	Avg	0.962	0.969	0.956	0.005	4.900	0.953	0.952	0.923	0.939	
acc	EHBCS	0.966	0.968	0.963	0.001	8.200	0.957	0.969	0.950	0.952	
	BGA	0.973	0.977	0.968	0.003	7.900	0.954	0.959	0.940	0.952	
	BPSO	0.911	0.935	0.879	0.020	16.400	0.957	0.961	0.937	0.946	
	Avg	0.950	0.960	0.937	0.008	10.833	0.956	0.963	0.942	0.948	
f	EHBCS	0.787	0.793	0.778	0.007	10.200	0.294	0.926	0.604	0.371	Breast Cancer Wisconsin (prognosis)
	BGA	0.831	0.848	0.808	0.010	9.500	0.200	0.958	0.593	0.277	
	BPSO	0.814	0.825	0.793	0.014	10.800	0.216	0.936	0.583	0.286	
	Avg	0.811	0.822	0.793	0.010	10.167	0.236	0.940	0.593	0.311	
acc	EHBCS	0.797	0.803	0.793	0.005	13.800	0.220	0.974	0.750	0.322	
	BGA	0.822	0.863	0.788	0.020	13.500	0.210	0.981	0.735	0.302	
	BPSO	0.806	0.829	0.793	0.010	15.200	0.191	0.968	0.733	0.289	
	Avg	0.808	0.832	0.791	0.012	14.167	0.207	0.974	0.739	0.304	
f	EHBCS	0.755	0.778	0.735	0.017	15.200	0.947	0.960	0.936	0.940	Sonar
	BGA	0.816	0.880	0.778	0.036	11.000	0.970	0.955	0.926	0.947	
	BPSO	0.637	0.663	0.610	0.018	25.200	0.954	0.941	0.907	0.930	
	Avg	0.736	0.774	0.708	0.024	17.133	0.952	0.502	0.704	0.789	
acc	EHBCS	0.752	0.773	0.730	0.014	16.600	0.957	0.969	0.950	0.952	
	BGA	0.800	0.865	0.760	0.035	13.400	0.954	0.959	0.940	0.945	
	BPSO	0.631	0.644	0.620	0.009	27.600	0.967	0.961	0.937	0.946	
	Avg	0.728	0.761	0.703	0.019	19.200	0.974	0.451	0.684	0.788	

4.5. Parameter Setting. The performance of the proposed EHBCS is compared against the Binary Genetic Algorithm (BGA) and Binary Particle Swarm Optimization (BPSO) algorithms. Table 3 lists the parameter values for each algorithm. The population size of all optimization algorithms is set to 30, and each algorithm was run 5 times to perform the feature selection task. All runs are executed in Matlab 2017, running on a Windows 10 operating system on a Huawei MagicBook with Intel(R) Core(TM) i5-8250U 1.6GHz with 8Gb of RAM.

4.6. Analysis of Computational Complexity. The EHBCS algorithm uses the Relief algorithm and the binary conversion of Lévy flight as well as the selection and crossover process. For the Relief algorithm, assuming that the number of runs is M, the number of iterations is m, the number of cases is N, and the individual dimension is d; the complexity of the algorithm is $O(m \times N \times d \times M)$. For Lévy flight and binary conversion, assuming that the number of individuals is n, the individual dimension is d, and the number of iterations is t; the computational complexity is $O(n^2 \times d \times t)$. For selection and crossover, assuming the number of individuals is n, the computational complexity is $O(n^2 \times t \times d)$. Therefore, the computational complexity is $O(m \times N \times d \times M + n^2 \times t \times d)$ for EHBCS algorithm.

5. Experimental Results

Figures 3 and 4 provide the performance of all optimization algorithms for feature selection using the medical datasets described in Section 4.1. They contain the following information:

Accuracy: classification accuracy for each datasets

All: classification accuracy before feature selection for each dataset

Table 5: Experimental results.

