Applied Soft Computing Journal 93 (2020) 106402

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

Feature selection based on improved binary global harmony search for data classification

Jafar Gholami [a], Farhad Pourpanah [b], Xizhao Wang [c,*]

[a] Department of Computer Engineering, Kermanshah Science and Research Branch, Islamic Azad University, Kermanshah, Iran
[b] College of Mathematics and Statistics, Shenzhen University, China
[c] College of Computer Science and Software Engineering, Guangdong Key Lab. of Intelligent Information Processing, Shenzhen University, China

ARTICLE INFO

Article history:
Received 16 February 2020
Received in revised form 20 April 2020
Accepted 12 May 2020
Available online 22 May 2020

Keywords:
Feature selection
Population-based optimization
Binary harmony search
Data classification

ABSTRACT

Harmony search (HS) is an effective meta-heuristic algorithm inspired by the music improvisation process, where musicians search for a pleasing harmony by adjusting their instruments' pitches. The HS algorithm and its variants have been widely used to solve binary and continuous optimization problems. In this paper, we propose an improved binary global harmony search algorithm, called IBGHS, to undertake feature selection problems. A modified improvisation step is introduced to enhance the global search ability and increase the convergence speed of the algorithm. In addition, the K-nearest neighbor (KNN) is used as an underlying learning model to evaluate the effectiveness of the selected feature subsets. The experimental results on eighteen benchmark problems indicate that the proposed IBGHS algorithm is able to produce comparable results as compared with other state-of-the-art population-based methods such as genetic algorithm (GA), particle swarm optimization (PSO), antlion optimizer (ALO), novel global harmony search (NGHS) and whale optimization algorithm (WOA) in solving feature selection problems.

1. Introduction

Feature selection (FS) is an optimization problem that plays an important role in tackling classification problems. It is a process of selecting an optimal subset of features from a data set so that the classifier can obtain better accuracy and/or reduce the computational burden. Nonetheless, removing irrelevant features is a challenging issue and time consuming owing to a large search space and wrapped relationship between the features [1–4].

FS techniques can be grouped into filter-, embedded- and wrapper-based methods [5]. The filter-based methods use the properties of the learning samples, such as distance and similarity, during the FS process [6]. Embedded-based methods search for the best feature subset during the training process, in order to reduce the computational burden [7]. While, wrapper-based methods use a classification algorithm to evaluate the quality of the various feature subsets, and a search mechanism to find the optimal ones. Among them, wrapper-based methods are more effective since they use a classifier to operate as a feedback mechanism to compute the fitness value of the selected feature subsets, but they are computationally expensive [8].

Traditional wrapper-based FS methods, such as sequential backward selection (SBS) [9] and sequential forward selection (SFS) [10], improve the performance of the learning model via sequentially adding or removing features from data set. In these methods, once features are removed or added, they cannot be updated in the next steps. Later, this problem was solved by integrating a floating technique into SBS and SFS [11]. However, they suffer from nesting effects and computationally expensive [12]. To alleviate these problems, population-based optimization algorithms, such as particle swarm optimization (PSO) [13–15], genetic algorithm (GA) [16–18], genetic programming (GP) [19,20], ant colony optimization (ACO) [21], brain storm optimization (BSO) [22,23] and harmony search (HS) [24], have been used. These algorithms start with a set of randomly generated solutions, and use a fitness function to evaluate them. Then, they generate new solutions based on the individuals that performed better in the previous iteration. As a result, these algorithms reduce the computational burden as they avoid generating new individuals similar to the low quality ones.

Among them, harmony search (HS) [25] is an effective meta-heuristic algorithm inspired by the music improvisation process of probing for a better state of harmony. HS has been widely applied to solve real-world optimization problems, such as control system [26] and financial management [27], due to its simple structure, easy to implement and less parameters [28]. However, the basic HS algorithm suffers from several limitations such as

---

* Corresponding author.
E-mail addresses: gholami2018@gmail.com (J. Gholami), farhad@szu.edu.cn (F. Pourpanah), xizhaowang@ieee.org (X. Wang).

https://doi.org/10.1016/j.asoc.2020.106402
1568-4946/

pre-mature convergence and not using the advantageous of the current best solution to move toward the best position. To alleviate these issues, many enhanced variants of the HS algorithm have been developed. In [29], an improved HS (IHS) algorithm with adaptive pitch tuning strategy was developed. The global-best HS (GHS) algorithm [30], which is inspired by the PSO, used the advantageous of the global best solution to move toward the best position. Later, novel global harmony search (NGHS) algorithm [31,32] was proposed to tackle reliability problems. The NGHS algorithm introduced a new updating mechanism to enable worst harmony categories move toward the best position and used a genetic mutation probability to jump out of local optimum. The IAHS algorithm [33], i.e., an improved adaptive harmony search, was proposed to solve the resource leveling problem.

Besides, several studies have been focused on developing binary HS algorithms. The BHS algorithm [34] integrated the binary-coding into the basic HS algorithm to solve water pump switching problem. In [35], a modified NGHS algorithm was introduced to solve the 0–1 knapsack problems. In [36], a simplified binary harmony search (SBHS) algorithm was presented to solve high-dimensional 0–1 knapsack problems. The ABHS algorithm [37], i.e., adaptive binary harmony search, introduced a new pitch adjustment role to effectively probe the search space.

This paper proposes an improved binary global harmony search (IBGHS) algorithm to undertake the FS problems. We introduce an efficient improvisation step that is able to effectively search the search space, and escape from local optimum. In addition, the K-nearest neighbor (KNN) is used as the underlying learning model to evaluate the effectiveness of the selected feature subsets. The IBGHS algorithm is evaluated using eighteen benchmark problems from the UCI machine leaning repository [38]. To sum up, our main contributions include:

- an improved binary global harmony search, i.e., IBGHS, algorithm with the modified improvisation step;
- demonstration of the effectiveness of the IBGHS algorithm as a FS technique using eighteen benchmark problems.

This paper consists of five sections. Section 2 provides a review of population-based FS methods for data classification problems. In Section 3, firstly, the structures of harmony search and novel global harmony search algorithms are presented. Then, the proposed improved binary harmony search algorithm is discussed in detail. Section 4 provides the experimental results and discussion. Finally, Section 5 concludes the paper and gives suggestion for future work.

2. Related work

In this section, we briefly give a review of population-based FS methods. The GA was the first that has been used to undertake FS problems [39]. In [40], the local search (LS) strategy was embedded in GA to handle FS problems. In [41,42], a hybrid model of Q-learning fuzzy ARTMAP (QFAM) [43] and GA, i.e., QFAM-GA, was proposed for solving FS and classification problems. Firstly, QFAM was used as the underlying learning model to create a number of prototype nodes. Then, the GA was adopted to select an optimal feature subset from the created prototype nodes. Later, QFAM-GA was used to recognize human motion recognition in [44]. The improved binary genetic algorithm with feature granulation (IBGAFG) [45], which is a hybrid model of GA and granulation information, was proposed to undertake FS and optimization problems.

