International Journal of Machine Learning and Cybernetics (2020) 11:573–602
https://doi.org/10.1007/s13042-019-00996-5

ORIGINAL ARTICLE

Feature selection based on rough set approach, wrapper approach, and binary whale optimization algorithm

Mohamed A. Tawhid[1] · **Abdelmonem M. Ibrahim**[1,2]

Received: 9 October 2017 / Accepted: 2 August 2019 / Published online: 21 August 2019
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract
The principle of any approach for solving feature selection problem is to find a subset of the original features. Since finding a minimal subset of the features is an NP-hard problem, it is necessary to develop and propose practical and efficient heuristic algorithms. The whale optimization algorithm is a recently developed nature-inspired meta-heuristic optimization algorithm that imitates the hunting behavior of humpback whales to solve continuous optimization problems. In this paper, we propose a novel binary whale optimization algorithm (BWOA) to solve feature selection problem. BWOA is especially desirable and appealing for feature selection problem whenever there is no heuristic information that can lead the search to the optimal minimal subset. Nonetheless, whales can find the best features as they hunt the prey. Rough set theory (RST) is one of the effective algorithms for feature selection. We use RST with BWOA as the first experiment, and in the second experiment, we use a wrapper approach with BWOA on three different classifiers for feature selection. Also, we verify the performance and the effectiveness of the proposed algorithm by performing our experiments using 32 datasets from the UCI machine learning repository and comparing the proposed algorithm with some powerful existing algorithms in the literature. Furthermore, we employ two nonparametric statistical tests, Wilcoxon Signed-Rank test, and Friedman test, at 5% significance level. Our results show that the proposed algorithm can provide an efficient tool to find a minimal subset of the features.

Keywords Feature selection · Classification · Whale optimization algorithm · Rough set theory · Wrapper approach · Logistic regression

1 Introduction

One of the most intrinsic problems in the field of machine learning is feature selection (FS). FS has been successfully applied in many applications, such as text categorization [1, 2], pattern recognition [3, 4], image processing [3, 5], bioinformatics [6, 7], and so on. The primary objective of FS is to find out a minimal feature subset of a problem domain while preserving an appropriately high accuracy in depicting the original features [8].

In real-life situations, eliminating noisy, irrelevant, and deceptive features will not influence learning performance. Indeed, elimination of noisy and irrelevant features may help learn a better model, as irrelevant and noisy features, may mislead the learning system and make memory and computation ineffectiveness [9].

The purpose of the FS [10] in data mining and machine learning is to: decrease the dimensionality of feature space, enhance the prognostic accuracy of a classification algorithm, and improve the visualization and the lucidity of the generated concepts.

Recently, many FS algorithms have been developed and suggested. Designing a FS algorithm relies on evaluating measures and search strategies. Several search strategies have been developed such as complete [11], heuristic [12], random [13], strategies. Evaluating measures methods can

✉ Mohamed A. Tawhid
mtawhid@tru.ca

Abdelmonem M. Ibrahim
abdelmonem@azhar.edu.eg

[1] Department of Mathematics and Statistics, Faculty of Science, Thompson Rivers University, Kamloops, BC V2C 0C8, Canada

[2] Department of Mathematics, Faculty of Science, Al-Azhar University, Assiut Branch, Assiut, Egypt

be unceremoniously classified into classifier independent [8, 14] and classifiers-specific [15–17].

The evaluation measures, which determines the goodness of the selected attributes, is an essential issue in dimension reduction. Relying on the evaluation measures, dimension reduction methods are commonly divided into two broad classes: wrapper approaches and filter approaches [18]. Wrapper approaches comprise a learning/classification method to calculate the chosen attributes. Thus, wrappers usually get better classification performance than filter methods, but they experience from the high computation cost and the loss of generality, i.e., specific to a particular classification algorithm. Filter methods are independent of any learning algorithm. Thus, filter approaches usually need proper evaluation measures.

The size of search space in dimension reduction problems grows exponentially along with the number of attributes. A meticulous search is inappropriate in most of the cases. Various heuristic search approaches have been applied to dimension reduction problems [19], but most of them still have the restrictions of the long computation time and being stuck in the local optima [18]. Thus, it is desirable to solve dimension reduction problems by the least expensive global search algorithm. Evolutionary computation (EC) algorithms are widely known as global search and stochastic algorithms, which have been employed for dimension reduction such as genetic programming (GP) [20], genetic algorithms (GAs) [21, 22], scatter search algorithms (SSAs) [23, 24], simulated annealing (SA) [25, 26], ant colony optimization (ACO) [27], GRASP [28], tabu search [28, 29], differential evolution (DE) [30, 31], bat algorithms (BA) [32], biogeography-based optimization (BBO) [33], particle swarm optimization (PSO) [34, 35], binary grey wolf optimization (bGWO) [36], and binary dragonfly algorithm (BDA) [37].

Most current EC based on dimension reduction algorithms are wrapper approaches. The use of such algorithms is limited in real-world applications because of the long computation time. The development of EC by filter dimension reduction approaches is still an open issue. Rough set theory [38] is able to deal with uncertainty, imprecision, and vagueness, which has been successfully used for dimension reduction [39]. One of the significant applications of the rough set theory is the attribute reduction, that is, the elimination of attributes considered to be redundant, while avoiding information loss [38, 40].

Whale optimization algorithm (WOA) [41] is one of the recent metaheuristic optimization algorithms and based on the hunting behavior of humpback whales. WOA employs the search process using artificial whales as search agents. The humpback whales have a unique hunting technique named bubble-net feeding [41]. It has been shown that this algorithm gives competitive results when it is compared to other metaheuristic algorithms such as PSO and GA. Indeed, WOA has been developed to solve optimization problems with continuous real search spaces and also for feature selection problem [42–44].

In this paper, we propose a new feature selection algorithm based on binary whale optimization algorithm (BWOA). BWOA starts by randomly selecting the features (whales), which change their positions according to a new proposed transfer function. BWOA has a strong search ability in the problem space and can efficiently obtain the minimal feature subset.

This paper proposes a rough method to remove the redundant and irrelevant features combining with BWOA as a first experiment. In rough set experiment, we compare our proposed algorithm BWOA with five existing rough set-based attribute reduction algorithms, namely, ant colony optimization for rough set attribute reduction (AntRSAR) [45], simulated annealing for rough set attribute reduction (SimRSAR) [46], a rough set approach to FS based on ACO (RSFSACO) [47], tabu search (TSAR) [29] and scatter search (SSAR) [24], and two resent powerful algorithms, bGWO [36], and BDA [37]. Also, we use a wrappers approach for feature selection based on three different classifiers, such as Logistic Regression (LR) [48, 49, 49], Decision Tree Classifier (C4.5) [50, 51] and Naïve Bayes (NB) [52, 53]. We utilize the classifiers in the experiments based on trial and error basis. In wrapper approach, we use two kinds of validation tests. In the first test, we use 50–50 training-validation test where the instances are divided into two sets, namely, training and validation. A feature subset is assessed by the validation performance of a classifier training on the training set using that feature subset. In the second test, we use 10-fold cross-validation test. Moreover, we apply our proposed feature selection algorithm on datasets and compare with recent existing algorithms in literature such as binary bat algorithm (BBA) [54], bGWO [36], WOA with crossover and mutation operators (WOA-CM) [42], WOA combined with SA (WOASA) [43], and binary particle swarm optimization combining with the gravitational search algorithm (BPSOGSA) [55]. Furthermore, we verify the performance of the proposed algorithm by performing our experiments using 32 datasets from the UCI machine learning repository.

The rest of the paper is organized as follows. Section 2 gives an outline of the standard whale optimization algorithm. Section 3 presents the proposed algorithm and Sect. 4 describes the experimental design for feature selection problem. Section 5 presents the experimental results and discussions. Section 6 concludes the study and provides an insight into the future trends.

2 Whale optimization algorithm

In 2016, Mirjalili and Lewis [41] introduced Whale Optimization Algorithm (WOA) as a nature-inspired meta-heuristic optimization algorithm, which imitates the hunting behavior of humpback whales to solve complex continuous nonlinear functions. The idea of this algorithm is inspired by the behavior of humpback whales which favor hunting school of krill or small fishes near to the surface. They plunge around 12 m down and then begin to initiate bubble in a spiral shape around the prey and swim up toward the surface, which is known bubble-net feeding method. The spiral bubble-net feeding maneuver is an essential idea adopted by WOA. The WOA algorithm consists of three steps of encircling prey, spiral bubble-net feeding maneuver, and search for prey, see Algorithm 1.

Algorithm 1: Pseudo code of WOA

```
Input  : N Number of humpback whales,
         n Number of variables,
         t Number of iterations.
Output : X* Optimal humpback whale position,
         f(X*) Best fitness value.
Initialize a population of N whales positions randomly;
Calculate the fitness value of each whale/agent;
while Stopping criteria are not met do
    for i ← 1 to N (all N whales in the population) do
        I Update a, A and C according to (5), (3) and (4);
        II Generate random numbers l ∈ [−1, 1], p ∈ [0, 1];
        for d ← 1 to n (all n variables in the current agent) do
            if p < 0.5 then
                if |A| < 1 then
                    % Shrinking encircling preys
                    Update the dth position of the agents according to (1);
                else
                    % Search for prey
                    Update the dth position of the agents based on a random agent (X_r)
                      according to (9);
                end
            else
                % Spiral bubble-net attacking
                Update the dth spiral position of the agents according to (6);
            end
        end
    end
    Evaluate the positions of individual agents;
    Set the best fitness f(X*) value to the better solution X*.
end
```

1. *Encircling prey* The WOA algorithm assumes that the current best candidate solution is the target prey in the search space that it is not a priori known, or is near to the optimum. This replicates to recognize the location of prey and encircle them by humpback whales. The new agents will update their positions across the best search agent after the best search agent is defined. It is expressed mathematically as follows [41]

$$X_i^d(t+1) = X^{*d}(t) - A_i D_i^d \tag{1}$$

$$D_i^d = \left| C_i X_i^{*d}(t) - X_i^d(t) \right| \tag{2}$$

where X_i is the ith candidate solution for the current iteration t, X^* is the best obtained solution, d denotes dimension of the search space $1 \le d \le n$ and $||$ is the absolute value. The coefficient vectors A_i and C_i for each ith solution are computed as follows:

$$A_i = 2\,a\,r_1 - a \tag{3}$$

$$C_i = 2\,r_2 \tag{4}$$

where r_1 and r_2 are random values within the range [0,1], and a is linearly reduced over the course of iterations to range from 2 to 0 according to

$$a = 2\left(1 - \frac{t}{MaxIter}\right) \tag{5}$$

where t is the number of iterations and $MaxIter$ is the total number of iterations allowed for the optimization. Eq. (1) enables any search agent according to the random value r for each dth dimension to update its position in the neighborhood of the current best solution and imitates encircling the prey.

2. *Spiral bubble-net attacking method* The humpback whales attack the prey with the bubble-net strategy which can be regarded in WOA as "exploitation phase". This method is mathematically expressed by two approaches as follows [41]:

    1. *Shrinking encircling mechanism* This is accomplished by reducing the value of a in Eq. (3). The alternate range of A is reduced by a, and A is a random value in the interval $[-a, a]$.
    2. *Spiral updating position* A spiral equation is constructed between the position of whale and prey to imitate the helix-shaped movement of humpback whales.

The simultaneous behavior of swimming humpback whales around the prey within a shrinking circle and along a spiral-shaped path simultaneously can be modeled by selecting between either the shrinking encircling mechanism or the spiral model with a probability of 50% to update the position of whales during the optimization, which can be expressed mathematically as follows:

$$X_i^d(t+1) = \begin{cases} X^{*d}(t) - A_i D_i^d, & \text{if } p < 0.5; \\ D_i^{\prime d} \exp(b\,l)\cos(2\pi\,l) + X^{*d}, & \text{if } p \ge 0.5. \end{cases} \tag{6}$$

$$D_i^{\prime d} = |X^{*d}(t) - X_i(t)|, \tag{7}$$

where $D_i^{'d}$ is the distance of the ith solution (whale) to the best obtained solution (prey) so far, b is a constant describing the shape of the logarithmic spiral, l is a uniformly distributed random number within the range $[-1, 1]$ and p is a random number within the range [0, 1].

3. *Search for prey* Another behavior of the humpback whales is that they randomly search for prey based on the position of each other and the bubble-net method. This is based on the value A_i in Eq. (3), which can be applied to search for prey (exploration phase). The exploration phase relies on updating the position of a search agent X_i as in Eq. (1) by randomly selecting a search agent X_r (a random whale) instead of the best-obtained search agent X^* to move far away from a reference whale, as in Eq. (9). When $|A| > 1$ with the random values, the exploration gives the WOA algorithm the ability to avoid trapping in the local minimum and achieve a global search [41], see Algorithm 1.

$$D_i^{''d} = \left| C_i X_r^d(t) - X_i^d(t) \right| \tag{8}$$

$$X_i^d(t+1) = X_r^d(t) - A_i D_i^{''d} \tag{9}$$

3 Binary whale optimization algorithm

The WOA is simple and effective to explore global solutions [41, 56]. We introduce a novel algorithm, the binary whale optimization algorithm (BWOA), in order to improve the solution accuracy and reliability of finding the best solutions for the FS problem. Note, the whales/agents positions of the binary version are in {0,1}. Our goal is to keep the proposed algorithm as simple as the original algorithm with the adjustment in proportion to the binary version, see Algorithm 2.

The idea of WOA is based on the humpback whales hunting method that is called bubble-net hunting strategy. Humpback whales swim around the prey within a shrinking circle and along a spiral-shaped path simultaneously to create special bubbles along a circle or "9"-shaped path. This behavior is imitated in WOA. There is a probability p of 50%. In our proposed algorithm BWOA we change this probability into 40% in order to choose between the shrinking encircling mechanism and the spiral model to give more intensification in the search space for the spiral model, see BWOA Flowchart 2. BWOA formulations are described as follows.