Fitness	Algorithm	Avgacc	Max	Min	Std	AvgN	SE	SP	Pre	F1	Dataset
f	EHBCS	0.899	0.901	0.886	0.006	931.833	0.927	0.860	0.938	0.922	Colon Tumor
	BGA	0.893	0.901	0.885	0.008	905.000	0.927	0.860	0.925	0.922	
	BPSO	0.877	0.886	0.868	0.008	984.300	0.927	0.753	0.888	0.901	
	Avg	0.889	0.896	0.879	0.007	933.744	0.927	0.824	0.917	0.915	
acc	EHBCS	0.898	0.903	0.886	0.006	1103.800	0.927	0.860	0.925	0.922	
	BGA	0.893	0.903	0.886	0.011	1033.000	0.927	0.860	0.905	0.901	
	BPSO	0.884	0.901	0.869	0.009	1632.100	0.927	0.793	0.910	0.913	
	Avg	0.892	0.902	0.881	0.007	1256.300	0.927	0.838	0.913	0.912	
f	EHBCS	0.876	0.876	0.876	0	2776.200	0.920	0.600	0.931	0.916	Medulloblastomas
	BGA	0.807	0.842	0.783	0.022	2798.600	0.920	0.600	0.931	0.917	
	BPSO	0.758	0.795	0.733	0.019	2927.800	0.910	0.600	0.921	0.912	
	Avg	0.814	0.838	0.797	0.014	2834.200	0.917	0.600	0.928	0.915	
acc	EHBCS	0.862	0.876	0.848	0.014	2899.800	0.910	0.550	0.915	0.907	
	BGA	0.798	0.817	0.783	0.015	2960.100	0.900	0.600	0.931	0.941	
	BPSO	0.764	0.767	0.762	0.002	3323.750	0.900	0.600	0.898	0.894	
	Avg	0.808	0.820	0.798	0.010	3061.217	0.903	0.583	0.915	0.914	
f	EHBCS	0.740	0.750	0.733	0.008	3315.100	0.436	0.894	0.767	0.525	Central Nervous System
	BGA	0.672	0.700	0.650	0.017	3425.300	0.360	0.794	0.430	0.380	
	BPSO	0.640	0.683	0.617	0.024	3569.100	0.360	0.737	0.396	0.360	
	Avg	0.684	0.711	0.667	0.016	3436.500	0.385	0.808	0.532	0.422	
acc	EHBCS	0.740	0.767	0.717	0.017	3693.600	0.500	0.894	0.787	0.572	
	BGA	0.670	0.717	0.633	0.023	3561.400	0.360	0.786	0.706	0.369	
	BPSO	0.643	0.650	0.633	0.008	4844.200	0.320	0.737	0.377	0.332	
	Avg	0.684	0.711	0.661	0.016	4033.067	0.393	0.806	0.623	0.425	
f	EHBCS	0.971	0.973	0.960	0.005	3482.000	1	0.930	0.953	0.975	Relation Leukemia
	BGA	0.970	0.973	0.960	0.006	3401.400	1	0.885	0.922	0.955	
	BPSO	0.952	0.960	0.947	0.007	3563.100	1	0.910	0.937	0.965	
	Avg	0.962	0.969	0.956	0.006	3482.167	1	0.908	0.937	0.965	
acc	EHBCS	0.973	0.987	0.960	0.006	4008.500	1	0.926	0.951	0.973	
	BGA	0.971	0.973	0.960	0.005	3573.100	1	0.910	0.937	0.965	
	BPSO	0.955	0.960	0.947	0.007	5661.200	1	0.910	0.937	0.965	
	Avg	0.967	0.973	0.956	0.006	4414.267	1	0.915	0.942	0.968	

SR: size reduction percentage is used to evaluate the percentage of removed features compared to all available features

Tables 4 and 5 provide the performance of all optimization algorithms for feature selection using binary label datasets described in Section 4.1. Each table column contains the following information:

Fitness: acc is accuracy as defined in Section 4.2 Function (23), and f is the proposed Function (28) as defined in Section 4.4