Chaotic binary PSO (CBPSO) [46] integrated two types of chaotic maps, i.e., logistic and tents maps, in a binary PSO (BPSO) for solving FS and classification problems. CBPSO used the chaotic maps to compute the inertia weight of the BPSO. LFPSO [47] combined PSO with the lévy flight (LF) strategy to help PSO jump out of local optimum. HPSO-LS [48], which is a hybrid model of PSO and LS strategy, employed LS to help PSO during the FS process using the correlation information of features. HPSO-SSM [49], namely a hybrid model of PSO and spiral-shaped mechanism (SSM), was presented for FS. It used the logistic map sequence to increase the diversity and introduced two new update mechanisms to improve the quality of the movement. In addition, SSM was adopted to locally search around the known optimal position.

BAFS [50] is a bacterial algorithm (BA)-based FS method for classification. A new parameters and update mechanism were introduced to reduce the execution duration of the model. In [51], two hybrid models of whale optimization algorithm (WOA) and simulated annealing (SA) were proposed to improve exploitation ability of the WOA. The first model integrated the SA into the WOA, while the second model added the SA at the end of each iteration to further improve the best solution found by the WOA. In [52], a FS method based on grasshopper optimization algorithm was developed to solve classification problems. A modified ALO (MALO) [53] was proposed as an effective FS method for hyperspectral image classification. MALO used the LF strategy to increase the global search ability of the model. In addition, the wavelet SVM was used as an underlying learning model to evaluate the effectiveness of the selected feature subsets. In [54], an improved Salp swarm Algorithm (ISSA) was introduced for FS. ISSA used opposition based learning (OBL) at the initialization phase and LS strategy to improve the population diversity and exploitation capability of the model.

In [55], a hybrid model of ABC and differential evolution (DE) was proposed for solving FS and classification problems. A new neighborhood search strategy and a new mutation operator were, respectively, introduced for ABC and DE algorithms to find optimal feature subsets. In [56], a hybrid model of DE and SVM was proposed for FS and classification of the EEG signals. MOFS-BDE [57], which is a multi-objective FS method based on binary DE algorithm, was proposed for classification. In this work, a new probability-based binary mutation operator, known as one-bit purifying search (OPS) operator, was introduced to improve the self-learning ability of the individuals.

A novel HS algorithm with adaptive parameter tuning scheme was proposed for FS [58]. SAHS [59], which is a self-adaptive harmony search-based FS method, was developed to solve the music genre-classification problems. In [60], a HS algorithm with modified improvisation step was proposed for FS to detect epileptic seizure. MRMR-COA-HS [61] is a two-stage gene selection method for cancer classification based on minimum redundancy and maximum relevance (MRMR) strategy, cuckoo optimization algorithm (COA) and HS algorithm. Firstly, the relevant genes were selected using the MRMR strategy. Then, the selected genes were fed into a wrapper-based FS method where COA-HS and SVM were used as an underlying search strategy and classification model, respectively.

3. The structure and dynamics of HS, NGHS and IBHGS models

In this section, firstly, we describe the structures of the HS and NGHS algorithms in detail. Then the details of the proposed improved binary harmony search algorithm is explained.

3.1. Harmony search algorithm

The HS algorithm represents each solution, which is called a “harmony”, by an N-dimensional input vector. The basic HS algorithm has three main phases: *(i)* initialization, (ii) improvisation of a harmony vector, and *(iii)* updating harmony memory

(HM). Firstly, HS algorithm generates *HMS* (harmony memory size) random harmony vectors and stores them in a HM. Then, a new candidate harmony is generated based on all individuals in HM by performing memory contraction and pitch adjustment rules, and random re-initialization. Finally, replacement takes place if the new candidate harmony is better than the worst harmony category in the HM. This search cycle is repeated until the termination condition is met. The detailed steps of the basic HS algorithm are as follows:

Step 1: Initialize parameters: Harmony memory size (*HMS*), number of decision variables (*N*), lower and upper bounds (*LB* and *UB*), harmony memory consideration rate (*HMCR*), pitch adjusting rate (*PAR*), bandwidth vector (*BW*), and maximum number of iteration (*NI*).

Step 2: Initialize the HM: The HS algorithm uses a uniform distribution to initialize the HM within the lower and upper bounds of the search space, as follows:

$$HM = \begin{bmatrix} x_{(1,1)} & x_{(1,2)} & \cdots & x_{(1,N)} \\ x_{(2,1)} & x_{(2,2)} & \cdots & x_{(2,N)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{(HMS,1)} & x_{(HMS,2)} & \cdots & x_{(HMS,N)} \end{bmatrix} \quad (1)$$

Step 3: Improvise a harmony: The process of generating new harmony is known as improvisation. In this step, a new harmony $X^{new} = (x_1^{new}, x_2^{new}, \ldots, x_N^{new})$ is generated based on HM consideration, pitch adjustment and a random choice. Details of the improvisation step is shown in Algorithm 1.

Algorithm 1: Improvisation step of the HS algorithm.

```
for each $i \in [1, N]$ do
    if $rand() \le HMCR$ then
        $x_i^{new} = x_{(j,i)}$ $(j = 1, 2, ..., HMS)$ % memory consideration
        if $rand() \le PAR$ then
            $x_i^{new} = x_{(j,i)} \pm r \times bw$ % pitch adjustment
            if $x_i^{new} > UB$ then
                $x_i^{new} = UB$
            if $x_i^{new} < LB$ then
                $x_i^{new} = LB$;
    else
        $x_i^{new} = LB + rand() \times (UB - LB)$ % random selection
```

where $x_{(j,i)}$ $j = 1, 2, \ldots, HMS$ is the *i*th element of *j*th harmony candidate in HM, *rand*() and *r* are random values between 0 and 1.

Step 4: Update HM: If the improvised harmony vector is better than the worst harmony category in the HM, replacement takes place.

Step 5: Check the termination criterion: If the termination condition is met stop, otherwise go to Step 3.

3.2. Novel global harmony search algorithm

Inspired by the PSO, the novel global harmony search (NGHS) algorithm [31] model modified the improvisation step in the original HSA. This modification allowed new individuals to mimic the successful companion, i.e., global-best, in the HM. The NGHS algorithm excluded HMCR and PAR, and introduced the genetic mutation probability (p_m). In addition, it replaces the worst harmony category in the HM with the new harmony without checking their fitness values. The step-by-step improvisation of the NGHS algorithm is summarized in Algorithm 2.