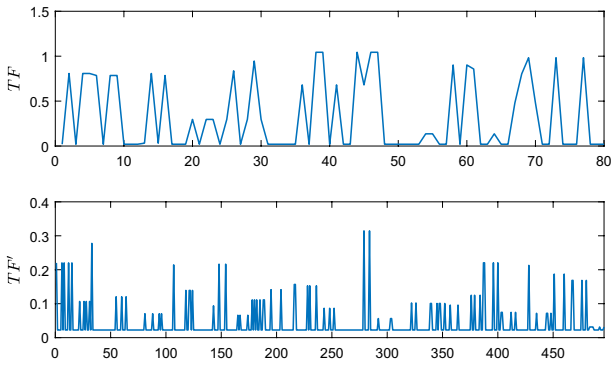

Fig. 1 The TF and TF' values using Zoo dataset

Algorithm 2: Pseudo code of BWOA

Input : N Number of humpback whales,
n Number of variables,
t Number of iterations,
$MaxItr$ Maximum number of iterations.
Output : X^* Optimal humpback whale position (minimal subset features)
$f(X^*)$ Best fitness value.
Initialize a binary population of N whales positions randomly;
while *Stopping criteria are not met* **do**
  for $i \leftarrow 1$ ***to*** N *(whales in the population)* **do**
    Calculate the fitness value of the ith whale/agent;
    if $f(X_i) <= f(X^*)$ **then**
      Set current fitness value to X^* value;
      Update the set of the best leaders ($Lbest$) so far;
    end
    if $t > \frac{MaxItr}{3}$ **then**
      Select X^* from the set of best leaders $Lbest$ randomly;
    end
    1. Update a, A and C according to Eqs. (5), (3) and (4);
    2. Generate a random numbers $l \in [-1, 1]$, $p \in [0, 1]$;
    for $d \leftarrow 1$ ***to*** n *(variables/features in the current agent)* **do**
      if $p < 0.4$ **then**
        if $|A| < 1$ **then**
          % **Shrinking encircling preys**
          Update the dth position of the agents according to best solution (X^*) as in Eq. (11);
        else
          % **Search for prey**
          Update the dth position of the agents based on a random agent (X_r) according to transfer function Eq. (15);
        end
      else
        % **Spiral bubble-net attacking**
        Update the dth spiral position of the agents according to Eq. (14);
      end
    end
  end
end
Give X^* as the minimal reduct.

3.1 Shrinking encircling preys

In BWOA, as in WOA, the currently best candidate solution is assumed to be the target prey and the other search agents try to update their positions towards it. Mathematically, this behavior is described in Eqs. (2), (3) and (5) to create a new position as in Eq. (1). We modify the V-shaped transfer function (TF) [57], by converting the step values in Eq. (2) into a new values in the range of [0,1] as follows:

$$TF_i^d = \left| \alpha \cdot \arctan\left(\frac{\pi}{3} A_i D_i^d \right) + \beta_1 \right| \tag{10}$$

Table 1 Run and evolutionary parameter values for β_1 of Eq. (10)

																	β_1	No. of target features	No. of 1s after changing	Changed %
X^*	0	0	0	1	1	1	0	1	0	0	1	0	1	0	0	1	0			
$X_i(t+1)$	**0**	**0**	0	1	1	1	**0**	1	**0**	0	1	0	1	**0**	0	1		5	0	0%
X^*	0	0	1	0	1	1	1	1	1	0	0	1	1	1	0	0	0.02			
$X_i(t+1)$	**0**	**0**	1	**0**	1	1	1	1	1	**0**	**0**	1	1	1	**0**	**1**		7	0 or 1	0-14%
X^*	1	0	0	0	0	1	1	0	1	1	0	1	1	1	1	0	0.3			
$X_i(t+1)$	1	**1**	**1**	**0**	**1**	1	1	**0**	1	1	**0**	1	1	1	1	**0**		7	3	43%
X^*	0	0	1	1	0	1	0	1	1	1	0	0	1	0	0	0	0.05			
$X_i(t+1)$	**0**	0	1	1	**1**	1	**0**	1	1	1	0	**1**	1	**1**	**1**	1		6	4	67%

where α is a constant value and equals to $3/\pi$, this value is chosen in order to make sure TF_i^d in the range [0, 1], see Fig. 1. A is a coefficient vector in Eq. (3), D_i^d denotes the distance between the ith whale and the prey and is given in Eq. (2) and β_1 is a new parameter has been added to Eq. (10) to avoid TF_i^d equating to zero in case of X^{*d} and X_i^d are zeros.

The new dth whale position/feature in our binary proposed method depends on the historically best feature X^* as in WOA (Eq. (1)). The updating whale features based on X^* can mathematically describe as follows.

$$X_i^d(t+1) = \begin{cases} 1 - X^{*d}(t), & rd_1 < TF_i^d; \\ X^{*d}(t), & \text{Otherwise}, \end{cases} \tag{11}$$

where rd_1 is an uniformly distributed random number in the interval [0,1]. The value of β_1 is selected to be 0.02 to avoid continuing updating features in such a case ($TF_i^d = 0$) in the long part of search unchanged. β_1 is very small to have slightly change for not all the features updated based on Eq. (11), see Table 1. It is worth mentioning that the selection of the parameters of the proposed algorithms is based on the computer experiment in selecting, adding, removing or changing new features appropriately during the search. For example, according to Eq. (11), if $\beta_1 = 0$, then the most of the attributes will change to the same of X^*, which leads to stick in the region of this local solution X^* during the search. On the other side, if β_1 is bigger (say 0.2), then the most of the attributes will change to the opposite of the considered features of X^*. This undoubtedly leads to a departure from the region of the current best solution that may be promising and thus missing the goal of the reduction in this search. For more details of setting the value of parameter β_1, see Table 1, where the underlines in the table indicate the best selection of β_1 (the results in this table are taken after the experiment on 'Zoo' dataset).

3.2 Spiral bubble-net feeding maneuver.

A spiral equation is used between the position of whale X and prey (best solution X^*) to imitate the helix-shaped movement of humpback whales. In BWOA, the spiral step in Eq. (6) is used to describe another new transfer function TF' as follows:

$$TF_i'^d = \frac{\left|\arctan\left(S_i^d\right) + \beta_2\right|}{4} \tag{12}$$

$$S_i^d = D_i'^d \exp(b\, l)\cos(2\pi\, l) \tag{13}$$

where S is walk steps derived from a spiral Eq. (6), and D_i' is the distance between the ith solution and the best solution (Eq. (7)). β_2 is a constant value which sets as 0.09. This value is chosen in order to make sure $TF_i'^d$ is not equal to zero (see Fig. 1), while the right side is divided by 4 to keep the TF' values in the range [0, 0.5]. One can see from Eq. (7), if X^{*d} and X_i^d are 1s or 0s (considered or non-considered features, respectively), then the value of D_i' equals to zero and therefore $TF_i'^d$ equals to $\beta_2/4$, i.e., in case $\beta_2/4$ is not exist, then the algorithm can easily trap at the local minimum (features of poor quality) because it is not able to change or replace these features. Consequently, the mathematical model of the updating of ith whale position/feature of spiral bubble-net feeding can be calculated as follows.

$$X_i^d(t+1) = \begin{cases} 1 - X^{*d}(t), & rd_2 > \lambda,\ TF_i'^d = \frac{\beta_2}{4}; \\ X^{*d}(t), & rd_2 > TF_i'^d; \\ X_i^d(t), & \text{Otherwise}, \end{cases} \tag{14}$$

where rd_2 is a uniformly distributed number in the range of [0, 1]. The value of λ is selected to be 0.92 to make sure the updating features can slightly switch or change in case of $D_i' = 0$ (i.e., $X^{*d} = X_i^d$) and do not stay during the search

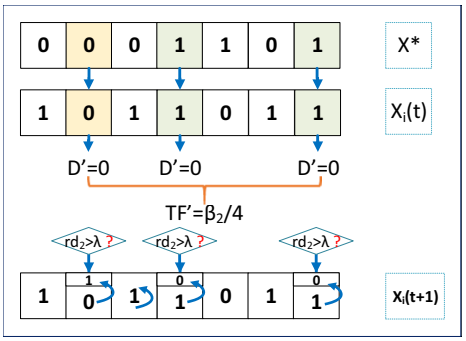

Fig. 2 Updating spiral features using Eq. (14) when $D' = 0$

without changing, see Fig. 2. This parameter λ makes a difference in the results, especially when those features are promising features. Also, it can update the current solution if there is more than one of the same solution in the current population,

3.3 Search for prey

In BWOA, we have a diversification when parameter A is greater than 1; the search agent is updated according to a randomly chosen search feature instead of the best search feature by using TF (in Eq. (10)). $D_i^{''d}$ (in Eq. (8)) is used instead of D_i^d. The ith whale position/feature based on a randomly chosen search features is mathematically modeled as the following:

$$X_i^d(t+1) = \begin{cases} 1 - X_r^d(t), & rd_3 < TF_i^d; \\ X_r^d(t), & \text{Otherwise.} \end{cases} \quad (15)$$

where rd_3 is a random number uniformly distributed in the range of [0, 1] and X_r is a random whale from the current population. When A is greater than 0.1, we update the search feature according to the best search feature from Eq. (11). When A is less than 0.1, we diversify the current solution based on selecting the features randomly from the other current solutions so far, the BWOA Flowchart is shown in Fig. 3.

To enhance the performance of the search and improve the quality of the new solution, we add a new procedure to BWOA with the aim of moving to a promising research area around the obtained best solutions so that we can find the optimum solution/minimum reduct. Since solving feature selection problem may have more than one optimal solution. The implementation of this procedure uses a memory structure that describes the visited best solutions (short-term: the set of best solutions recently considered). The set of the best solution/leaders is denoted by *Lbest*, the length of this set is selected as m of best solution so far. Also, during the search, the method may stick in suboptimal regions (features of poor

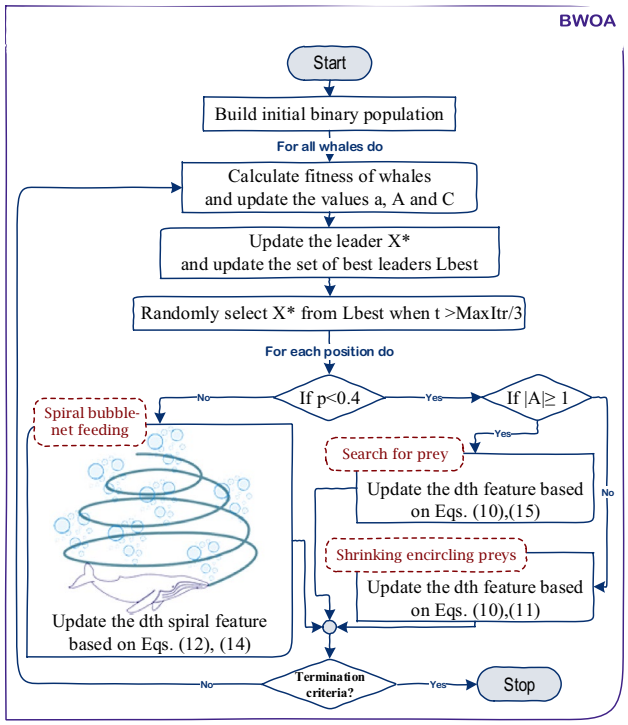

Fig. 3 Flowchart of BWOA

quality) or on plateaus where many solutions are equally fit. Thus, we apply this procedure after $\frac{MaxItr}{3}$ iterations, i.e., after giving the proposed algorithm a good time to get the best solutions, we randomly select $X^*(t) \in Lbest$ in order to improve the current solutions (see Algorithm 2). The length of *Lbest* m is set equal to 3 in this study. The length 3 may be enough to give a chance to another best solution during the search and achieve solution superior to those found in a regular search. This would allow updating the solutions based on one of the old best solution which may get solutions better than the currently known solutions. At the end of the search, BWOA selects the best solution X^* in *Lbest* as the minimum feature set.

4 Binary whale optimization algorithm for feature selection

Many researchers have attracted to the problem of finding a reduct of an information system. The most basic solution is to locate a subset, generate all possible subsets and retrieve those with minimum function value. This is an expensive solution to the problem and is only practical for a simple dataset. Most of the time only one reduct is required, typically, only one subset of features is used to reduce a dataset, so all the calculations involved in discovering the rest are pointless [46, 58, 59].

In the FS problem, a solution is a binary vector that describes the current subset. A subset is perturbed by randomly changing a small number of members in the subset, which drives a particular version of the WOA. The objective function can be the rough set which measures the solution whether it contains redundant features or not, also the classification accuracy or some other criteria that might consider the best trade-off between attribute extraction computational burden and efficiency [60].

In this section, we utilize the binary version of whale optimization algorithm (BWOA) in FS based on rough set and wrapper approaches, respectively, as we will explain later. Many researchers have proposed many methods based on the rough set approach to demonstrate the efficiency and effectiveness of their proposed methods, see, e.g., [58, 61, 62], but in this work, we propose our algorithm based on not only rough set approach but also wrapper approach to show the efficiency of our algorithm. It is known that there is a vast space for different feature reduction that would be 2^n where n is the length of features. Therefore, we adopt the BWOA to search the feature space for the best features combination and reach the optimal solution x^* systematically and efficiently.

4.1 Rough set theory and wrappers for feature selection

4.1.1 Rough set theory

Rough set theory (RST) [38, 63] is one of the dominant approaches to FS, which can maintain the meaning of the features. The intrinsic nature of the rough set approach to FS is to find a subset of the original features [47, 64]. RST is a mathematical approach for dealing with imperfectness knowledge, i.e., to imprecision (or vagueness). Objects may be difficult or impossible to perceive due to the limited available information. A rough set is characterized by a pair of precise concepts, called lower and upper approximations, generated using indistinctness of the object. Here, the most important problems are the reduction of attributes and the generation of decision rules. In rough set theory, repugnance is not corrected or aggregated. Instead, the lower and upper approximations of all decision concepts are calculated, and rules are produced. The rules are classified into specific and approximate (possible) rules based on the lower and upper approximations, respectively.