Algorithm: it provides the abbreviations of the algorithms, Elite Hybrid Binary Cuckoo Search (EHBCS), Binary Genetic Algorithm (BGA), and Binary Particle Swarm Optimization (BPSO)

Avgacc, Max, Min: average accuracy, maximum accuracy, minimum accuracy of an algorithm during the 5 runs

Std: standard deviation of classification accuracy

AvgN: average number of features returned by the algorithm during the 5 runs

SE, SP, Pre, F1: average sensitivity, specificity, precision, F-measure of an algorithm during the 5 runs

Dataset: the dataset used for experimentation as described in Table 1

Avg: average of all corresponding data obtained by the three algorithms

The experimental results show that the average feature subsets are smaller for all datasets, and the average classification accuracy is improved to different degrees. Compared with the original datasets, the number of the average feature subsets after feature selection by the optimization algorithms was reduced by about 18.395%-89.667%, and the average classification accuracy was improved by about 3.3%-34.6%. For the Breast Cancer Wisconsin (diagnostic) dataset, the maximum average

classification accuracy improvement was achieved at 34.6%. All these imply that the feature selection methods based on metaheuristic optimization algorithms can effectively eliminate redundant features and significantly improve the classification accuracy especially for some datasets.

For low-dimensional datasets, such as Cervical Cancer Behavior Risk, Breast Cancer Wisconsin (diagnostic), Breast Cancer Wisconsin (prognosis), and Sonar, the EHBCS algorithm can effectively reduce features to obtain a smaller subset of target features. It can get minimum standard deviation in three algorithms, which shows the EHBCS algorithm is the most stable of three. But it is the second of the three optimization algorithms in terms of classification accuracy, SE, SP, Pre, and F1. Compared with the data corresponding to Avg, the EHBCS algorithm has minimum standard deviation, higher classification accuracy, SE, SP, Pre, and F1 in entirety. Compared with the original dataset classification, the number of subset features after feature selection by the EHBCS algorithm is reduced by 58.182%-80%, and the classification accuracy is improved by 5%-33.9%. The results show that the EHBCS algorithm can efficiently diminish the number of features to ensure accuracy, but it did not perform well in low-dimensional datasets.

For high-dimensional datasets, such as Colon Tumor, Medulloblastomas, Central Nervous System and Relation Leukemia, the average classification accuracy, standard deviation, SE, SP, Pre, and F1 obtained by the EHBCS algorithm were superior to BGA and BPSO on the whole. Compared with the data corresponding to Avg, the average classification accuracy of the EHBCS algorithm is improved by 1%-10.6%, and the EHBCS gets lower standard deviation. But it needs to be explained that the standard deviation of the EHBCS algorithm is greater than the data corresponding to Avg when adopting fitness acc (Function (23)) for dataset Medulloblastomas and Central Nervous System. In addition to these, SE, SP, Pre, and F1 are optimal overall. Compared with the original dataset classification, the number of subset features after feature selection by the EHBCS algorithm is reduced by 43.772%-53.498%, and the classification accuracy is improved by 4.5%-22.8%. The results show that the feature selection method based on EHBCS has higher classification accuracy, SE, SP, Pre, F1, and smaller standard deviation. EHBCS algorithm is more suitable for the feature selection of high-dimensional datasets.

It should be emphasized that the purpose of feature selection is to reduce irrelevant or weakly correlated features as much as possible on the premise of ensuring classification accuracy. However, the number of feature subsets cannot be reduced indefinitely. Too few feature subsets may lead to the loss of important features, thus affecting the classification accuracy of the datasets. Therefore, it is necessary to balance the relationship between classification accuracy and the number of feature subsets. In practical applications, evaluation function models should be set scientifically and reasonably to ensure the classification performance of feature subsets.