Algorithm 2: Improvisation step the NGHS [31] algorithm.

```
for each $i \in [1, N]$ do
    $x_R = 2 \times x_i^{best} - x_i^{worst}$.
    if $x_R > UB$ then
        $x_R = UB$
    if $x_R < LB$ then
        $x_R = LB$
    $x_i^{new} = x_i^{worst} + r \times (x_R - x_i^{worst})$ % position updating
    if $rand() \le p_m$ then
        $x_i^{new} = LB + rand() \times (UB - LB)$ % genetic mutation
```

Where x_i^{best} and x_i^{worst} are the best and worst harmonies in the HM, respectively.

3.3. The improved binary global harmony search algorithm

Although, the NGHS algorithm is able to produce better results as compared with the original HS algorithm, it suffers from two limitations: *(i)* trapping into local optima, and *(ii)* slow convergence. These are mainly due to the tendency of the new harmony categories to move toward the global-best harmony category in the HM, consequently, they do not effectively probe the entire search space. To alleviate the aforementioned limitations of NGHS, we proposed an improved binary global harmony search (IBGHS) algorithm, as follows.

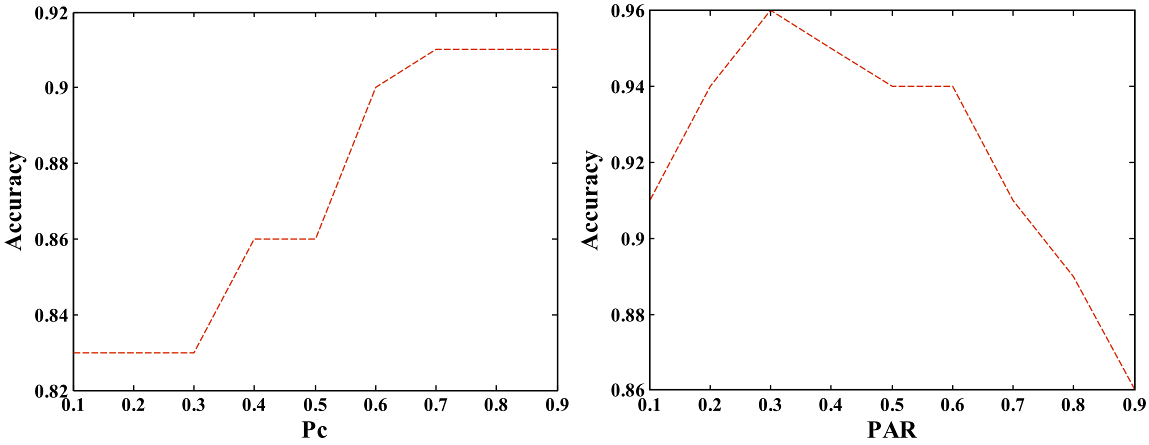
The proposed IBGHS algorithm is a binary variant of the NGHS algorithm with modified improvisation step. As such, the IBGHS algorithm generates *NHS* random binary harmony vectors, and stores them in the HM. Then, a new harmony is generated based on the modified improvisation step, as shown in Algorithm 3.

Algorithm 3: Improvisation step of the proposed IBGHS algorithm.

```
1  for each $i \in [1, N]$ do
2      if $rand() \le P_c$ then
3          $x_R = 2 \times x_i^{best} - x_i^{worst}$.
4          if $x_R > UB$ then
5              $x_R = UB$
6          if $x_R < LB$ then
7              $x_R = LB$
8          $x_i^{new} = x_i^{worst} + r_1 \times (x_R - x_i^{worst})$ % position updating
9          if $rand() \le p_m$ then
10             $x_i^{new} = LB + rand() \times (UB - LB)$ % genetic mutation
11     else
12         $x_i^{new} = 0.7x_i^{best} + 0.3x_i^{worst}$
13         if $rand() \le PAR$ then
14             $x_i^{new} = x_i^{new} + r_2 \times BW$
15             if $x_i^{new} > UB$ then
16                 $x_i^{new} = UB$
17             if $x_i^{new} < LB$ then
18                 $x_i^{new} = LB$
```

Where *rand*(), r_1 and r_2 are random values between 0 and 1, and P_c and *PAR* are control parameter and pitch adjustment rate, respectively.

In order to enhance the global search ability and convergence speed of the NGHS algorithm, lines 12–18 are added. Specifically,



(a) (b)

Fig. 1. Accuracy rates of IBGHS with different *(a)* P_c, and *(b)* *PAR* settings for Penglung EW data set.

line 12 allows the IBGHS algorithm to move faster toward the global best solution, while line 14 allows the algorithm to randomly search the entire search space. The effectiveness of this modification will be discussed in Section 4.2. After generating a new harmony, its fitness value is calculated and compared with the worst harmony in the HM. Replacement takes place if the new harmony is outperformed the worst one. This process is repeated till the termination condition is met. Algorithm 4 summaries the step-by-step procedure of IBGHS.

Algorithm 4: The IBGHS algorithm procedure.

```
Input: Parameters of the IBGHS algorithm;
Output: An optimal harmony category;
Generate NHS random harmonies and store them in HM;
Calculate the fitness value of all harmonies;
while the termination condition is not met do
    for each harmony in HM do
        Generate a new harmony using Algorithm 3;
        Calculate the fitness value of the new harmony;
        if new harmony is better than the worst harmony in HH then
            Do replacement.
```

4. Experimental studies

The performance of the proposed IBGHS algorithm is evaluated using eighteen benchmark classification problems from the UCI machine leaning repository [38]. Table 1 summarizes the details of the used data sets in this study. These data sets contain various characteristics, such as level of difficulty, and numbers of samples and features, that are useful to assess the FS capability of IBGHS. In addition, these data sets are chosen to compare the performance of the proposed IBGHS algorithm with other population-based FS techniques. The K-nearest neighbor (KNN) (k=5) with Euclidean distance metric is used to evaluate the effectiveness of the selected feature subsets and find an optimal one [62]. The classification-error is used as fitness value. For all experiments, the K-fold cross-validation, i.e. data sets are split into K folds, then $k-1$ folds are used for training and the remaining for validation, is used and repeated 5 times. Data sets are split into two equal subsets, i.e., training and validation.

The experimental parameters of the IBGHS algorithm are set as follows. $PAR = 0.3$ and $P_c = 0.9$ are optimized in Section 4.1, $p_m = 0.005$ (adopted from [32]). In order to have a fair comparison with other algorithms reported in [51], *NI* and *HMS* are set to 100 and 10, respectively. All experiments are conducted using Matlab 2018a with core i7, 4 GHz CPU and 16 GB memory.

Table 1
Details of the UCI data sets.