Let $S = (U, \mathbb{A})$ be an information system, where $\{U\}$ is a non-empty set of finite objects, called the universe of discourse, and $\mathbb{A}$ is a non-empty set of attributes. With every attribute $a \in \mathbb{A}$, a set of its values V_a is associated [38]. For a subset of attributes P, the indiscernibility relation is defined by $IND(P)$ [38, 64] as follows.

$$IND(P) = \{(\xi, \eta) \in U \times U | \forall a \in P, a(\xi) = a(\eta)\}. \tag{16}$$

The relation $IND(P), P \subseteq \mathbb{A}$, constitutes a partition of U, which is denoted $U/IND(P)$. If $(\xi, \eta) \in IND(P)$, then ξ and η are indiscernible by attributes from P. The equivalence classes of the P-indiscernibility relation are denoted $[\xi]_P$. For a subset $\Xi \subseteq U$, the P-lower approximation of Ξ can be defined as

$$\underline{P}\Xi = \{\xi | [\xi]_P \subseteq \Xi\}. \tag{17}$$

Let $IND(P)$ and $IND(Q)$ be indiscernibility relations on U, which are defined by the subset of attributes $P \subseteq \mathbb{A}$ and $Q \subseteq \mathbb{A}$, respectively. An often applied measure is the dependency degree of Q on P, which is defined as follows [38]:

$$\gamma_P(Q) = \frac{|POS_P(Q)|}{|U|}, \tag{18}$$

where $|F|$ denotes the cardinality of set F and $POS_P(Q) = \bigcup_{\Xi \in U/IND(Q)} \underline{P}\Xi$, called a positive region of the partition $U/IND(Q)$ with respect to P, is the set of all elements of U that can be uniquely classified to blocks of the partition $U/IND(Q)$ by means of P. We say that Q depends totally on P if $\gamma_P(Q) = 1$ and Q depends partially on P if $\gamma_P(Q) < 1$. The dependency degree expresses the ratio of all objects of U that can be properly classified to the blocks of the partition $U/IND(Q)$ using the knowledge in P.

Information system $S = (U, A)$ is called a decision system, if $A = \mathbb{C} \cup \mathbb{D}$, and $\mathbb{C} \cap \mathbb{D} = \phi$, where $\mathbb{C}$ is the set of condition features and $\mathbb{D}$ is the set of decision features. The degree of dependency between the condition and decision features, $\gamma_{\mathbb{C}}(\mathbb{D})$, is called the reduct of $\mathbb{C}$ [65].

The goal of feature reduction is to remove redundant features so that the reduced set provides the same quality of classification as the original. A reduct is defined as a subset R of the condition feature set $\mathbb{C}$ such that $\gamma_R(\mathbb{D}) = \gamma_{\mathbb{C}}(\mathbb{D})$ and $\forall B \subset R, \gamma_B(\mathbb{D}) \neq \gamma_{\mathbb{C}}(\mathbb{D})$. A subset $R_{sup} \subseteq \mathbb{C}$ is called a super reduct, if $\gamma_{R_{sup}}(\mathbb{D}) = \gamma_{\mathbb{C}}(\mathbb{D})$. A given decision table may have many reducts, the set of all reducts is defined as

$$\mathfrak{R} = \{R : R \subseteq \mathbb{C} | \gamma_R(\mathbb{D}) = \gamma_{\mathbb{C}}(\mathbb{D}), \gamma_B(\mathbb{D}) \neq \gamma_{\mathbb{C}}(\mathbb{D})\}. \tag{19}$$

In rough set feature reduction, a reduct with minimal cardinality is called the minimal reduct, which can be defined as follows

$$R_{\min} = \{R \in \mathfrak{R} | \forall R' \in \mathfrak{R}, |R| \leq |R'|, \}. \tag{20}$$

4.1.2 Wrappers for feature selection

Wrapper (WR) methods are commonly named because they wrap a classifier up in the FS algorithm [21, 22, 36, 66]. A set of features is usually selected; the effectiveness of this set is evaluated, and some perturbation is made to replace the original set, and the effectiveness of the new set is calculated. The problem with these methods is that feature space is vast

and considers all combinations would take a lot of time and computation. Therefore, some heuristic search approaches must be developed to find optimum sets of features. The main characteristic of the wrapper approach in FS is the use of the classifier as an indicator of FS procedure. Wrapper-based FS method can be categorized into three main components: search method, classification method, and feature evaluation criteria.

In this study, we employ three popular classifiers as a classification method step for wrappers approach, namely, Logistic Regression (LR) [48, 49, 49], Decision Tree Classifier (C4.5) [50, 51] and Naïve Bayes (NB) [52, 53], more details in "Appendix".

4.2 Formulation of fitness function

To evaluate individual whales positions (features subsets) for the both experiments RS and wrapper approach, we use the fitness function (minimization problem) as shown in Eq. (21) [36].

$$f = \mu \times E_R + \nu \times \frac{|L_R|}{|n|}, \tag{21}$$

where $E_R = 1 - \gamma_R$ for RS experiment, γ_R is the reduct value of the reduct subset R (see Sect. 4.1.1), while in wrapper method E_R indicates the error rate of the classifier for the subset R (misclassification), L_R is the length of reduction attributes/features, and n is the total number of original features. Besides, $\mu \in [0, 1]$ and $\nu = 1 - \mu$ are constants to find a trade-off between classification and the length of reduct features [36].

Note that the most basic solution is to locate a subset, the RS dependency degree/reduct of subset R (γ_R) is in the range [0,1] (R is the solution that given by BWOA), where the maximum RS dependency degree equals to 1. The algorithm stops when it gets the minimum reduct number "$\gamma_R = 1$" of subset features for dataset under study (the column of 'MinRed' in Table 4), which means that the current solution is the globally optimal solution or continues until the end of iterations is reached. In the second experiment, we use three classifiers, namely, LR, C4.5, and NB, as a classification step for the wrappers approach. The algorithm proceeds until the end of iterations is reached.

5 Computational experiments

We consider thirty-two datasets from the UCI machine learning repository [67]. We present two approaches in congruence with our experiments and comparisons results. We designate nineteen of the datasets for rough set experiment and eighteen datasets for the wrapper experiment (note that some of the datasets are used in both experiments). The details of the shared datasets are described in the following subsections.

Table 2 parameter settings of BWOA for experiments

Parameter	Value	
	RS	WR
n	Problem features No.	
Population size	$2n$	8
Maximum number of iterations	50	70
No. of runs	20	20
α	$3/\pi$	$3/\pi$
β_1	0.02	0.02
β_2	0.09	0.09
b	1	1
λ	0.92	0.92

In Table 2, we summarize all the initial parameters and assign their values of both experiments RS and WR, and the parameter sitting of BWOA (selection of the parameter settings values β_1, β_2, and λ are explained in the subsections of Sect. 3). In our implementations to RS and WR approaches, the parameters of the associated algorithms in literature are assigned from their original papers as follows: In BBA, the loudness of emitted sound A is 0.25, pulse emission rate r is 0.1, and frequency minimum and frequency maximum are 0 and 2, respectively. In BDA, the swarming factors $w = 0.9 - 0.4, s = 0.1, a = 0.1, c = 0.7, f = 1, e = 1,$ and the constant β in Lèvy flight is equal to 1.5. In PSOGSA, the initial gravitational constant G_0 is 1, the accelerating factors $c_1' = -2\frac{t3}{T^3} + 2$ and $c_1' = -2\frac{t3}{T^3} + 2$, where t indicates the current iteration, T is the maximum number of iterations, and the descending coefficient α is 20. In addition, we compare with two of the new improvements of WOA for feature selection, namely, WOA with crossover and mutation operators (WOA-CM) [42] and the hybrid WOA with SA (WOASA) [43]. BDA, bGWO, WOA-CM and WOASA are parameter-less algorithms, where in BDA algorithm, the inertia weight $w = 0.9 - 0.4$. Note that the versions bGWO2 and WOASA-2 are selected and used in this study as they have proved their efficiency in solving the FS problem [36]. The results are shown in following tables in which the bold font indicate the best results among all the methods.

5.1 Numerical experiments for rough set theory

We conduct a rough set theory for attribute reduction to show the efficacy of the proposed feature selection algorithm. Also, we perform BWOA and the shared algorithms with random initial binary populations for each dataset. To evaluate the performance of our BWOA using RST for feature selection problem, we compare BWOA with five existing results of rough set-based attribute reduction

Table 3 Related work for feature selection problem based on rough set approach

Authors	Algorithms based on rough set approach
Jue et al. [24]	In this paper, the authors suggested a rough set approach to feature selection based on scatter search metaheuristic and called the proposed algorithm by scatter search rough set attribute reduction (SSAR). They applied SSAR on 13 known datasets from UCI machine learning repository and compared it with typical attribute reduction methods including genetic algorithm, ant colony, simulated annealing, and Tabu search
Hedar et al. [29]	The authors proposed a tabu search (TS)-based method, named tabu search for attribute reduction (TSAR), to solve the problem of attribute reduction of an information system. TSAR employs a 0–1 variable representation of solutions in searching for reducts. A rough set dependency degree function is implemented to estimate the solution qualities. The search process in TSAR is a high-level TS with long-term memory. Thus, TSAR carried out exploration and exploitation search schemes besides the TS neighborhood search technique. Also, the authors applied TSAR on 13 known datasets from the UCI machine learning repository to show the performance of TSAR
Emarya et al. [36]	In this work, the authors developed a binary version of the grey wolf optimization (GWO) and used to select the optimal feature subset for classification purposes. Grey wolf optimizer (GWO) is one of the latest bio-inspired optimization algorithms, which emulates the hunting process of grey wolves in nature. The authors used a wrapper approach for feature selection based on KNN classifier. However, they have not used a rough set approached with GWO. Therefore we implemented binary GWO with rough set approach on 18 datasets from UCI and compared it with our proposed algorithm binary whale optimization algorithm in order to show the efficiency and performance of our algorithm
Mirjalili [37]	In this paper, the author developed a novel swarm intelligence optimization technique called dragonfly algorithm (DA). The DA algorithm is inspired by the static and dynamic swarming behaviors of dragonflies in nature. Two primary phases of optimization, diversification, and intensification, are formed by considering the social interaction of dragonflies in searching for foods, navigating, and avoiding enemies when swarming statistically or dynamically. The author also considered the proposal of binary and multi-objective versions of DA. The proposed algorithms are benchmarked by several mathematical test functions. However, this algorithm was not considered for feature selection problem, so we implemented this algorithm and combined with the rough set approach to handle feature selection problem, and we applied the binary DA on 18 datasets from UCI. We further compared binary DA with our proposed algorithm binary whale optimization algorithm in solving the FS problem
Ke et al. [45]	In this paper, the authors introduced an approach based on ant colony optimization (ACO) for attribute reduction and verified ACO algorithm by carrying out on thirteen datasets and three gene expression datasets. Also, they showed that this algorithm could give efficient solutions
Jensen and Shen [46]	In this work, the authors studied Computational Intelligence (CI) tools for the attribute reduction (AR) problem. They presented three CI algorithms, GenRSAR, AntRSAR, and SimRSAR, to solve the AR problem. GenRSAR is a genetic algorithm-based algorithm, and its fitness function considers both the size of the subset and its evaluated suitability. AntRSAR is an ant-colony-based algorithm in which the number of ants is set to the number of attributes, with each ant starting on a different attribute. Ants construct possible solutions until they reach a rough set reduct. SimRSAR employs a simulated-annealing-based attribute selection mechanism. SimRSAR tries to update solutions, which are attribute subsets, by regarding three attributes to be added to the current solution or to be removed from it. Optimizing the objective function attempts to maximize the rough set dependency while minimizing the subset cardinality. They applied their tools on 13 datasets from UCI
Yumin et al. [68]	In this study, the author proposed a rough set approach to feature selection based on ACO, which employs mutual information based feature significance as heuristic information. Their algorithm starts from the feature core, which changes the complete graph to a smaller one. Also, the authors verified the efficiency of the algorithm by applying it on nine standard UCI datasets. Their results show that the algorithm can provide an efficient solution to find a minimal subset of the features

algorithms, namely, SSAR [24], TSAR [29], AntRSAR [45], SimRSAR [46] and RSFSACO [47]. In addition, we implement two well-known binary metaheuristic methods in the literature, BDA [37] and bGWO [36] to compare with BWOA. In Table 3, we summarize the considered and comparative algorithms that we compare with our proposed algorithm based on rough set approach.

The same parameter settings as suggested in the literature [24] are followed where the population size is set to $2n$, where n is the features number and 50 is the maximum number of iterations, see Table 2. It is worth mentioning that the number of population is fixed at $2n$ in applying BDA and bGWO algorithms for more comparable results.

In Tables 4 and 5, we verify the performance of the proposed algorithm BWOA and other algorithms by finding the minimum reduct of attributes of 19 datasets using rough set theory. In Table 4, we give the comparison results of the numbers of attributes in the best reducts obtained by each algorithm for 19 datasets where superscripts represent the number of successful runs, where the original features number "Feat#" for each dataset is described in the second column of the table.