6. Conclusion

This paper proposes an Elite Hybrid Binary Cuckoo Search Algorithm that adopts feature weighting and elite strategy. The proposed EHBCS algorithm aims to optimize the feature selection task on binary label datasets. The experimental results show that EHBCS achieves a better classification performance. Besides, all statistical metrics (standard deviation (Std), sensitivity (SE), specificity (SP), precision (Pre), and F-measure ($F1$)) reveal markedly the EHBCS is superior to BGA and BPSO. However, the algorithm still has shortcomings, such as increased computational complexity.

Future work requires further modification of the proposed algorithm to make it suitable for feature selection of multiclass datasets and to evaluate the results using different datasets and classification models.

Data Availability

The data are available at the dataset site: http://archive.ics.uci.edu/mlhttp://portals.broadinstitute.org/cgi-bin/cancer/datasets.cgihttp://csse.szu.edu.cn/staff/zhuzx/Datasets.html.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Z. Q. Li, J. Q. Du, B. Nie, W. Xiong, C. Hung, and H. Li, "Summary of feature selection methods," *Computer Engineering and Application*, vol. 55, no. 24, pp. 10–19, 2019.

[2] H. Liu and R. Setiono, "A probabilistic approachto feature selection: a filter solution," in *13th International Conference on Machine Learning (ICML'96)*, San Francisco, CA, USA, 1996.

[3] H. Liu, M. C. Zhou, and Q. Liu, "An embedded feature selection method for imbalanced data classification," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 3, pp. 703–715, 2019.

[4] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence Journal*, vol. 97, no. 1-2, pp. 273–324, 1997.

[5] L. Y. Chuang, S. W. Tsai, and C. H. Yang, "Improved binary particle swarm optimization using catfish effect for feature selection," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12699–12707, 2011.

[6] W. H. Bangyal, J. Ahmad, I. Shafi, and Q. Abbas, "A forward only counter propagation network-based approach for contraceptive method choice classification task," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 24, no. 2, pp. 211–218, 2012.

[7] Q. F. Zhao and Y. L. Zhang, "Ensemble method of feature selection and reverse construction of gene logical network based on information entropy," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 34, no. 2, article 2059004, 2020.

[8] J. X. Liu and Y. L. Zhang, "An attribute-weighted Bayes classifier based on asymmetric correlation coefficient," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 34, no. 10, article 2050025, 2020.

[9] Y. L. Zhang, T. Feng, S. Wang et al., "A novel XGBoost method to identify cancer tissue-of-origin based on copy number variations," *Frontiers in genetics*, vol. 11, pp. 585029–585029, 2020.

[10] D. Rodrigues, L. A. M. Pereira, T. N. S. Almeida et al., "BCS: a binary cuckoo search algorithm for feature selection," in *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 465–468, Beijing, 2013.

[11] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, MIT Press, Cambridge, MA, USA, 1992.

[12] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IProceedings of ICNN'95-International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Perth, WA, Australia, 1995.

[13] W. H. Bangyal, J. Ahmad, and H. T. Rauf, "Optimization of neural network using improved bat algorithm for data classification," *Journal of Medical Imaging and Health Informatics*, vol. 9, no. 1, pp. 669–680, 2019.

[14] M. Junaid, W. H. Bangyal, and J. Ahmad, "A novel Bat Algorithm using sobol sequence for the initialization of population," in *2020 IEEE 23rd International Multitopic Conference (INMIC)*, pp. 1–6, Bahawalpur, Pakistan, 2020.

[15] X. S. Yang and S. Deb, "Cuckoo search via Levy flights," in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pp. 210–214, Coimbatore, 2009.

[16] Y. Liu, L. Xueming, W. Zhang, J. Peng, X. Liao, and W. U. Zhongfu, "Feature subset selection based on genetic algorithm," *Computer Engineering*, vol. 29, no. 6, pp. 19-20, 2003.

[17] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection," *Pattern Recognition Letters*, vol. 10, no. 5, pp. 335–347, 1989.

[18] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, vol. 5, pp. 4104–4108, Orlando, FL, USA, 1997.