Data set	No. of features	No. of samples
Breast cancer	9	699
Breast EW	30	569
Congress EW	16	435
Exactly	13	1000
Exactly2	13	1000
Heart EW	13	270
Ionosphere EW	34	351
Kr-vs-kp EW	36	3196
Lymphography	18	148
M-of-n	13	1000
Penglung EW	325	73
Sonar EW	60	208
Spect EW	22	267
Tic-tac-toe	9	958
Vote	16	300
Waveform	40	5000
Wine	13	178
Zoo	16	101

4.1. Parameters setting

This section aims to find optimal parameters, i.e., P_c and *PAR*, for the IBHGS algorithm that yields the best results. To achieve this, both P_c and *PAR* are varied from 0.1 to 0.9 for all data sets. In overall, the IBGHS algorithm achieves the best performance when P_c and *PAR* are set to 0.9 and 0.3, respectively. For instance, Fig. 1 shows the accuracy rates of IBGHS with different P_c and *PAR* settings for the Penglung EW data set. As can be seen, IBGHS produces highest accuracy rates when P_c is set between 0.7 and 0.9 (Fig. 1(a)), and $PAR = 0.3$ (Fig. 1(b)). Therefore, for the rest of the experiments P_c and *PAR* are set to 0.9 and 0.3, respectively.

4.2. Results and discussions

In this section, the performance of IBGHS is compared with NGHS and other population-based FS techniques such as GA, ALO and WOASAT-2 (a hybrid model of WOA and SA with tournament selection) [51] in terms of classification accuracy, number of selected features, computational time and convergence speed. To have a fair comparison with other techniques, the same procedure in [51], is adopted, i.e. for each data set, K-fold cross-validation is used and repeated 5 times. Table 2 shows the mean

Table 2
Mean accuracy rates for UCI data sets.

Data set	WOASAT-2	ALO	GA	PSO	NGHS	IBGHS
Breast cancer	0.97	0.96	0.96	0.95	0.96	**0.98**
Breast EW	**0.98**	0.93	0.94	0.94	0.95	0.97
Congress EW	**0.98**	0.93	0.94	0.94	0.96	**0.98**
Exactly	**1**	0.66	0.67	0.68	0.81	**1**
Exactly2	0.75	0.75	0.76	0.75	0.71	**0.80**
Heart EW	0.85	0.83	0.82	0.78	0.85	**0.89**
Ionosphere EW	0.96	0.87	0.83	0.84	0.94	**0.96**
Kr-vs-kp EW	**0.98**	0.96	0.92	0.94	0.94	**0.98**
Lymphography	0.89	0.79	0.71	0.69	0.85	**0.93**
M-of-n	**1**	0.86	0.93	0.86	0.97	**1**
Penglung EW	0.94	0.63	0.7	0.72	0.86	**1**
Sonar EW	**0.97**	0.74	0.73	0.74	0.90	**0.97**
Spect EW	0.88	0.8	0.78	0.77	0.88	**0.91**
Tic-tac-toe	0.79	0.73	0.71	0.73	**0.82**	**0.82**
Vote	0.97	0.92	0.89	0.89	0.96	**0.98**
Waveform	0.76	**0.77**	**0.77**	0.76	0.74	0.75
Wine	0.99	0.91	0.93	0.95	0.95	**1**
Zoo	0.97	0.91	0.88	0.83	0.94	**1**
Average	0.92	0.83	0.83	0.82	0.88	**0.94**

Table 3
Average number of selected features.

Data set	WOASAT-2	ALO	GA	PSO	NGHS	IBGHS
Breast cancer	**4.20**	6.28	5.09	5.72	5.40	4.80
Breast EW	**11.60**	16.08	16.35	16.56	14.00	12.20
Congress EW	6.40	6.98	6.62	6.83	7.80	**3.80**
Exactly	**6.00**	6.62	10.82	9.75	7.00	6.20
Exactly2	2.80	10.70	6.18	6.18	7.20	**1.20**
Heart EW	5.40	10.31	9.49	7.94	6.80	**3.80**
Ionosphere EW	12.8	9.42	17.31	19.18	14.60	**10.71**
Kr-vs-kp EW	18.40	24.7	22.43	20.81	19.40	**14.80**
Lymphography	7.20	11.05	11.05	8.98	10.80	**5.60**
M-of-n	**6.00**	11.08	6.83	9.04	7.60	**6.00**
Penglung EW	127.4	164.13	177.13	178.75	141.20	**81.60**
Sonar EW	26.40	37.92	33.30	31.20	29.20	**23.00**
Spect EW	9.40	16.15	11.75	12.50	11.40	**6.20**
Tic-tac-toe	6.00	6.99	6.85	6.61	6.60	**5.40**
Vote	5.20	9.52	6.62	8.80	6.80	**4**
Waveform	20.6	35.72	25.28	22.72	19.00	**16.8**
Wine	6.40	10.70	8.63	8.36	**6.20**	**6.20**
Zoo	5.60	13.97	10.11	9.74	7.40	**4.00**
Average	15.99	22.68	21.77	21.65	18.24	**12.00**

accuracy rates of IBGHS, NGHS, PSO, GA, ALO and WOASAT-2. As can be seen, IBGHS yields the highest accuracy rates for ten out of eighteen data sets, including Breast cancer, Exactly 2, Heart EW, Ionosphere EW, Lymphography, Penglung EW, Spect EW, Vote, Wine Ew and Zoo. While IBGHS does not outperform other methods for Breast EW and Waveform EW data sets, it produces similar results as WOASAT-2 for Congress EW, Exactly, Kr-vs-kp EW, M-of-n and Sonar EW data sets. Overall, IBGHS outperformed other FS techniques in term mean accuracy rate, i.e., 94%, for all data sets.

Table 3 shows the average number of selected features of IBGHS, NGHS, PSO, GA, ALO and WOASAT-2. IBGHS selects fewer numbers of features for Congress EW, Exactly 2, Heart EW, Ionosphere EW, Kr-vs-kp EW, Lymphography, Penglung EW, Sonar EW, Spec EW, Tic-Tac-Toe, Vote, Waveform EW, Wine and Zoo data sets. It also selects similar numbers of features to WOASAT-2 and NGHS for M-of-n and Wine, respectively. While IBGHS is ranked second best performance for Breast cancer, Breast EW and Exactly data set. Overall, IBGHS selected fewer number of features, i.e., 12.00, for all data sets as compared with other FS techniques.