From Table 4, we can see that BWOA can obtain the best minimal reducts for all tested datasets except for one dataset (Credit) where the BWOA obtains 15 times of the minimum reduct (8) while SimRSAR obtains 18 times of

Table 4 Best results obtained from BWOA and other algorithms by using rough set approach

Dataset	Feat#	MinRed	BWOA	BDA	bGWO	SSAR	TSAR	AntRSAR	SimRSAR	RSFSACO
Monk1	6	3	$\mathbf{3}^{(20)}$	$\mathbf{3}^{(20)}$	$3^{(8)}4^{(11)}5^{(1)}$	N/A	N/A	N/A	N/A	$\mathbf{3}^{(20)}$
Monk3	6	4	$\mathbf{4}^{(20)}$	$\mathbf{4}^{(20)}$	$4^{(6)}5^{(13)}6^{(1)}$	N/A	N/A	N/A	N/A	$\mathbf{4}^{(20)}$
Breast-cancer	9	4	$\mathbf{4}^{(20)}$	$\mathbf{4}^{(20)}$	$4^{(12)}5^{(8)}$	N/A	N/A	N/A	N/A	$\mathbf{4}^{(20)}$
M-of-N	13	6	$\mathbf{6}^{(20)}$	$\mathbf{6}^{(20)}$	$6^{(2)}7^{(4)}8^{(1)}9^{(5)}10^{(3)}$	$\mathbf{6}^{(20)}$	$\mathbf{6}^{(20)}$	$\mathbf{6}^{(20)}$	$\mathbf{6}^{(20)}$	N/A
Exactly	13	6	$\mathbf{6}^{(20)}$	$\mathbf{6}^{(20)}$	$7^{(2)}8^{(4)}9^{(4)}10^{(3)}11^{(3)}$	$\mathbf{6}^{(20)}$	$\mathbf{6}^{(20)}$	$\mathbf{6}^{(20)}$	$\mathbf{6}^{(20)}$	N/A
Exactly2	13	10	$\mathbf{10}^{(20)}$	$10^{(18)}11^{(2)}$	$11^{(6)}12^{(9)}13^{(5)}$	$\mathbf{10}^{(20)}$	$\mathbf{10}^{(20)}$	$\mathbf{10}^{(20)}$	$\mathbf{10}^{(20)}$	N/A
Heart	13	6	$\mathbf{6}^{(20)}$	$6^{(9)}7^{(11)}$	$6^{(1)}7^{(10)}8^{(5)}9^{(3)}10^{(1)}$	$\mathbf{6}^{(20)}$	$\mathbf{6}^{(20)}$	$6^{(18)}7^{(2)}$	$6^{(29)}7^{(1)}$	N/A
Vote	16	8	$\mathbf{8}^{(20)}$	$8^{(19)}9^{(1)}$	$10^{(3)}11^{(9)}12^{(7)}13^{(1)}$	$\mathbf{8}^{(20)}$	$\mathbf{8}^{(20)}$	$\mathbf{8}^{(20)}$	$8^{(15)}9^{(15)}$	9
Zoo	16	5	$\mathbf{5}^{(20)}$	$5^{(15)}6^{(5)}$	$5^{(1)}6^{(9)}7^{(4)}8^{(3)}9^{(3)}10^{(1)}$	N/A	N/A	N/A	N/A	$\mathbf{5}^{(20)}$
Credit	20	8	$8^{(15)}9^{(4)}10^{(1)}$	$8^{(3)}9^{(9)}10^{(5)}10^{(3)}$	$9^{(1)}10^{(4)}11^{(5)}12^{(8)}13^{(2)}$	$8^{(9)}9^{(8)}10^{(3)}$	$8^{(13)}9^{(5)}10^{(2)}$	$8^{(12)}9^{(4)}10^{(4)}$	$\mathbf{8}^{(18)}\mathbf{9}^{(1)}\mathbf{11}^{(1)}$	N/A
Mushroom	22	4	$\mathbf{4}^{(20)}$	$4^{(11)}5^{(9)}$	$5^{(3)}6^{(8)}7^{(7)}8^{(1)}9^{(1)}$	$4^{(12)}5^{(8)}$	$4^{(17)}5^{(3)}$	$\mathbf{4}^{(20)}$	$\mathbf{4}^{(20)}$	$4^{(13)}5^{(7)}$
LED	24	5	$\mathbf{5}^{(20)}$	$5^{(18)}6^{(2)}$	$8^{(2)}9^{(2)}10^{(4)}11^{(4)}12^{(3)}$	$\mathbf{5}^{(20)}$	$\mathbf{5}^{(20)}$	$5^{(12)}6^{(4)}7^{(3)}$	$\mathbf{5}^{(20)}$	N/A
Letters	25	8	$\mathbf{8}^{(20)}$	$8^{(12)}9^{(7)}10^{(1)}$	$8^{(2)}9^{(5)}10^{(8)}11^{(3)}13^{(1)}$	$8^{(5)}9^{(15)}$	$8^{(17)}9^{(3)}$	$\mathbf{8}^{(20)}$	$\mathbf{8}^{(20)}$	N/A
Derm	34	6	$\mathbf{6}^{(20)}$	$6^{(2)}7^{(8)}8^{(10)}$	$6^{(1)}8^{(2)}9^{(6)}10^{(7)}11^{(3)}$	$\mathbf{6}^{(20)}$	$6^{(14)}7^{(6)}$	$6^{(17)}7^{(3)}$	$6^{(12)}7^{(8)}$	N/A
Derm2	34	8	$\mathbf{8}^{(14)}\mathbf{9}^{(6)}$	$8^{(3)}9^{(7)}10^{(8)}11^{(2)}$	$9^{(1)}10^{(5)}11^{(5)}12^{(7)}$	$8^{(2)}9^{(18)}$	$8^{(2)}9^{(14)}10^{(4)}$	$8^{(3)}9^{(17)}$	$8^{(3)}9^{(7)}$	N/A
Chess-king	36	–	$\mathbf{29}^{(20)}$	$28^{(2)}29^{(5)}30^{(8)}31^{(2)}32^{(3)}$	$30^{(2)}31^{(5)}32^{(5)}33^{(4)}33^{(4)}$	N/A	N/A	N/A	N/A	$\mathbf{29}^{(20)}$
WQ	38	12	$\mathbf{12}^{(4)}\mathbf{13}^{(16)}$	$12^{(1)}13^{(2)}14^{(16)}15^{(1)}$	$14^{(2)}15^{(8)}16^{(6)}17^{(3)}18^{(1)}$	$13^{(12)}14^{(8)}$	$12^{(1)}13^{(13)}14^{(6)}$	$12^{(2)}13^{(7)}14^{(11)}$	$13^{(16)}14^{(4)}$	N/A
Lung	56	4	$\mathbf{4}^{(20)}$	$5^{(2)}6^{(6)}7^{(10)}8^{(2)}$	$8^{(1)}11^{(3)}12^{(4)}13^{(5)}$	$\mathbf{4}^{(20)}$	$4^{(6)}5^{(13)}6^{(1)}$	$\mathbf{4}^{(20)}$	$4^{(7)}5^{(12)}6^{(1)}$	N/A
Audiology	69	7	$\mathbf{7}^{(9)}\mathbf{8}^{(9)}\mathbf{9}^{(2)}$	$11^{(1)}12^{(1)}13^{(2)}14^{(16)}15^{(1)}$	$19^{(1)}20^{(1)}21^{(2)}21^{(5)}23^{(1)}$	N/A	N/A	N/A	N/A	$13^{(4)}14^{(16)}$

Table 5 Average results obtained from BWOA and other algorithms by using rough set approach

Dataset	BWOA	BDA	bGWO	SSAR	TSAR	AntRSAR	SimRSAR	RSFSACO
Monk1	**3**	**3**	3.65	N/A	N/A	N/A	N/A	**3**
Monk3	**4**	**4**	4.75	N/A	N/A	N/A	N/A	**4**
Breast-cancer	**4**	**4**	4.4	N/A	N/A	N/A	N/A	**4**
M-of-N	**6**	**6**	9	**6**	**6**	**6**	**6**	N/A
Exactly	**6**	**6**	9.65	**6**	**6**	**6**	**6**	N/A
Exactly2	**10**	10.1	11.95	**10**	**10**	**10**	**10**	N/A
Heart	**6**	6.55	7.65	**6**	**6**	6.1	6.0333	N/A
Vote	**8**	8.05	11.3	**8**	**8**	**8**	8.5	9
Zoo	**5**	5.25	7.95	N/A	N/A	N/A	N/A	**5**
Credit	8.3	9.4	11.3	8.7	8.45	8.6	**8.2**	N/A
Mushroom	**4**	4.45	6.45	4.4	4.15	**4**	**4**	4.35
LED	**5**	5.1	11.9	**5**	**5**	5.5263	**5**	N/A
Letters	**8**	8.45	9.85	8.75	8.15	**8**	**8**	N/A
Derm	**6**	7.4	9.65	**6**	6.3	6.15	6.4	N/A
Derm2	**8.3**	9.45	12.1	8.9	9.1	8.85	8.7	N/A
Chess-king	**29**	29.95	32.15	N/A	N/A	N/A	N/A	**29**
WQ	**12.8**	13.85	15.65	13.4	13.25	13.45	13.2	N/A
Lung	**4**	6.6	12.8	**4**	4.75	**4**	4.7	N/A
Audiology	**7.65**	13.4	22.8	N/A	N/A	N/A	N/A	13.8

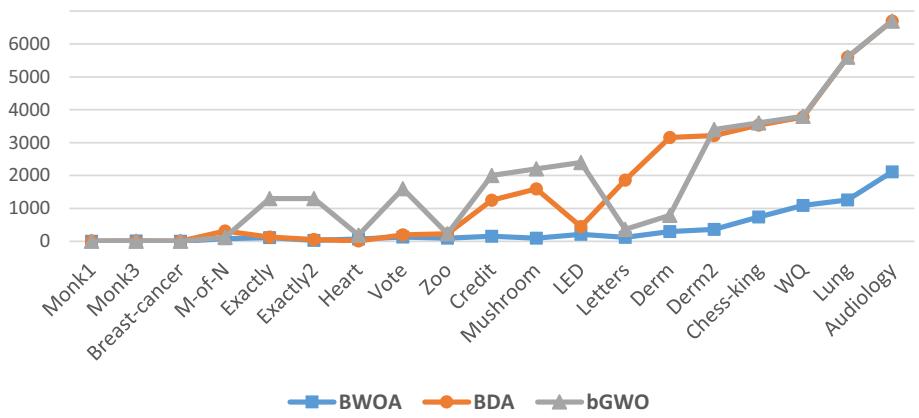

Fig. 4 Rough set minimum function evaluations using the different optimizers

the same minimum reduct. For 15 datasets out of 19, the BWOA can obtain the best minimal reducts in all 20 runs and outperform the other algorithms while SSAR gets eight of the exact minimal reducts of datasets. The best mean value of the reduct attributes for each dataset is presented in Table 5. For 95% of all datasets, BWOA gave the best results, followed by SSAR for 61% out of 13 datasets while BDA achieved 26% of 19 datasets. Concerning the results of the global optimum feature set, BWOA is a consistent and fast algorithm. Also, BWOA provides the best standard deviation and less evaluation.

In general, our results show that BWOA can converge quickly in locating the optimal solution. BWOA has the fastest convergence rate to find the optimal subset or the best reduct for many of datasets within tens or less of iterations overcome other algorithms, see Fig. 4. Moreover, the mean of function evaluation is almost less than half of the maximum evaluation of each dataset, see Figs. 5 and 6. As it is indicated in those figures, BWOA keeps this success in different datasets and also affirms the searching the ability of BWOA.

5.2 Numerical experiments for wrappers approach

In this study, we use a wrapper approach for solving feature selection problem. Three classifiers LR, C4.5 and NB. We show the ability of our proposed algorithm BWOA in solving problems like classification under attribute

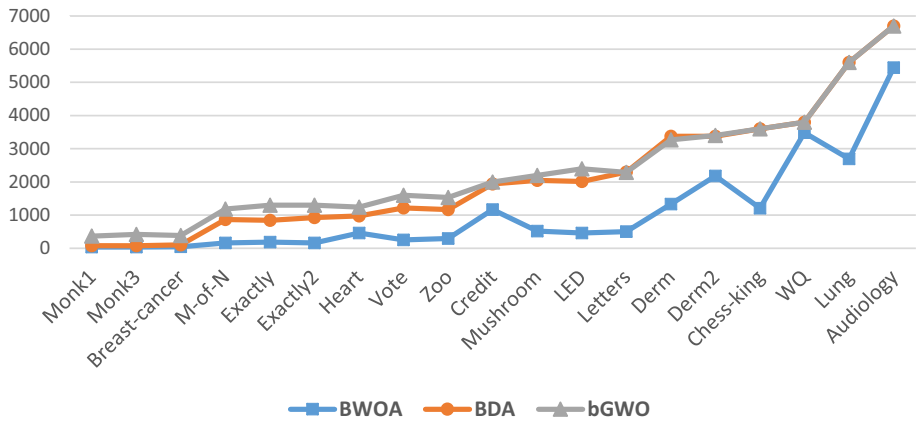

Fig. 5 Rough set mean function evaluations using the different optimizers

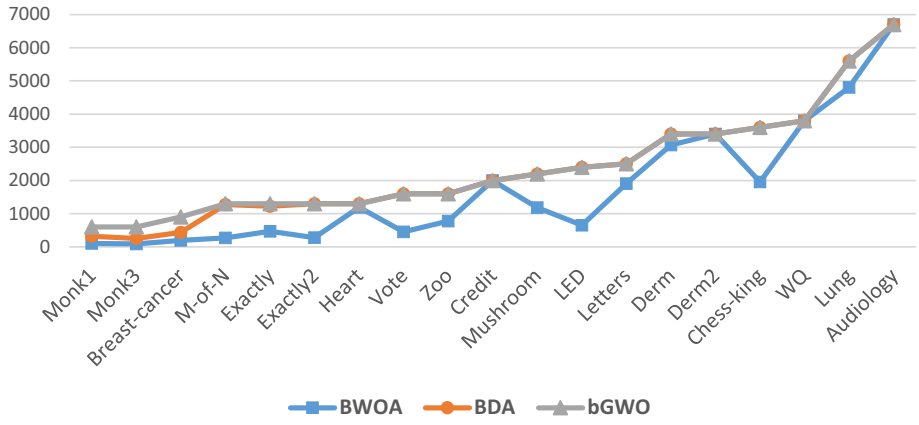

Fig. 6 Rough set maximum function evaluations using the different optimizers

Table 6 Datasets description that used in wrappers experiments

Dataset	Number of features	Number of instances	Number of classes
Breast-cancer	9	699	2
Breast	30	569	2
Congress	16	435	2
Exactly	13	1000	2
Exactly2	13	1000	2
Heart	13	270	2
Ionosphere	34	351	2
Krvskp	36	3196	2
Lymphography	18	148	4
M-of-n	13	1000	2
Penglung	325	73	7
Sonar	60	208	2
Spect	22	267	2
Tic-tac-toe	9	958	2
Vote	16	300	2
Waveform	40	5000	3
Wine	13	178	3
Zoo	16	101	7

reduction (wrappers approach) on 18 UCI databases (details in Table 6). The datasets are selected to have various numbers of attributes and instances as representatives of various kinds of issues that the proposed technique will be tested on.