[19] H. A. Firpi and E. Goodman, "Swarmed feature selection," *International Symposium on Information Theory*, vol. 1, pp. 112–118, 2004.

[20] X. S. Yang and S. Deb, "Cuckoo search: recent advances and applications," *Neural Computing and Applications*, vol. 24, no. 1, article 1367, pp. 169–174, 2014.

[21] P. Civicioglu and E. Besdok, "A conceptual comparison of the cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms," *Artificial Intelligence Review*, vol. 39, no. 4, article 9276, pp. 315–346, 2013.

[22] E. Valian, S. Mohanna, and S. Tavakoli, "Improved cuckoo search algorithm for feed forward neural network training," *International Journal of Artificial Intelligence & Applications*, vol. 2, no. 3, pp. 36–43, 2011.

[23] R. A. Vazquez, "Training spiking neural models using cuckoo search algorithm," in *2011 IEEE Congress of Evolutionary Computation (CEC)*, pp. 679–686, New Orleans, LA, 2011.

[24] O. Pauline, "Adaptive cuckoo search algorithm for unconstrained optimization," *The Scientific World Journal*, vol. 2014, Article ID 943403, 8 pages, 2014.

[25] D. FENG, Q. RUAN, and L. du, "Binary cuckoo search algorithm," *Journal of Computer Applications*, vol. 33, no. 6, pp. 1566–1570, 2013.

[26] A. Gherboudj, A. Layeb, and S. Chikhi, "Solving 0-1 knapsack problems by a discrete binary version of cuckoo search algorithm," *International Journal of Bio Inspired Computation*, vol. 4, no. 4, pp. 229–236, 2012.

[27] L. A. M. Pereira, D. Rodrigues, T. N. S. Almeida et al., "A Binary Cuckoo Search and Its Application for Feature Selection," in *Cuckoo Search and Firefly Algorithm*, vol. 516 of Studies in Computational Intelligence, , pp. 141–154, Springer International Publishing, 2014.

[28] K. K. Bhattacharjee and S. P. Sarmah, "A binary cuckoo search algorithm for knapsack problems," in *2015 International Conference on Industrial Engineering and Operations Management (IEOM)*, pp. 1–5, Dubai, United Arab Emirates (UAE), 2015.

[29] M. N. Sudha and S. Selvarajan, "Feature selection based on enhanced cuckoo search for breast cancer classification in mammogram image," *Circuits and Systems*, vol. 7, no. 4, pp. 327–338, 2016.

[30] M. A. E. Aziz and A. E. Hassanien, "Modified cuckoo search algorithm with rough sets for feature selection," *Neural Computing and Applications*, vol. 29, no. 4, article 2473, pp. 925–934, 2018.

[31] S. Salesi and G. Cosma, "A novel extended binary cuckoo search algorithm for feature selection," in *2017 2nd International Conference on Knowledge Engineering and Applications (ICKEA)*, pp. 6–12, London, 2017.

[32] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Proceedings of the Ninth International Workshop on Machine Learning (ML 1992)*, pp. 249–256, Aberdeen, Scotland, UK, 1992.

[33] M. Lichman, "UCI machine learning repository," 2020, http://archive.ics.uci.edu/ml/.

[34] Medulloblastomas, "Cancer program datasets," 2020, http://portals.broadinstitute.org/cgi-bin/cancer/datasets.cgi/.

[35] High-dimentional datasets, "Microarray Datasets in Weka ARFF format," 2020, http://csse.szu.edu.cn/staff/zhuzx/Datasets.html/.

[36] S. Yadav and S. Shukla, "Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification," in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, pp. 78–83, Bhimavaram, 2016.

[37] G. Magna, P. Casti, S. V. Jayaraman et al., "Identification of mammography anomalies for breast cancer detection by an ensemble of classification models based on artificial immune system," *Knowledge Based Systems*, vol. 101, pp. 60–70, 2016.

[38] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *ACM Fifth Annual Workshop on Computational Lerning Theory*, pp. 144–152, Pittsburgh, 1992.