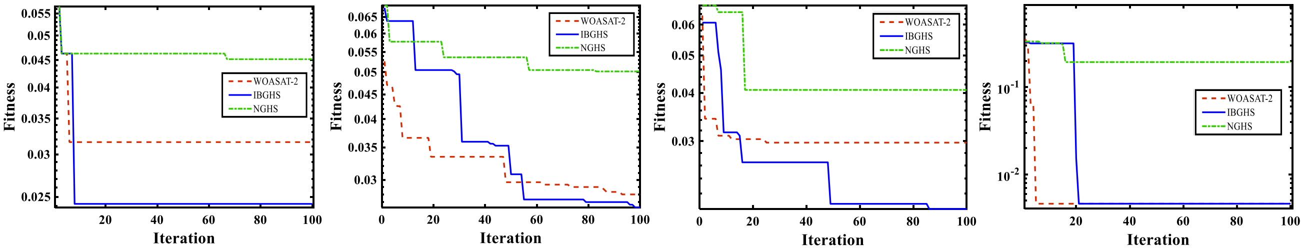
The obtained mean, best and worst fitness values for the UCI data sets are shown in Table 4. As can be seen, IBGHS achieves better and similar mean fitness values for eleven and three data sets, respectively. In term of best fitness value, IBHGS outperforms other FS techniques for seven data sets, and produces similar results for eight data sets. While NGHS produces the worst fitness values for most of the data sets. To sum up, IBGHS and WOASAT-2, respectively, ranked first and second best methods in terms of mean, best and worst fitness values for all data sets.

Table 5 shows the execution time (in second) of WOASAT-2, ALO, GA, PSO, NGHS and IBHGS algorithms for the UCI data sets. All algorithms use the same data sets, the number of iteration and population size. As can be seen, both IBGHS and NGHS algorithms require the shortest execution time as compared with other algorithms. Specifically, the IBHGS algorithm has the best computational time for eleven out of eighteen data sets, i.e. Breast cancer, Congress EW, Exactly, Exactly2, Heart EW, Kr-vs-kp EW, Penglung EW, Spect EW, Waveform EW, Wine EW and Zoo. While IBGHS needs longer computational time for Breast EW, Ionosphere EW, Lymphography and Tic-tac-toe data sets than NGHS, it requires similar computational time as NGHS for M-of-n, Sonar EW and Vote data sets. In contrast, WOASAT-2 requires the longest computational time for all data sets.

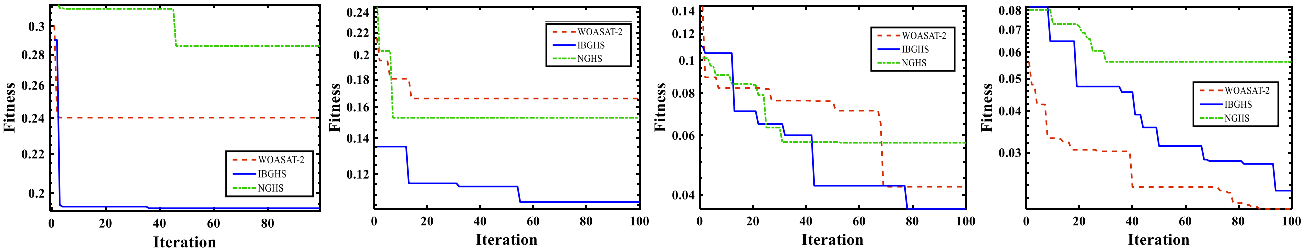
Finally, convergence of IBGHS, NGHS and WOASAT-2 is compared. Fig. 2 shows the fitness values of IBGHS, NGHS and WOASAT-2 with different iteration numbers. As can be seen, the NGHS algorithm traps into local optimum for most of the data

Table 4
Mean, worst and best fitness values for UCI data sets.

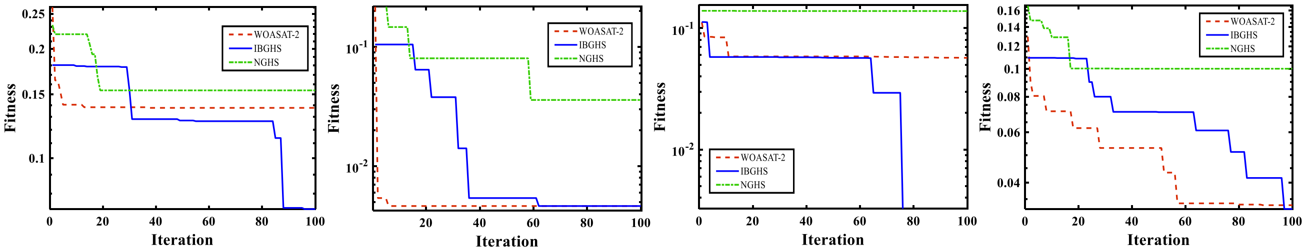
Data set	Mean						Best						Worst					
	WOASAT-2	ALO	GA	PSO	NGHS	IBGHS	WOASAT-2	ALO	GA	PSO	NHGS	IBGHS	WOASAT-2	ALO	GA	PSO	NGHS	IBGHS
Breast cancer	0.04	**0.02**	0.03	0.03	0.04	0.03	0.03	**0.02**	**0.02**	0.03	0.03	**0.02**	0.04	0.03	0.04	0.03	**0.05**	0.04
Breast EW	0.03	0.03	0.04	0.03	0.05	**0.02**	**0.02**	0.03	**0.02**	**0.02**	0.05	**0.02**	0.04	0.04	0.05	0.05	**0.06**	0.03
Congress EW	0.03	0.05	0.04	0.04	0.05	**0.02**	0.02	0.03	0.03	0.03	0.04	**0.01**	0.05	**0.06**	**0.06**	0.04	0.05	0.02
Exactly	0.01	0.29	0.28	0.28	0.28	**0**	0.01	0.28	0.27	0.21	0.19	**0**	0.01	0.29	0.31	**0.32**	**0.32**	0
Exactly2	0.25	0.24	0.25	0.25	0.31	**0.2**	0.23	0.23	0.22	0.22	0.28	**0.19**	0.27	0.25	0.3	0.31	**0.33**	0.22
Heart EW	0.16	**0.12**	0.14	0.15	0.32	**0.12**	0.13	0.11	0.12	0.13	0.15	**0.10**	0.18	0.13	0.14	0.18	**0.79**	0.13
Ionosphere EW	**0.04**	0.11	0.13	0.14	0.20	0.05	**0.03**	0.1	0.09	0.12	**0.05**	0.03	0.05	0.12	0.16	**0.17**	0.15	0.06
Kr-vs-kp EW	**0.02**	0.05	0.07	0.05	0.09	**0.02**	**0.02**	0.03	0.03	0.03	0.05	**0.02**	0.02	0.07	0.13	0.07	**0.15**	0.03
Lymphography	0.11	0.14	0.17	0.19	0.22	**0.08**	0.09	0.08	0.12	0.14	0.15	**0.07**	0.14	0.16	**0.27**	**0.27**	**0.27**	0.09
M-of-n	0.01	0.11	0.08	0.11	0.17	**0**	0.01	0.09	0.02	0.06	0.03	**0**	0.01	0.12	0.15	0.16	**0.26**	0
Penglung EW	0.06	0.14	0.22	0.22	0.16	**0.02**	0.03	**0**	0.13	0.13	0.13	**0**	0.11	0.21	**0.29**	**0.29**	0.19	0.05
Sonar EW	**0.03**	0.18	0.13	0.13	0.12	0.05	**0.01**	0.13	0.07	0.07	0.10	0.03	0.05	**0.26**	0.23	0.22	0.13	0.06
Spect EW	0.13	0.12	0.14	0.13	0.17	**0.11**	0.11	**0.09**	0.12	0.1	0.11	**0.09**	0.15	0.15	0.15	0.16	**0.20**	0.12
Tic-tac-toe	0.21	0.22	0.24	0.24	0.22	**0.19**	0.2	0.2	0.21	0.21	0.18	**0.17**	0.23	0.24	0.26	**0.27**	0.24	0.21
Vote	0.04	0.04	0.05	0.05	0.05	**0.02**	**0.02**	0.03	0.03	0.03	0.04	**0.02**	0.04	0.05	**0.08**	**0.08**	0.07	0.02
Waveform EW	0.25	**0.21**	0.2	0.22	0.26	0.24	0.23	**0.19**	**0.19**	0.21	0.25	0.24	0.26	0.22	0.21	0.23	**0.27**	0.25
Wine EW	**0.01**	0.02	**0.01**	0.02	0.05	0.01	**0**	**0**	**0**	**0**	0.04	**0**	0.03	0.03	0.03	0.03	**0.06**	0.01
Zoo	0.04	0.07	0.08	0.1	0.07	**0**	**0**	0.04	**0**	0.03	0.06	**0**	0.1	0.12	0.18	**0.21**	0.08	0.02
Total	1.83	2.15	2.3	2.39	2.73	**1.18**	1.57	1.7	1.69	1.77	1.93	**1.01**	2.14	2.54	3.03	3.08	**3.67**	1.36