Two kinds of testing the datasets are considered, *50–50 training-validation* and *10-fold cross-validation*. For *50–50 training-validation* test, for each dataset, the instances are randomly divided into two equal sets (50% each dataset), namely, training and validation. Through the training process, every whale position represents one attribute subset. Training set is used to evaluate the classifier on the validation set throughout the optimization to guide the feature selection process. The test data is kept hidden from the optimization and is left for final evaluation [(training (50%), validation (25%) and test (25%)]. For all algorithms, we repeat the partitioning of each dataset for five times for each solution during the search process to give a chance for selections of different instances/objects of the training and validation sets each time and select the best classification accuracy among them to verify the statistical significance of the results. In the other test, *10-fold cross-validation*, the training set is divided into 10 folds. A single fold is used as the sub-test data, and the remaining 9 folds are used as the

Table 7 Related work for feature selection problem based on wrapper approach

Authors	Algorithms based on wrapper approach
Emarya et al. [36]	In this work, the authors developed a binary version of the grey wolf optimization (GWO) and used to select the optimal feature subset for classification purposes. Grey wolf optimizer (GWO) is one of the latest bio-inspired optimization algorithms, which emulates the hunting process of grey wolves in nature. The authors used a wrapper approach for feature selection based on KNN classifier. We implement this algorithm with various classifiers such as Logistic Regression (LR), Decision Tree Classifier (C4.5) and Naïve Bayes (NB). Moreover, we compare BGWO with our proposed algorithm BWOA
Mafarja and Mirjalili [42]	In this article, the authors suggested a wrapper feature selection approach based on WOA to get the minimal feature subsets. They developed a binary version of WOA and integrated with some of the evolutionary operators (selection, crossover, and mutation) to improve both exploration and exploitation of this algorithm. The results showed that the involvement of the cross over and mutation operators in the WOA algorithm (WOA-CM) outperforms other approaches. Also, the authors tested the efficiency of the proposed WOA-CM algorithm is tested on 20 UCI datasets and compared with three algorithms, namely, GA, PSO, and Ant Lion Optimizer (ALO), and five filter feature selection methods. They used KNN as a classifier in their approach
Mafarja and Mirjalili [43]	In this work, the authors proposed a hybrid Whale Optimization Algorithm (WOA) and Simulated Annealing (SA) algorithms in a wrapper feature selection method, called this algorithm by WOASA. They integrated the SA algorithm with the global search of WOA in order to enhance the exploitation by searching the most promising regions located by WOA algorithm. Also, they checked the performance of their proposed algorithm by applying it on 18 standard benchmark datasets from the UCI repository and compared with three well-known wrapper feature selection methods in the literature. Each solution is computed according to the proposed fitness function, which relies on the KNN classifier
Mirjalili and Yang [54]	In this article, the authors developed a binary bat algorithm (BBA) in order to solve unconstrained optimization problems with binary parameters. They tested the performance of BBA by applying it on twenty-two benchmark functions (unconstrained optimization problems) and comparing it with binary PSO and GA. Also, they showed in their numerical results that BBA outperform other algorithms in the literature on the majority of the benchmark functions. We implement BBA to deal with feature selection problem based on wrapper approach with three different classifiers, such as Logistic Regression (LR), Decision Tree Classifier (C4.5) and Naïve Bayes (NB). Further, we compare BBA with our proposed algorithm BWOA
Mirjalili et al. [55]	In this study, the authors combined both particle swarm optimization (PSO) and gravitational search algorithm (GSA) and called PSOGSA as a hybrid optimization algorithm. This algorithm is suitable for problems with continuous search space. Also, the authors developed a binary version of PSOGSA, denoted by BPSOGSA to deal with problems that have binary parameters. Also, the authors applied both PSOGSA and BPSOGSA on 22 benchmark functions (unconstrained optimization problems). They showed their algorithms outperforms binary gravitational search algorithm (BGSA), binary particle swarm optimization (BPSO), and genetic algorithm. We implement BPSOGSA to solve feature selection problem based on wrapper approach with three different classifiers, such as Logistic Regression (LR), Decision Tree Classifier (C4.5) and Naïve Bayes (NB). Further, we compare BPSOGSA with our proposed algorithm BWOA

sub-training data. This process is repeated ten times with each of the 10-folds as the subtest data. Then the averaged error rates from the ten times are used as the fitness value of the corresponding feature subset (individual) during the evolutionary feature selection process.

We start in Table 7 by summarizing the algorithms from literature that we compare them with our proposed algorithm.

We apply our proposed binary algorithm on 18 datasets and compare with the recent existing binary algorithms in the literature by implementing these algorithms such as BBA [54], bGWO [36], WOA-CM [42], WOASA [43], and BPSOGSA [55]. In Table 2, we outline the parameter settings of BWOA for wrapper approach by considering the same parameters as suggested in the literature, for example, the population size is 8, the maximum number of iterations is 70, and the maximum number of individual runs is 20.

In Tables 8, 9, 10, 11 and 12, the results of datasets show that the proposed algorithm, BWOA, outperforms other algorithms concerning the best, mean, standard deviation, maximum and mean classification accuracy of the results. In each run, we record the following measures from the test data: Getting the minimum (best) fitness function Eq. (21) for 20 runs. According to the statistical results in Table 8, BWOA provides better results of the 18 benchmark datasets on more than 67%, 56% and 50% for LR, C4.5 and NB, respectively, followed in the performance of LR by BBA, bGWO, WOA-CM and BPSOGSA with 17%, 6% and 17%, respectively, while WOASA could not show good performance on this set of datasets. In C4.5, the performance by bGWO, BDA, and BPSOGSA with 44%, 17% and 6%, respectively, while BBA could not show good performance of any dataset. In NB, the performance by BBA, bGWO, BDA, and BPSOGSA are with 17%, 83%, 44%, and 17%, respectively. Therefore, the

Table 8 Best fitness function values obtained over 20 runs for different binary algorithms by using wrapper approach

Dataset	Classifier	50–50 training-validation						10-fold cross-validation					
		BWOA	BBA	bGWO	WOA-CM	WOASA	BPSOGSA	BWOA	BBA	bGWO	WOA-CM	WOASA	BPSOGSA
Breast-cancer	LR	0.031	**0.030**	**0.030**	0.031	0.033	0.032	**0.010**	0.012	0.016	0.016	0.029	0.019
	C4.5	0.017	**0.016**	0.018	0.019	0.025	0.019	0.025	0.027	0.027	**0.024**	0.048	0.033
	NB	**0.031**	**0.031**	**0.031**	0.032	0.035	**0.031**	0.103	0.103	0.103	**0.031**	**0.031**	0.120
Breast	LR	0.017	0.018	0.021	**0.017**	0.023	0.054	0.006	0.009	0.023	**0.005**	0.025	0.032
	C4.5	0.040	0.039	0.044	**0.038**	0.058	0.057	**0.020**	0.026	0.026	0.023	0.060	0.042
	NB	**0.022**	0.026	0.033	0.026	0.031	0.062	**0.017**	0.018	0.028	0.022	0.027	0.059
Congress	LR	0.044	**0.043**	0.046	0.045	0.049	0.052	0.019	**0.010**	0.020	0.015	0.042	0.032
	C4.5	**0.026**	**0.026**	0.032	**0.026**	0.037	0.034	0.011	0.014	0.019	**0.009**	0.049	0.040
	NB	**0.042**	0.046	0.052	0.050	0.056	0.056	0.068	0.068	0.068	**0.052**	0.057	0.091
Exactly	LR	**0.310**	**0.310**	**0.310**	**0.310**	**0.310**	**0.310**	**0.247**	0.255	0.253	0.253	0.293	0.259
	C4.5	**0.005**	**0.005**	**0.005**	**0.005**	0.078	0.292	**0.185**	0.216	0.205	0.217	0.331	0.299
	NB	**0.310**	**0.310**	**0.310**	**0.310**	**0.310**	**0.310**	**0.310**	**0.310**	**0.310**	**0.310**	**0.310**	0.311
Exactly2	LR	**0.241**	**0.241**	**0.241**	**0.241**	**0.241**	**0.241**	**0.182**	0.185	0.184	0.190	0.220	0.194
	C4.5	0.237	**0.236**	0.238	0.239	0.241	0.241	0.203	0.203	**0.201**	0.203	0.247	0.223
	NB	**0.241**	**0.241**	**0.241**	**0.241**	**0.241**	**0.241**	**0.241**	**0.241**	**0.241**	**0.241**	**0.241**	0.241
Heart	LR	0.151	**0.150**	0.152	0.154	0.165	0.186	**0.078**	0.099	0.124	0.083	0.140	0.143
	C4.5	0.152	0.153	**0.150**	0.151	0.180	0.181	**0.133**	0.139	0.136	**0.133**	0.249	0.267
	NB	0.156	0.156	**0.154**	0.157	0.169	0.193	0.165	0.165	0.165	**0.157**	0.163	0.276
Ionosphere	LR	**0.076**	0.080	0.079	0.085	0.102	0.109	**0.043**	0.051	0.057	0.050	0.103	0.069
	C4.5	**0.051**	0.060	0.063	0.058	0.097	0.058	0.061	0.050	0.057	**0.046**	0.109	0.068
	NB	**0.049**	0.051	0.060	0.051	0.074	0.069	0.042	**0.039**	0.045	0.046	0.055	0.069
Krvskp	LR	**0.033**	0.036	0.040	0.039	0.044	0.361	**0.029**	0.032	0.336	0.034	0.042	0.339
	C4.5	**0.014**	0.014	0.015	0.018	0.019	0.348	0.028	**0.027**	0.030	0.030	0.052	0.728
	NB	**0.169**	0.171	0.183	0.178	0.190	0.361	0.050	**0.048**	0.066	0.168	0.199	0.366
Lymph.	LR	0.120	**0.119**	0.121	0.140	0.163	0.190	**0.057**	0.061	0.109	0.071	0.169	0.133
	C4.5	**0.110**	0.118	0.113	0.117	0.179	0.197	**0.111**	0.125	0.115	0.124	0.210	0.425
	NB	0.119	0.120	**0.102**	0.134	0.163	0.210	0.187	0.187	0.188	0.087	**0.082**	0.406
M-of-n	LR	**0.005**	**0.005**	**0.005**	**0.005**	0.005	0.159	**0.005**	**0.005**	0.119	**0.005**	0.005	0.124
	C4.5	**0.005**	**0.005**	0.005	**0.005**	0.005	0.159	0.105	**0.103**	0.104	0.109	0.139	0.165
	NB	**0.150**	**0.150**	0.154	**0.150**	0.155	0.189	0.238	0.238	0.238	**0.151**	0.157	0.284
Penglung	LR	0.015	0.044	0.073	**0.014**	0.083	0.366	**0.002**	0.006	0.294	0.028	0.063	0.339
	C4.5	0.153	0.154	0.183	**0.122**	0.218	0.421	0.271	0.352	0.354	**0.268**	0.518	0.448
	NB	0.070	0.072	0.086	**0.041**	0.111	0.380	**0.002**	0.017	0.018	0.014	0.044	0.501
Sonar	LR	**0.148**	0.160	0.164	0.153	0.197	0.296	**0.090**	0.111	0.210	0.115	0.206	0.261
	C4.5	**0.113**	0.137	0.140	0.117	0.213	0.306	**0.101**	0.109	0.130	0.117	0.233	0.282
	NB	**0.118**	0.133	0.155	0.121	0.181	0.310	**0.029**	0.029	0.049	0.057	0.093	0.292

Table 8 (continued)

Dataset	Classifier	50–50 training-validation						10-fold cross-validation					
		BWOA	BBA	bGWO	WOA-CM	WOASA	BPSOGSA	BWOA	BBA	bGWO	WOA-CM	WOASA	BPSOGSA
Spect	LR	0.128	**0.127**	**0.127**	0.131	0.153	0.173	0.069	**0.057**	0.099	0.084	0.130	0.119
	C4.5	0.129	**0.126**	0.135	0.129	0.163	0.170	**0.161**	0.168	0.163	0.172	0.246	0.343
	NB	**0.135**	**0.135**	0.148	0.153	0.152	0.205	0.205	0.205	0.205	**0.160**	0.163	0.205
Tic-tac-toe	LR	**0.270**	**0.270**	**0.270**	0.273	0.283	0.292	0.226	0.232	0.244	**0.213**	0.275	0.255
	C4.5	0.066	0.064	**0.062**	0.072	0.078	0.133	**0.165**	**0.165**	0.170	0.174	0.218	0.211
	NB	0.283	0.284	**0.282**	0.285	0.289	0.287	0.326	0.326	0.326	**0.286**	**0.286**	0.343
Vote	LR	0.186	0.186	0.379	**0.173**	0.341	0.222	**0.034**	0.091	**0.034**	0.041	0.345	0.113
	C4.5	**0.030**	0.031	0.037	0.036	0.041	0.045	0.018	**0.017**	0.019	**0.017**	0.068	0.060
	NB	**0.042**	0.050	0.054	0.050	0.069	0.057	**0.055**	**0.055**	0.060	0.055	0.061	0.082
Waveform	LR	0.133	**0.133**	0.133	0.139	0.135	0.291	**0.122**	0.123	0.273	0.123	0.135	0.272
	C4.5	0.233	0.232	**0.230**	0.235	0.248	0.382	**0.216**	0.222	0.223	0.224	0.247	0.331
	NB	**0.176**	0.177	0.179	0.180	0.188	0.333	**0.170**	**0.170**	0.174	0.173	0.180	0.337
Wine	LR	**0.005**	**0.005**	**0.005**	0.010	0.022	0.038	**0.003**	0.004	0.004	0.005	0.051	0.016
	C4.5	**0.025**	0.031	0.034	0.032	0.065	0.033	**0.002**	0.004	0.003	**0.002**	0.036	0.033
	NB	**0.004**	0.005	0.005	**0.004**	0.020	0.020	**0.005**	**0.005**	**0.005**	**0.005**	0.006	0.021
Zoo	LR	**0.014**	0.015	0.015	0.023	0.036	0.082	**0.003**	0.004	0.022	0.004	0.029	0.034
	C4.5	**0.023**	**0.023**	**0.023**	0.042	0.042	0.082	**0.003**	**0.003**	0.004	**0.003**	0.029	0.152
	NB	**0.055**	0.055	0.056	0.056	0.065	0.151	**0.045**	**0.045**	0.047	0.055	0.048	0.170

Table 9 Mean fitness function values obtained over 20 runs for different binary algorithms by using wrapper approach