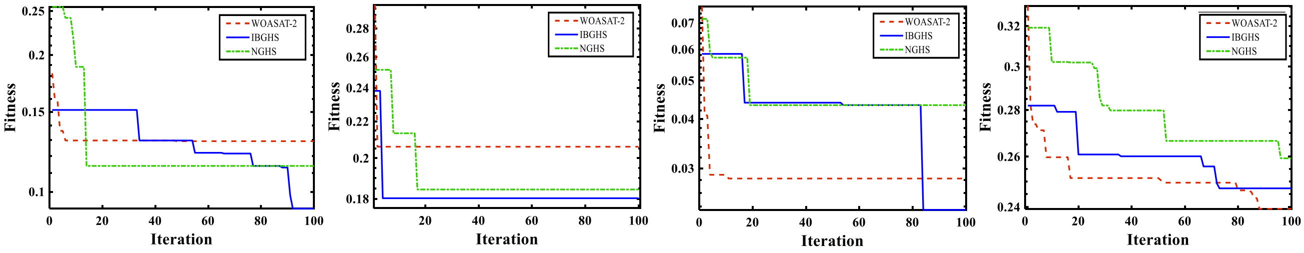
(a) Breast cancer (b) Breast EW (c) Congress EW (d) Exactly



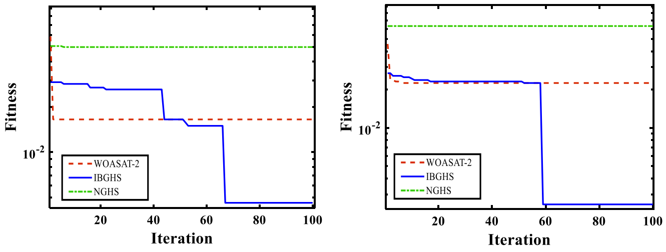
(e) Exactly 2 (f) Heart EW (g) Ionosphere EW (h) Kr-vs-Kp Ew



(i) Lymphography EW (j) M-of-n (k) Penglung EW (l) Sonar Ew



(m) Spect Ew (n) Tic-tac-toe (o) Vote (p) Waveform Ew



(q) Wine Ew (r) Zoo

Fig. 2. The convergence of the IBGHS, NGHS and WOASAT-2 algorithms for UCI data sets.

Table 5
Computational time (in second) of WOASAT-2, ALO, GA, PSO, NGHS and IBHGS algorithms for the UCI data sets.

Dataset	WOASAT-2	ALO	GA	PSO	NGHS	IBGHS
Breast cancer	16.32	6.94	3.59	1.57	0.45	**0.44**
Breast EW	17.29	6.82	3.53	1.54	**0.44**	0.50
Congress EW	14.10	7.26	3.76	1.64	0.47	**0.42**
Exactly	20.31	10.31	5.34	2.33	0.67	**0.52**
Exactly2	24.19	9.70	5.03	2.19	0.63	**0.48**
Heart EW	11.96	6.24	3.23	1.41	0.40	**0.38**
Ionosphere EW	13.68	6.50	3.37	1.47	**0.42**	0.44
Kr-vs-kp EW	243.13	60.09	31.14	13.58	3.88	**3.84**
Lymphography	11.34	6.32	3.27	1.43	**0.41**	0.43
M-of-n	20.87	9.65	5.00	2.18	**0.62**	**0.62**
Penglung EW	11.94	8.64	4.48	1.95	0.56	**0.41**
Sonar EW	11.22	5.99	3.10	1.35	**0.39**	**0.39**
Spect EW	11.72	6.71	3.48	1.52	0.43	**0.42**
Tic-tac-toe	19.02	10.63	5.51	2.40	**0.50**	0.69
Vote	11.29	6.00	3.11	1.36	**0.39**	**0.39**
Waveform EW	679.45	170.57	88.40	38.54	11.02	**8.47**
WineEW	10.77	5.85	3.03	1.32	0.38	**0.37**
Zoo	10.59	6.66	3.45	1.50	0.43	**0.40**

sets, as the fitness values remain constant after some iterations. This is mainly because of NGHS that tends to move toward the best harmony category, i.e., x_i^{best}, found so far. While, IBGHS alleviate this issue by generating new harmony categories within x_i^{best} and x_i^{worst} or randomly within the search space. This helps IBGHS to escape from the local optimum, and converge faster than NGHS toward the best solution. In addition, IBGHS operates better or similar to the WOASAT-2 algorithm.

Overall, the IBGHS algorithm can produce better results as compared with NGHS and other algorithms reported in the literature. This is mainly because of the modified improvisation step that help the algorithm to jump out of local optimum and converge fast toward the best solution. However, it contains more parameters to be optimized as compared with the NGHS algorithm.

5. Summary

In this paper, a FS method known as IBGHS, which is an improved binary global harmony search algorithm, for solving data classification problems has been developed. Specifically, a modified improvisation step was introduced that can increase the convergence speed and avoid trapping into local optimum. Eighteen benchmark problems have been conducted to evaluate the effectiveness of the IBGHS algorithm in terms of classification accuracy, number of selected features and computational time. The outcome indicates that the IBGHS algorithm is able to produce comparable results as compared with the basic HS, NGHS and other population based FS techniques such as GA, PSO, ALO and WOA. Our future work is focused on modifying the IBGHS algorithm to solve combinatorial optimization problems. In addition, it can be adapted to handle dynamic optimization problems.

CRediT authorship contribution statement

Jafar Gholami: Conceptualization, Methodology, Software, Data curation, Writing-Original draft. **Farhad Pourpanah:** Writing - review & editing, Visualization, Supervision. **Xizhao Wang:** Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is partially supported by the National Natural Science Foundation of China (Grant nos. (61976141 and 61732011)), and JCYJ20180305125850156.

References

[1] W.W. Ng, D.S. Yeung, M. Firth, E.C. Tsang, X. Wang, Feature selection using localized generalization error for supervised classification problems using RBFNN, Pattern Recognit. 41 (12) (2008) 3706–3719.

[2] C. Wang, Q. Hu, X. Wang, D. Chen, Y. Qian, Z. Dong, Feature selection based on neighborhood discrimination index, IEEE Trans. Neural Netw. Learn. Syst. 29 (7) (2018) 2986–2999.

[3] W. Zhang, X. Wang, Feature extraction and classification for human brain CT images, in: International Conference on Machine Learning and Cybernetics, Vol. 2, 2007, pp. 1155–1159.

[4] X. Wang, Y. Zhao, F. Pourpanah, Recent advances in deep learning, Int. J. Mach. Learn. Cybern. 11 (4) (2020) 747–750.

[5] A.K. Das, S. Sengupta, S. Bhattacharyya, A group incremental feature selection for classification using rough set theory based genetic algorithm, Appl. Soft Comput. 65 (2018) 400–411.

[6] B. Ma, Y. Xia, A tribe competition-based genetic algorithm for feature selection in pattern classification, Appl. Soft Comput. 58 (2017) 328–338.

[7] Y. Zhang, D. Gong, Y. Hu, W. Zhang, Feature selection algorithm based on bare bones particle swarm optimization, Neurocomputing 148 (2015) 150–157.

[8] F. Pourpanah, C.P. Lim, X. Wang, C.J. Tan, M. Seera, Y. Shi, A hybrid model of fuzzy min–max and brain storm optimization for feature selection and data classification, Neurocomputing 333 (2019) 440–451.

[9] T. Marill, D. Green, On the effectiveness of receptors in recognition systems, IEEE Trans. Inform. Theory 9 (1) (1963) 11–17.

[10] K.J. Wang, K.H. Chen, M.A. Angelia, An improved artificial immune recognition system with the opposite sign test for feature selection, Knowl.-Based Syst. 71 (2014) 126–145.

[11] P. Pudil, J. Novovičová, J. Kittler, Floating search methods in feature selection, Pattern Recognit. Lett. 15 (11) (1994) 1119–1125.

[12] E. Hancer, B. Xue, M. Zhang, D. Karaboga, B. Akay, Pareto front feature selection based on artificial bee colony optimization, Inform. Sci. 422 (2018) 462–479.

[13] M.Z. Moghadam, M. Abdar, Z. Davarzani, X. Zhou, P. Pławiak, U. Acharya, Hybrid particle swarm optimization for rule discovery in the diagnosis of coronary artery disease, Expert Syst. (2019) e12485.

[14] M. Abdar, V.N. Wijayaningrum, S. Hussain, R. Alizadehsani, P. Plawiak, U.R. Acharya, V. Makarenkov, IAPSO-AIRS: A novel improved machine learning-based system for wart disease treatment, J. Med. Syst. 43 (7) (2019) 220.

[15] A. Koohestani, M. Abdar, A. Khosravi, S. Nahavandi, M. Koohestani, Integration of ensemble and evolutionary machine learning algorithms for monitoring diver behavior using physiological signals, IEEE Access 7 (2019) 98971–98992.

[16] F. Pourpanah, C.P. Lim, J.M. Saleh, A hybrid model of fuzzy ARTMAP and genetic algorithm for data classification and rule extraction, Expert Syst. Appl. 49 (2016) 74–85.

[17] P. Pławiak, M. Abdar, J. Pławiak, V. Makarenkov, U.R. Acharya, DGHNL: A new deep genetic hierarchical network of learners for prediction of credit scoring, Inform. Sci. 516 (2020) 401–418.

[18] M. Hassoon, M.S. Kouhi, M.Z. Moghadam, M. Abdar, Rule optimization of boosted C5.0 classification using genetic algorithm for liver disease prediction, in: International Conference on Computer and Applications, 2017, pp. 299–305.

[19] D.P. Muni, N.R. Pal, J. Das, Genetic programming for simultaneous feature selection and classifier design, IEEE Trans. Syst. Man Cybern. B 36 (1) (2006) 106–117.

[20] M. Abdar, U.R. Acharya, N. Sarrafzadegan, V. Makarenkov, NE-nu-SVC: A new nested ensemble clinical decision support system for effective diagnosis of coronary artery disease, IEEE Access 7 (2019) 167605–167620.

[21] K.R. Robbins, W. Zhang, J.K. Bertrand, R. Rekaya, The ant colony algorithm for feature selection in high-dimension gene expression data for disease classification, Math. Med. Biol. J. IMA 24 (4) (2007) 413–426.

[22] F. Pourpanah, Y. Shi, C.P. Lim, Q. Hao, C.J. Tan, Feature selection based on brain storm optimization for data classification, Appl. Soft Comput. 80 (2019) 761–775.

[23] F. Pourpanah, R. Wang, X. Wang, Y. Shi, D. Yazdani, mBSO: A multi-population brain storm optimization for multimodal dynamic optimization problems, in: IEEE Symposium Series on Computational Intelligence, SSCI, 2019, pp. 673–679.

[24] A. Moayedikia, K.L. Ong, Y.L. Boo, W.G. Yeoh, R. Jensen, Feature selection for high dimensional imbalanced class data using harmony search, Eng. Appl. Artif. Intell. 57 (2017) 38–49.

[25] Z.W. Geem, J.H. Kim, G. Loganathan, A new heuristic optimization algorithm: Harmony search, SIMULATION 76 (2) (2001) 60–68.

[26] B.E. Sedhom, M.M. El-Saadawi, A.Y. Hatata, A.S. Alsayyari, Hierarchical control technique-based harmony search optimization algorithm versus model predictive control for autonomous smart microgrids, Int. J. Electr. Power Energy Syst. 115 (2020) 105511.

[27] R. Dash, P. Dash, R. Bisoi, A self adaptive differential harmony search based optimized extreme learning machine for financial time series prediction, Swarm Evol. Comput. 19 (2014) 25–42.