Dataset	Classifier	50–50 training-validation						10-fold cross-validation					
		BWOA	BBA	bGWO	WOA-CM	WOASA	BPSOGSA	BWOA	BBA	bGWO	WOA-CM	WOASA	BPSOGSA
Breast-c	LR	**0.032**	0.032	0.034	0.033	0.039	0.036	**0.016**	0.018	0.019	0.022	0.040	0.019
	C4.5	**0.021**	0.022	0.021	0.023	0.032	0.022	**0.029**	0.032	0.032	0.031	0.072	0.033
	NB	**0.032**	0.034	0.035	0.036	0.039	0.036	0.103	0.103	0.103	**0.035**	0.037	0.120
Breast	LR	**0.021**	0.022	0.025	0.022	0.037	0.057	**0.010**	0.014	0.033	0.011	0.042	0.032
	C4.5	0.045	0.047	0.052	**0.045**	0.074	0.065	0.031	0.034	0.033	**0.031**	0.084	0.042
	NB	**0.026**	0.031	0.041	0.033	0.054	0.068	**0.020**	0.022	0.036	0.028	0.048	0.059
Congress	LR	**0.048**	0.050	0.052	0.053	0.066	0.056	0.026	0.029	0.030	**0.022**	0.075	0.032
	C4.5	**0.030**	0.033	0.036	0.034	0.048	0.039	**0.021**	0.022	0.024	0.025	0.072	0.040
	NB	**0.051**	0.056	0.069	0.058	0.083	0.068	0.072	0.073	0.079	**0.061**	0.079	0.091
Exactly	LR	**0.310**	**0.310**	0.311	**0.310**	0.311	0.311	**0.259**	0.267	0.262	0.262	0.313	0.259
	C4.5	**0.005**	0.140	0.141	0.215	0.203	0.303	**0.222**	0.265	0.247	0.273	0.356	0.299
	NB	**0.310**	**0.310**	0.311	**0.310**	0.310	**0.310**	**0.310**	**0.310**	0.311	**0.310**	0.311	0.311
Exactly2	LR	**0.241**	**0.241**	0.242	**0.241**	0.241	0.241	**0.194**	0.198	0.195	0.195	0.242	0.194
	C4.5	**0.241**	0.241	0.241	0.241	0.241	0.241	**0.212**	0.213	0.214	0.213	0.353	0.223
	NB	**0.241**	**0.241**	0.241	**0.241**	0.241	0.241	**0.241**	**0.241**	0.242	**0.241**	0.241	0.241
Heart	LR	**0.158**	0.159	0.160	0.167	0.180	0.193	**0.104**	0.110	0.140	0.108	0.202	0.143
	C4.5	**0.161**	0.162	0.163	0.170	0.203	0.206	**0.161**	0.170	0.177	0.169	0.332	0.267
	NB	**0.159**	0.164	0.166	0.171	0.194	0.222	**0.168**	0.180	0.190	0.172	0.186	0.276
Ionosphere	LR	**0.087**	0.092	0.092	0.098	0.122	0.115	**0.063**	0.066	0.071	0.068	0.154	0.069
	C4.5	**0.068**	0.074	0.076	0.071	0.119	0.072	0.073	0.072	0.072	0.073	0.177	**0.068**
	NB	**0.057**	0.062	0.068	0.065	0.091	0.085	0.046	**0.045**	0.059	0.055	0.069	0.069
Krvskp	LR	**0.038**	0.052	0.043	0.072	0.052	0.368	**0.034**	0.036	0.342	0.054	0.049	0.339
	C4.5	**0.016**	0.017	0.018	0.043	0.021	0.360	**0.031**	0.033	0.036	0.055	0.069	0.728
	NB	**0.178**	0.184	0.207	0.232	0.267	0.366	0.055	**0.054**	0.084	0.218	0.284	0.366
Lymph.	LR	**0.139**	0.145	0.142	0.154	0.192	0.205	0.087	**0.085**	0.132	0.100	0.264	0.133
	C4.5	**0.144**	0.151	0.146	0.158	0.216	0.211	**0.137**	0.149	0.140	0.152	0.315	0.425
	NB	0.134	0.143	**0.129**	0.161	0.182	0.233	0.187	0.200	0.197	0.115	**0.091**	0.406
M-of-n	LR	**0.005**	0.064	0.029	0.123	0.024	0.182	**0.005**	0.035	0.129	0.070	0.033	0.124
	C4.5	**0.005**	0.011	0.013	0.095	0.024	0.178	0.117	0.120	**0.114**	0.123	0.168	0.165
	NB	**0.152**	0.158	0.169	0.168	0.211	0.221	0.242	0.244	0.244	**0.175**	0.205	0.284
Penglung	LR	0.050	0.066	0.082	**0.048**	0.113	0.411	**0.018**	0.022	0.336	0.053	0.154	0.339
	C4.5	0.201	0.210	0.226	**0.186**	0.350	0.446	0.360	0.388	0.400	**0.330**	0.637	0.448
	NB	0.080	0.084	0.089	**0.074**	0.136	0.404	**0.015**	0.023	0.031	0.040	0.062	0.501
Sonar	LR	**0.168**	0.184	0.180	0.179	0.233	0.322	**0.132**	0.138	0.252	0.155	0.289	0.261
	C4.5	**0.163**	0.168	0.178	0.170	0.270	0.333	**0.143**	0.149	0.151	0.158	0.298	0.282
	NB	**0.151**	0.162	0.178	0.162	0.232	0.325	**0.050**	0.055	0.076	0.082	0.114	0.292

Table 9 (continued)

Dataset	Classifier	50–50 training-validation						10-fold cross-validation					
		BWOA	BBA	bGWO	WOA-CM	WOASA	BPSOGSA	BWOA	BBA	bGWO	WOA-CM	WOASA	BPSOGSA
Spect	LR	**0.132**	0.136	0.139	0.146	0.170	0.190	**0.089**	0.092	0.119	0.115	0.207	0.119
	C4.5	**0.140**	0.144	0.149	0.151	0.193	0.178	0.196	0.199	**0.185**	0.204	0.309	0.343
	NB	**0.152**	0.195	0.193	0.197	0.201	0.205	0.205	0.205	0.206	**0.203**	0.203	0.205
Tic-tac-toe	LR	**0.274**	0.281	0.280	0.288	0.295	0.297	**0.242**	0.244	0.255	0.252	0.302	0.255
	C4.5	0.069	0.069	**0.066**	0.130	0.086	0.190	0.177	0.181	**0.176**	0.189	0.250	0.211
	NB	**0.287**	0.288	0.288	0.292	0.308	0.295	0.327	0.336	0.332	**0.294**	0.298	0.343
Vote	LR	**0.300**	0.314	0.391	0.346	0.409	0.353	0.105	0.254	**0.096**	0.231	0.478	0.113
	C4.5	**0.039**	0.041	0.042	0.045	0.055	0.049	**0.026**	0.027	0.029	0.029	0.100	0.060
	NB	**0.052**	0.059	0.065	0.055	0.094	0.072	**0.059**	0.060	0.070	0.061	0.085	0.082
Waveform	LR	**0.134**	0.135	0.134	0.149	0.139	0.301	0.127	**0.127**	0.277	0.140	0.144	0.272
	C4.5	0.237	0.238	**0.237**	0.246	0.255	0.389	0.228	0.229	**0.227**	0.235	0.260	0.331
	NB	**0.179**	0.181	0.185	0.190	0.195	0.340	**0.172**	0.174	0.180	0.188	0.187	0.337
Wine	LR	**0.015**	0.023	0.019	0.025	0.067	0.052	**0.004**	0.009	0.017	0.018	0.113	0.016
	C4.5	0.045	**0.044**	0.047	0.052	0.085	0.058	**0.009**	0.013	0.011	0.021	0.092	0.033
	NB	**0.008**	0.010	0.012	0.014	0.031	0.035	**0.005**	0.007	0.008	0.010	0.015	0.021
Zoo	LR	**0.023**	0.027	0.032	0.045	0.083	0.137	**0.004**	0.006	0.028	0.016	0.128	0.034
	C4.5	**0.032**	0.042	0.048	0.056	0.077	0.093	**0.004**	0.004	0.006	0.021	0.099	0.152
	NB	**0.061**	0.069	0.072	0.099	0.090	0.210	**0.048**	0.072	0.074	0.082	0.085	0.170

Table 10 Worst fitness function values obtained over 20 runs for different binary algorithms by using wrapper approach

Dataset	Classifier	50–50 training-validation						10-fold cross-validation					
		BWOA	BBA	bGWO	WOA-CM	WOASA	BPSOGSA	BWOA	BBA	bGWO	WOA-CM	WOASA	BPSOGSA
Breast-c	LR	**0.032**	0.035	0.041	0.040	0.050	0.043	**0.020**	0.022	0.022	0.029	0.051	0.022
	C4.5	**0.023**	0.026	0.023	0.026	0.046	0.024	**0.035**	0.046	0.035	0.038	0.118	0.038
	NB	**0.036**	0.038	0.037	0.045	0.051	0.048	0.103	0.103	0.104	**0.038**	0.047	0.151
Breast	LR	**0.025**	0.026	0.031	0.028	0.054	0.059	**0.014**	0.021	0.037	0.017	0.066	0.040
	C4.5	**0.053**	0.055	0.060	0.055	0.084	0.073	**0.038**	0.046	0.041	0.039	0.114	0.054
	NB	**0.030**	0.041	0.046	0.046	0.104	0.076	**0.023**	0.026	0.043	0.034	0.093	0.074
Congress	LR	**0.054**	0.058	0.060	0.058	0.081	0.061	**0.032**	0.037	0.041	0.033	0.122	0.039
	C4.5	**0.034**	0.037	0.040	0.044	0.062	0.043	**0.028**	0.029	0.030	0.036	0.153	0.053
	NB	**0.061**	0.071	0.089	0.076	0.112	0.088	**0.076**	0.088	0.101	0.082	0.124	0.136
Exactly	LR	**0.310**	**0.310**	0.312	**0.310**	0.312	0.312	**0.266**	0.274	0.268	0.270	0.335	0.266
	C4.5	**0.005**	0.310	0.310	0.310	0.308	0.311	**0.270**	0.292	0.294	0.298	0.390	0.338
	NB	**0.310**	**0.310**	0.311	**0.310**	0.311	**0.310**	**0.310**	**0.310**	0.312	**0.310**	0.311	0.311
Exactly2	LR	**0.241**	**0.241**	0.243	**0.241**	0.242	0.242	0.202	0.207	0.202	0.202	0.264	**0.200**
	C4.5	**0.241**	**0.241**	0.243	**0.241**	0.242	0.242	**0.220**	0.224	0.232	0.222	0.992	0.240
	NB	**0.241**	**0.241**	0.242	**0.241**	0.243	0.243	**0.241**	**0.241**	0.243	**0.241**	0.243	0.242
Heart	LR	**0.165**	0.169	0.168	0.187	0.194	0.206	**0.120**	0.128	0.151	0.127	0.252	0.157
	C4.5	**0.168**	0.190	0.178	0.180	0.279	0.237	**0.186**	0.222	0.230	0.234	0.466	0.300
	NB	**0.164**	0.195	0.211	0.181	0.226	0.265	**0.174**	0.214	0.218	0.187	0.252	0.340
Ionosphere	LR	**0.095**	0.107	0.112	0.113	0.139	0.123	**0.077**	0.079	0.081	0.100	0.370	0.081
	C4.5	**0.079**	0.089	0.094	0.086	0.161	0.080	0.084	0.096	0.080	0.098	0.350	**0.080**
	NB	**0.066**	0.073	0.083	0.076	0.116	0.111	**0.051**	0.057	0.069	0.073	0.087	0.092
Krvskp	LR	**0.043**	0.171	0.046	0.246	0.159	0.385	**0.039**	0.049	0.347	0.145	0.058	0.345
	C4.5	**0.019**	0.022	0.032	0.127	0.025	0.381	**0.033**	0.048	0.047	0.148	0.162	0.852
	NB	**0.191**	0.251	0.234	0.281	0.448	0.388	**0.057**	0.059	0.128	0.281	0.523	0.385
Lymph.	LR	**0.160**	0.166	0.173	0.177	0.221	0.242	**0.110**	0.114	0.150	0.122	0.529	0.150
	C4.5	**0.166**	0.192	0.175	0.202	0.266	0.244	0.181	0.218	**0.167**	0.191	0.516	0.538
	NB	**0.147**	0.172	0.149	0.208	0.224	0.276	0.188	0.226	0.228	0.164	**0.129**	0.443
M-of-n	LR	**0.005**	0.168	0.154	0.176	0.176	0.241	**0.006**	0.123	0.135	0.141	0.260	0.129
	C4.5	**0.005**	0.134	0.139	0.176	0.172	0.231	**0.125**	0.143	0.138	0.141	0.256	0.311
	NB	**0.154**	0.173	0.240	0.240	0.303	0.255	0.244	0.261	0.247	**0.240**	0.264	0.304
Penglung	LR	0.084	0.086	0.101	**0.083**	0.143	0.448	**0.033**	0.033	0.375	0.108	0.243	0.375
	C4.5	**0.264**	0.290	0.292	0.272	0.493	0.488	0.406	0.407	0.461	**0.402**	0.759	0.509
	NB	0.098	0.100	0.101	**0.097**	0.169	0.434	**0.029**	0.030	0.033	0.055	0.082	0.583
Sonar	LR	**0.191**	0.204	0.192	0.208	0.257	0.334	**0.157**	0.168	0.277	0.192	0.374	0.277
	C4.5	0.206	**0.195**	0.203	0.220	0.332	0.353	**0.167**	0.176	0.168	0.184	0.361	0.296
	NB	**0.185**	0.195	0.202	0.205	0.280	0.339	**0.067**	0.086	0.102	0.108	0.151	0.306

Table 10 (continued)