[28] S. Kulluk, L. Ozbakir, A. Baykasoglu, Training neural networks with harmony search algorithms for classification problems, Eng. Appl. Artif. Intell. 25 (1) (2012) 11–19.

[29] M. Mahdavi, M. Fesanghary, E. Damangir, An improved harmony search algorithm for solving optimization problems, Appl. Math. Comput. 188 (2) (2007) 1567–1579.

[30] M.G. Omran, M. Mahdavi, Global-best harmony search, Appl. Math. Comput. 198 (2) (2008) 643–656.

[31] D. Zou, L. Gao, J. Wu, S. Li, Y. Li, A novel global harmony search algorithm for reliability problems, Comput. Ind. Eng. 58 (2) (2010) 307–316.

[32] E. Valian, S. Tavakoli, S. Mohanna, An intelligent global harmony search approach to continuous optimization problems, Appl. Math. Comput. 232 (2014) 670–684.

[33] J.P. Tienda, A.S. Bernal, E. Pellicer, J.B. Marco, Improved Adaptive Harmony Search algorithm for the Resource Leveling Problem with minimal lags, Autom. Constr. 77 (2017) 82–92.

[34] Z.W. Geem, Harmony search in water pump switching problem, in: L. Wang, K. Chen, Y.S. Ong (Eds.), Advances in Natural Computation, 2005, pp. 751–760.

[35] D. Zou, L. Gao, S. Li, J. Wu, Solving 0–1 knapsack problem by a novel global harmony search algorithm, Appl. Soft Comput. 11 (2) (2011) 1556–1564.

[36] X. Kong, L. Gao, H. Ouyang, S. Li, A simplified binary harmony search algorithm for large scale 0–1 knapsack problems, Expert Syst. Appl. 42 (12) (2015) 5337–5355.

[37] L. Wang, R. Yang, Y. Xu, Q. Niu, P.M. Pardalos, M. Fei, An improved adaptive binary Harmony Search algorithm, Inform. Sci. 232 (2013) 58–87.

[38] D. Dua, C. Graff, UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences, 2017, http://archive.ics.uci.edu/ml.

[39] W. Siedlecki, J. Sklansky, A note on genetic algorithms for large-scale feature selection, Pattern Recognit. Lett. 10 (5) (1989) 335–347.

[40] I.S. Oh, J.S. Lee, B.R. Moon, Hybrid genetic algorithms for feature selection, IEEE Trans. Pattern Anal. Mach. Intell. 26 (11) (2004) 1424–1437.

[41] F. Pourpanah, C.J. Tan, C.P. Lim, J.M. Saleh, A Q-learning-based multi-agent system for data classification, Appl. Soft Comput. 52 (2017) 519–531.

[42] F. Pourpanah, R. Wang, C.P. Lim, X. Wang, M. Seera, C.J. Tan, An improved fuzzy ARTMAP and Q-learning agent model for pattern classification, Neurocomputing 359 (2019) 139–152.

[43] F. Pourpanah, C.P. Lim, Q. Hao, A reinforced fuzzy ARTMAP model for data classification, Int. J. Mach. Learn. Cybern. 10 (7) (2019) 1643–1655.

[44] F. Pourpanah, B. Zhang, R. Ma, Q. Hao, Non-intrusive human motion recognition using distributed sparse sensors and the genetic algorithm based neural network, in: 2018 IEEE SENSORS, 2018, pp. 1–4.

[45] H. Dong, T. Li, R. Ding, J. Sun, A novel hybrid genetic algorithm with granular information for feature selection and optimization, Appl. Soft Comput. 65 (2018) 33–46.

[46] L.Y. Chuang, C.H. Yang, J.C. Li, Chaotic maps based on binary particle swarm optimization for feature selection, Appl. Soft Comput. 11 (1) (2011) 239–248.

[47] H. Haklı, H. Uğuz, A novel particle swarm optimization algorithm with Levy flight, Appl. Soft Comput. 23 (2014) 333–345.

[48] P. Moradi, M. Gholampour, A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy, Appl. Soft Comput. 43 (2016) 117–130.

[49] K. Chen, F.Y. Zhou, X.F. Yuan, Hybrid particle swarm optimization with spiral-shaped mechanism for feature selection, Expert Syst. Appl. 128 (2019) 140–156.

[50] H. Wang, B. Niu, A novel bacterial algorithm with randomness control for feature selection in classification, Neurocomputing 228 (2017) 176–186.

[51] M.M. Mafarja, S. Mirjalili, Hybrid Whale Optimization Algorithm with simulated annealing for feature selection, Neurocomputing 260 (2017) 302–312.

[52] A. Zakeri, A. Hokmabadi, Efficient feature selection method using real-valued grasshopper optimization algorithm, Expert Syst. Appl. 119 (2019) 61–72.

[53] M. Wang, C. Wu, L. Wang, D. Xiang, X. Huang, A feature selection approach for hyperspectral image based on modified ant lion optimizer, Knowl.-Based Syst. 168 (2019) 39–48.

[54] M. Tubishat, N. Idris, L. Shuib, M.A. Abushariah, S. Mirjalili, Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection, Expert Syst. Appl. 145 (2020) 113122.

[55] E. Zorarpacı, S.A. Özel, A hybrid approach of differential evolution and artificial bee colony for feature selection, Expert Syst. Appl. 62 (2016) 91–103.

[56] M.Z. Baig, N. Aslam, H.P. Shum, L. Zhang, Differential evolution algorithm as a tool for optimal feature subset selection in motor imagery EEG, Expert Syst. Appl. 90 (2017) 184–195.

[57] Y. Zhang, D. wei Gong, X. zhi Gao, T. Tian, X. yan Sun, Binary differential evolution with self-learning for multi-objective feature selection, Inform. Sci. 507 (2020) 67–85.

[58] R. Diao, Q. Shen, Feature selection with harmony search, IEEE Trans. Syst. Man Cybern. B 42 (6) (2012) 1509–1523.

[59] Y.F. Huang, S.M. Lin, H.Y. Wu, Y.S. Li, Music genre classification based on local feature selection using a self-adaptive harmony search algorithm, Data Knowl. Eng. 92 (2014) 60–76.

[60] Z. Zainuddin, K.H. Lai, P. Ong, An enhanced harmony search based algorithm for feature selection: Applications in epileptic seizure detection and prediction, Comput. Electr. Eng. 53 (2016) 143–162.

[61] V. Elyasigomari, D. Lee, H. Screen, M. Shaheed, Development of a two-stage gene selection method that incorporates a novel hybrid approach using the cuckoo optimization algorithm and harmony search for cancer classification, J. Biomed. Inform. 67 (2017) 11–20.

[62] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary ant lion approaches for feature selection, Neurocomputing 213 (2016) 54–65.