Dataset	Classifier	50–50 training-validation						10-fold cross-validation					
		BWOA	BBA	bGWO	WOA-CM	WOASA	BPSOGSA	BWOA	BBA	bGWO	WOA-CM	WOASA	BPSOGSA
Spect	LR	**0.147**	0.154	0.151	0.163	0.213	0.205	0.109	**0.104**	0.133	0.142	0.323	0.133
	C4.5	**0.164**	0.164	0.168	0.173	0.226	0.205	0.226	0.220	**0.207**	0.234	0.385	0.394
	NB	**0.205**	**0.205**	0.208	**0.205**	0.212	0.205	**0.205**	**0.205**	0.208	**0.205**	0.208	0.206
Tic-tac-toe	LR	**0.283**	0.294	0.295	0.297	0.350	0.306	**0.256**	0.263	0.264	0.279	0.326	0.269
	C4.5	**0.071**	0.104	**0.071**	0.207	0.099	0.221	**0.187**	0.196	0.189	0.234	0.293	0.239
	NB	**0.291**	0.304	0.300	0.307	0.345	0.311	0.334	0.345	0.346	0.341	**0.315**	0.346
Vote	LR	**0.384**	**0.384**	0.434	**0.384**	0.609	0.387	0.300	0.316	**0.176**	0.325	0.696	0.206
	C4.5	**0.043**	0.049	0.049	0.051	0.069	0.053	**0.036**	0.037	0.040	0.043	0.224	0.070
	NB	**0.060**	0.075	0.078	0.069	0.137	0.105	**0.061**	0.072	0.093	0.082	0.125	0.095
Waveform	LR	**0.135**	0.142	0.136	0.161	0.155	0.326	**0.134**	0.137	0.281	0.164	0.153	0.278
	C4.5	**0.241**	0.247	0.242	0.257	0.264	0.418	0.234	0.240	**0.231**	0.246	0.290	0.354
	NB	**0.182**	0.187	0.192	0.219	0.211	0.396	**0.176**	0.188	0.184	0.208	0.196	0.365
Wine	LR	**0.023**	0.044	0.036	0.049	0.144	0.082	**0.006**	0.018	0.028	0.036	0.700	0.017
	C4.5	**0.055**	0.061	0.066	0.070	0.109	0.121	**0.018**	0.025	0.029	0.059	0.164	0.048
	NB	**0.017**	0.027	0.025	0.031	0.043	0.061	**0.005**	0.016	0.012	0.017	0.036	0.048
Zoo	LR	**0.034**	0.047	0.048	0.101	0.652	0.366	**0.005**	0.008	0.043	0.043	0.840	0.044
	C4.5	**0.052**	0.062	0.065	0.072	0.121	0.169	**0.006**	**0.006**	0.008	0.083	0.242	0.215
	NB	**0.066**	0.112	0.105	0.169	0.192	0.327	**0.055**	0.102	0.096	0.142	0.221	0.199

Table 11 Standard deviation of the obtained fitness function values over 20 runs for different binary algorithms by using wrapper approach

Dataset	Classifier	50–50 training-validation						10-fold cross-validation					
		BWOA	BBA	bGWO	WOA-CM	WOASA	BPSOGSA	BWOA	BBA	bGWO	WOA-CM	WOASA	BPSOGSA
Breast-c	LR	**0.001**	0.001	0.002	0.002	0.004	0.003	0.003	0.002	**0.002**	0.003	0.006	0.002
	C4.5	0.002	0.003	0.002	0.002	0.005	**0.001**	0.003	0.005	**0.002**	0.004	0.017	0.003
	NB	**0.001**	0.003	0.002	0.004	0.003	0.004	**0.000**	**0.000**	0.001	0.002	0.003	0.009
Breast	LR	0.002	0.003	0.003	0.003	0.007	**0.001**	**0.002**	0.003	0.004	0.004	0.011	0.004
	C4.5	**0.004**	0.005	0.004	0.006	0.007	0.004	0.004	0.005	**0.004**	0.005	0.016	0.004
	NB	**0.002**	0.004	0.003	0.005	0.014	0.004	**0.002**	0.002	0.004	0.003	0.015	0.006
Congress	LR	0.003	0.004	0.004	0.004	0.008	**0.002**	**0.003**	0.007	0.005	0.006	0.020	0.004
	C4.5	0.003	**0.002**	0.002	0.006	0.007	0.003	0.004	0.004	**0.004**	0.007	0.025	0.005
	NB	**0.004**	0.007	0.010	0.008	0.016	0.009	**0.002**	0.005	0.008	0.009	0.019	0.017
Exactly	LR	**0.000**	**0.000**	0.000	**0.000**	0.000	0.000	0.006	0.004	**0.004**	0.004	0.010	0.006
	C4.5	**0.000**	0.154	0.116	0.142	0.063	0.007	0.024	0.023	0.025	0.018	0.019	**0.017**
	NB	**0.000**	**0.000**	0.000	**0.000**	0.000	**0.000**	**0.000**	**0.000**	0.000	**0.000**	0.000	0.000
Exactly2	LR	**0.000**	**0.000**	0.001	**0.000**	0.000	0.000	0.006	0.005	0.005	**0.003**	0.013	0.005
	C4.5	0.001	0.001	0.001	0.000	0.000	**0.000**	**0.005**	0.006	0.008	0.005	0.220	0.010
	NB	**0.000**	**0.000**	0.000	**0.000**	0.000	0.001	**0.000**	**0.000**	0.000	**0.000**	0.000	0.000
Heart	LR	**0.003**	0.004	0.004	0.009	0.007	0.006	0.012	0.009	**0.007**	0.010	0.026	0.009
	C4.5	**0.005**	0.009	0.006	0.008	0.022	0.016	**0.011**	0.021	0.026	0.028	0.059	0.020
	NB	**0.002**	0.009	0.014	0.009	0.017	0.020	**0.003**	0.021	0.023	0.010	0.022	0.035
Ionosphere	LR	0.005	0.008	0.007	0.007	0.009	**0.004**	0.008	0.007	**0.005**	0.011	0.053	0.006
	C4.5	0.007	0.007	0.009	0.007	0.018	**0.007**	0.007	0.011	0.007	0.012	0.056	**0.005**
	NB	**0.005**	0.006	0.005	0.006	0.011	0.011	**0.002**	0.004	0.006	0.006	0.008	0.007
Krvskp	LR	0.002	0.036	**0.002**	0.055	0.025	0.007	**0.002**	0.004	0.003	0.030	0.004	0.004
	C4.5	**0.001**	0.002	0.003	0.033	0.001	0.011	**0.001**	0.006	0.005	0.034	0.023	0.057
	NB	**0.007**	0.017	0.015	0.037	0.055	0.008	**0.002**	0.003	0.019	0.032	0.075	0.005
Lymph.	LR	**0.010**	0.010	0.011	0.011	0.017	0.014	0.012	0.015	0.012	0.012	0.095	**0.010**
	C4.5	0.015	0.018	0.016	0.020	0.025	**0.011**	0.019	0.023	**0.013**	0.019	0.083	0.072
	NB	**0.007**	0.014	0.013	0.019	0.018	0.013	**0.000**	0.017	0.008	0.021	0.010	0.018
M-of-n	LR	**0.000**	0.075	0.054	0.070	0.050	0.034	**0.000**	0.050	0.004	0.061	0.074	0.005
	C4.5	**0.000**	0.029	0.030	0.076	0.050	0.030	**0.006**	0.009	0.008	0.010	0.026	0.050
	NB	**0.002**	0.009	0.019	0.026	0.038	0.015	0.002	0.005	**0.002**	0.027	0.033	0.008
Penglung	LR	0.018	0.013	**0.009**	0.019	0.020	0.020	0.014	**0.013**	0.022	0.021	0.045	0.034
	C4.5	0.025	0.037	0.027	0.030	0.066	**0.018**	0.034	**0.020**	0.027	0.036	0.068	0.027
	NB	0.008	0.008	**0.005**	0.014	0.020	0.018	0.004	0.007	**0.004**	0.014	0.009	0.038
Sonar	LR	0.011	0.010	**0.007**	0.016	0.018	0.009	0.018	0.017	**0.014**	0.020	0.042	0.014
	C4.5	0.022	0.016	0.016	0.022	0.033	**0.013**	0.016	0.018	**0.012**	0.018	0.037	0.012
	NB	0.016	0.014	0.014	0.021	0.028	**0.008**	**0.010**	0.016	0.013	0.018	0.014	0.010

Table 11 (continued)

Dataset	Classifier	50–50 training-validation						10-fold cross-validation					
		BWOA	BBA	bGWO	WOA-CM	WOASA	BPSOGSA	BWOA	BBA	bGWO	WOA-CM	WOASA	BPSOGSA
Spect	LR	**0.004**	0.008	0.007	0.009	0.013	0.011	0.012	0.011	**0.010**	0.015	0.041	0.011
	C4.5	0.010	0.010	0.009	0.013	0.020	**0.009**	0.016	0.013	**0.013**	0.016	0.041	0.036
	NB	0.024	0.023	0.023	0.017	0.013	**0.000**	**0.000**	**0.000**	0.001	0.010	0.009	0.000
Tic-tac-toe	LR	**0.003**	0.008	0.008	0.007	0.014	0.004	0.009	0.008	**0.006**	0.016	0.013	0.008
	C4.5	**0.002**	0.009	0.002	0.037	0.006	0.026	0.006	0.008	**0.004**	0.015	0.021	0.012
	NB	**0.002**	0.004	0.005	0.006	0.013	0.005	**0.002**	0.008	0.007	0.013	0.009	0.003
Vote	LR	0.079	0.071	**0.015**	0.073	0.059	0.046	0.057	0.077	**0.036**	0.095	0.117	0.037
	C4.5	0.003	0.004	0.003	0.006	0.006	**0.003**	0.005	**0.004**	0.005	0.006	0.038	0.007
	NB	**0.005**	0.006	0.007	0.005	0.022	0.013	**0.001**	0.004	0.009	0.006	0.016	0.011
Waveform	LR	**0.000**	0.002	0.001	0.007	0.004	0.009	0.004	0.004	**0.002**	0.011	0.005	0.003
	C4.5	**0.002**	0.004	0.003	0.006	0.005	0.008	0.004	0.004	**0.002**	0.006	0.012	0.006
	NB	**0.002**	0.002	0.003	0.009	0.006	0.013	**0.002**	0.004	0.003	0.010	0.004	0.009
Wine	LR	**0.006**	0.011	0.008	0.012	0.027	0.012	**0.001**	0.005	0.007	0.010	0.140	0.002
	C4.5	0.008	0.009	**0.008**	0.009	0.015	0.022	0.006	**0.006**	0.008	0.013	0.033	0.009
	NB	**0.004**	0.006	0.006	0.007	0.005	0.012	**0.000**	0.003	0.003	0.003	0.007	0.007
Zoo	LR	**0.003**	0.007	0.008	0.021	0.134	0.063	0.001	**0.001**	0.009	0.013	0.172	0.010
	C4.5	**0.009**	0.010	0.013	0.012	0.023	0.021	**0.001**	0.001	0.001	0.021	0.059	0.028
	NB	**0.005**	0.017	0.013	0.030	0.025	0.044	**0.005**	0.019	0.015	0.022	0.041	0.014

Table 12 Average performance of the features selected by different binary algorithms on the test data by using wrapper approach

Dataset	Classifier	50–50 training-validation						10-fold cross-validation					
		BWOA	BBA	bGWO	WOA-CM	WOASA	BPSOGSA	BWOA	BBA	bGWO	WOA-CM	WOASA	BPSOGSA
Breast-c	LR	0.97	**0.97**	0.97	0.97	0.97	0.97	**0.99**	0.99	0.99	0.98	0.97	0.99
	C4.5	0.98	0.98	**0.99**	0.98	0.97	0.98	**0.98**	0.97	0.97	0.97	0.93	0.97
	NB	**0.97**	0.97	0.97	0.97	0.97	0.97	0.90	0.90	0.90	0.97	**0.97**	0.88
Breast	LR	**0.98**	0.98	0.98	0.98	0.97	0.94	**0.99**	0.99	0.97	0.99	0.96	0.97
	C4.5	**0.96**	0.96	0.95	0.96	0.93	0.94	0.97	0.97	**0.97**	0.97	0.92	0.96
	NB	**0.98**	0.97	0.97	0.97	0.95	0.93	**0.98**	0.98	0.97	0.97	0.96	0.94
Congress	LR	**0.95**	0.95	0.95	0.95	0.94	0.95	0.98	0.98	0.97	**0.98**	0.93	0.97
	C4.5	**0.97**	0.97	0.97	0.97	0.96	0.96	**0.98**	0.98	0.98	0.98	0.94	0.96
	NB	**0.95**	0.95	0.94	0.94	0.92	0.93	0.93	0.93	0.93	**0.94**	0.92	0.91
Exactly	LR	**0.69**	**0.69**	**0.69**	**0.69**	**0.69**	**0.69**	**0.74**	0.74	0.74	0.74	0.69	0.74
	C4.5	**1**	0.86	0.86	0.79	0.80	0.70	**0.78**	0.74	0.76	0.73	0.65	0.70
	NB	**0.69**	**0.69**	**0.69**	**0.69**	**0.69**	**0.69**	**0.69**	**0.69**	**0.69**	**0.69**	**0.69**	**0.69**
Exactly2	LR	**0.76**	**0.76**	**0.76**	**0.76**	**0.76**	**0.76**	**0.81**	0.81	0.80	0.81	0.76	0.80
	C4.5	**0.76**	0.76	**0.76**	0.76	0.76	0.76	**0.79**	0.79	0.79	0.79	0.65	0.78
	NB	**0.76**	**0.76**	**0.76**	**0.76**	**0.76**	**0.76**	**0.76**	**0.76**	**0.76**	**0.76**	**0.76**	**0.76**
Heart	LR	**0.85**	0.85	0.85	0.84	0.83	0.81	**0.90**	0.90	0.86	0.89	0.80	0.86
	C4.5	0.84	0.84	**0.84**	0.83	0.80	0.79	**0.84**	0.83	0.83	0.83	0.67	0.73
	NB	**0.84**	0.84	0.84	0.83	0.81	0.78	**0.83**	0.82	0.81	0.83	0.82	0.72
Ionosphere	LR	**0.92**	0.91	0.91	0.90	0.88	0.88	**0.94**	0.94	0.93	0.93	0.85	0.93
	C4.5	**0.94**	0.93	0.93	0.93	0.89	0.93	0.93	0.93	**0.93**	0.93	0.83	0.93
	NB	**0.95**	0.94	0.94	0.94	0.91	0.92	0.96	**0.96**	0.95	0.95	0.94	0.93
Krvskp	LR	**0.97**	0.95	0.97	0.93	0.96	0.63	**0.97**	0.97	0.66	0.95	0.96	0.66
	C4.5	**0.99**	0.99	0.99	0.96	0.99	0.64	**0.97**	0.97	0.97	0.95	0.94	0.27
	NB	**0.82**	0.82	0.80	0.77	0.74	0.63	0.95	**0.95**	0.92	0.78	0.72	0.63
Lymph.	LR	**0.87**	0.86	0.86	0.85	0.81	0.80	0.92	**0.92**	0.87	0.90	0.74	0.87
	C4.5	**0.86**	0.85	**0.86**	0.84	0.79	0.79	**0.87**	0.85	0.87	0.85	0.69	0.57
	NB	0.87	0.86	**0.88**	0.84	0.83	0.77	0.82	0.80	0.81	0.89	**0.92**	0.59
M-of-n	LR	**1**	0.94	0.98	0.88	0.98	0.82	**1**	0.97	0.87	0.93	0.98	0.88
	C4.5	**1**	0.99	0.99	0.91	0.98	0.82	0.89	0.89	**0.89**	0.88	0.84	0.84
	NB	**0.85**	0.84	0.84	0.83	0.79	0.78	0.76	0.76	0.76	**0.83**	0.80	0.72
Penglung	LR	**0.95**	0.94	0.92	**0.95**	0.89	0.58	**0.99**	0.98	0.66	0.95	0.85	0.66
	C4.5	0.80	0.79	0.78	**0.81**	0.65	0.55	0.64	0.61	0.60	**0.67**	0.36	0.55
	NB	0.92	0.92	0.92	**0.93**	0.87	0.59	**0.99**	0.98	0.97	0.96	0.94	0.49
Sonar	LR	**0.83**	0.82	0.82	0.82	0.77	0.68	**0.87**	0.87	0.75	0.85	0.72	0.74
	C4.5	**0.84**	0.83	0.83	0.83	0.73	0.66	**0.86**	0.85	0.85	0.84	0.71	0.72
	NB	**0.85**	0.84	0.83	0.84	0.77	0.67	**0.95**	0.95	0.93	0.92	0.89	0.71
Spect	LR	**0.87**	0.87	0.87	0.86	0.83	0.81	**0.91**	0.91	0.88	0.89	0.80	0.88
	C4.5	**0.86**	0.86	0.86	0.85	0.81	0.82	0.81	0.81	**0.82**	0.80	0.70	0.66
	NB	**0.85**	0.80	0.81	0.80	0.80	0.79	0.79	0.79	0.79	**0.80**	0.80	0.79
Tic-tac-toe	LR	**0.73**	0.72	0.73	0.71	0.71	0.71	**0.76**	0.76	0.75	0.75	0.70	0.75
	C4.5	0.94	0.94	**0.94**	0.88	0.92	0.82	0.83	0.82	**0.83**	0.82	0.76	0.79
	NB	0.72	0.72	**0.72**	0.71	0.70	0.71	0.68	0.67	0.67	**0.71**	0.71	0.66
Vote	LR	**0.70**	0.68	0.61	0.65	0.59	0.65	0.90	0.75	**0.90**	0.77	0.52	0.89
	C4.5	**0.96**	0.96	0.96	0.96	0.95	0.95	**0.98**	0.98	0.98	0.98	0.91	0.94
	NB	**0.95**	0.95	0.94	0.95	0.91	0.93	**0.94**	0.94	0.94	0.94	0.92	0.92
Waveform	LR	0.87	0.87	**0.87**	0.86	0.87	0.70	0.88	**0.88**	0.72	0.87	0.86	0.73
	C4.5	0.77	0.77	**0.77**	0.76	0.75	0.61	0.78	0.77	**0.78**	0.77	0.75	0.67
	NB	**0.83**	0.82	0.82	0.81	0.81	0.66	**0.83**	0.83	0.83	0.81	0.82	0.66

Table 12 (continued)

Dataset	Classifier	50–50 training-validation						10-fold cross-validation					
		BWOA	BBA	bGWO	WOA-CM	WOASA	BPSOGSA	BWOA	BBA	bGWO	WOA-CM	WOASA	BPSOGSA
Wine	LR	**0.99**	0.98	0.99	0.98	0.94	0.95	**1**	1.00	0.99	0.99	0.89	0.99
	C4.5	**0.96**	0.91	0.96	0.95	0.92	0.95	**0.99**	0.99	0.99	0.98	0.92	0.97
	NB	**1.00**	1.00	0.99	0.99	0.98	0.97	**1**	1.00	1.00	0.99	0.99	0.98
Zoo	LR	**0.98**	0.98	0.97	0.96	0.92	0.86	**1**	**1**	0.98	0.99	0.88	0.97
	C4.5	**0.97**	0.96	0.96	0.95	0.93	0.91	**1**	**1**	**1**	0.98	0.91	0.85
	NB	**0.95**	0.94	0.94	0.91	0.92	0.79	**0.96**	0.93	0.93	0.92	0.92	0.83

proposed algorithm has high performance in finding the global solution with minimum reduct and highest classification accuracy of 18 datasets. On the other side of 8, the BBA is the same or slightly better than BWOA for getting the best results using *10-fold cross-validation*.

Finding the average of the solutions from running the optimization algorithm for different 20 runs. In Table 9, the mean values of the fitness function of the proposed algorithm has better performance in both validation testes. These findings prove that the proposed algorithm has the best exploitation ability and convergence rate with 78%, 61% and 56% for LR, C4.5, and NB, respectively, in *50–50 training-validation*. While BBA obtained 17%, 0% and 11%, and WOA-CM obtained 5%, 11% and 39% for LR, C4.5, and NB, respectively. Also, BWOA achieved the best results overcome the other competitive algorithms in *10-fold cross-validation* test with 94%, 56% and 89% for LR, C4.5, and NB, respectively. One can see that the BWOA algorithm can find the best average fitness values that lead in finding a subset feature that has the maximum classification accuracy and the minimum number of selected features based on Eq. (21).

The worst solution among the best solutions is obtained for running an optimization algorithm for 20 runs. According to the results in Table 10, in the first validation test, BWOA shows the best results of the worst fitness functions in fifteen of the benchmark datasets (83% of the datasets /LR). The best results of BBA, bGWO, WOA-CM and BPSOGSA algorithms are given as follows: for WOA-CM for five datasets (28%), and the WOASA algorithm does not provide competitive results in LR and C4.5. 94% is the best results that are achieved for BWOA in *10-fold cross-validation* for each of LR, C4.5, and NB.

These findings prove that the BWOA algorithm can stay in good search space. Also, the results give good evidence for high exploration of BWOA.

In Table 11, Std is a representation for the discrepancy of the acquired best solutions found for running a stochastic optimizer. The success of the BWOA algorithm on average values becomes more evident with its standard deviation. BWOA has the best standard deviations of 20 runs of each data in NB, while bGWO has the best standard deviation in LR and equals or less than BWOA in C.4.5. The situation is more better when it is compared in the second validation test, where BWOA overcomes other algorithms for all considered classifiers.

5.3 Comparison of classification accuracy

Classification average accuracy is a measure to illustrate how precise is the specific classifier for the chosen feature set. We write the classification average accuracy as [36]

$$AvgPerf = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{N}\sum_{i=1}^{N} Match(C_i, L_i)$$

where M is the number of runs and equals to 20, N is the number of points in the test set, C_i is the classifier output label for data point i, L_i is the reference class label for data point i, and Match is a function that produces 1 when the two input labels are the same and produces 0 when they are different. In this work, we focus on the wrapper approach on selecting the optimal feature subset for classification purposes. In Table 12, the results are averaged over all classification accuracy experiments obtained by the proposed algorithm and other algorithms. We apply the classification of the 18 datasets over 20 independent runs to ensure stability and verify the statistical significance of the results.

Table 12 shows the accuracy of the classification performed on the validation set obtained by binary algorithms. In 50–50 training-validation test, BWOA has the best mean accuracy with 78%, 72% and 83% of the 18 datasets for LR, C4.5 and NB, after the number of features has been reduced by wrappers approach. Moreover, the BWOA can obtain the optimal classification accuracy 100% of three datasets (namely, M-of-n, Wine, and Zoo) and overcome all algorithms while bGWO can get high accuracy 99.5% for only one dataset (Zoo). In wrappers experiments using C4.5 and NB, it is also observed that BWOA provides the highest accuracy obtained in most datasets. Whereas, in 10-fold cross-validation test, 83%,

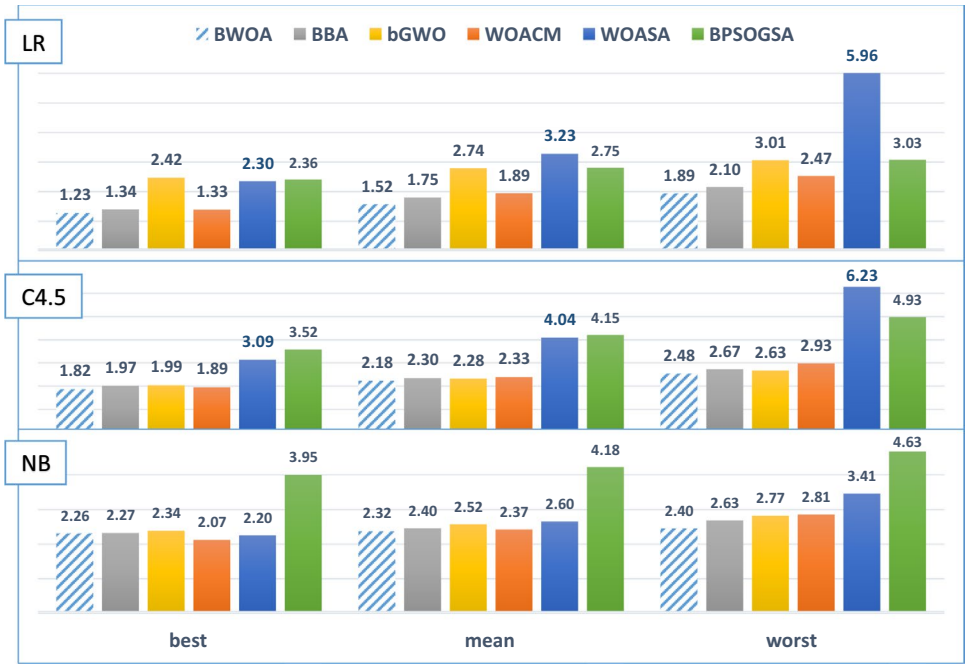

Fig. 7 The total of the best, mean and worst fitness obtained over all 20 runs of 18 datasets from the different algorithms for 50–50 training-validation test

Fig. 8 The total of the best, mean and worst fitness obtained over all 20 runs of 18 datasets from the different algorithms for 10-fold cross-validation test

72% and 89% of the 18 datasets for LR, C4.5 and NB are achieved by BWOA, moreover five of datasets with 100% accuracy.

We note that the best performance by the proposed BWOA is attained in the average, best and worst obtained fitness function value over runs, which demonstrates the ability of the BWOA for exploring the feature space adjustably better than the other algorithms and BWOA is superior compared to other algorithms as shown in Tables 8, 9, 10 and 12.

5.4 Overall performance assessment

We add the total of best, mean and worst fitness obtained over all the datasets from various optimizers, moreover, the total number of all classification accuracies for each

Fig. 9 The total of the classification accuracy obtained over all 20 runs of 18 datasets from the different algorithms for 50–50 training-validation test

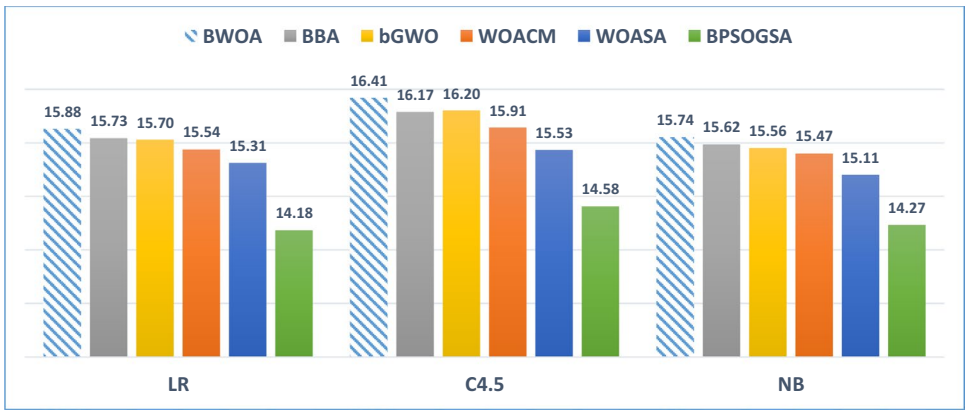

Fig. 10 The total of the classification accuracy obtained over all 20 runs of 18 datasets from the different algorithms for 10-fold cross-validation test

algorithm that found in Tables 8, 9, 10, 11 and 12. Figures 7, 8, 9 and 10 summarize the total of best, mean and worst fitness function values, and the classification accuracy for the both validation testes over all the datasets using various algorithms from the literature. Figures 7 and 8 demonstrate the ability of our proposed algorithm to find the best accuracy and how it performs better than other recent algorithms. The results of total average accuracy show that the performance of the selected features on validation data of BWOA is better than the accuracy of features selected by the competitive algorithms (Figs. 9, 10). The results prove that the BWOA is promising and has merit among other binary metaheuristic optimization algorithms. Also, the results of the BWOA on the feature selection problem show that this algorithm can be useful and efficient for solving other various applications problems.

5.5 Statistical analysis

Two of nonparametric tests, namely, Wilcoxon rank sum test and Friedman's test, are used to determine whether there is a statistically significant difference between our proposed algorithm and other algorithms used in this paper. We conduct the nonparametric Wilcoxon rank sum test [69] at 5% significance level on the results of the proposed algorithm and other comparative algorithms (BBA, bGWO, BDA, and BPSOGSA) on the 18 datasets. Wilcoxon rank sum test is applied to the final results of best, mean, worst results and the average of best classification accuracy as shown in Tables 8, 9, 10, 11 and 12. In Tables 13 and 14, we present the test results and show that the result is significant at p value ≤ 0.05, where a small p value indicates strong evidence against the null hypothesis. The symbols $(+, \cong, -)$ indicate that a given algorithm performs significantly better $(+)$, significantly worse $(-)$, or not significantly different $(\cong)$ compared to BWOA. As illustrated in Table 13, BWOA outperforms all the compared algorithms and finds better solutions of best, mean, worst and the average accuracy (AvgPerf), while in Std it is slightly better or equal to other algorithms in the first validation test. For instance, BWOA outperforms all competitive algorithm of the mean function values using LR and C4.5, in the optimization of 100% over the 18 datasets, while for NB, BWOA performs either the same or slightly better than WOA-CM and WOASA. In 10-fold cross-validation test (Table 14), BWOA outperforms all competitive algorithm of the mean and worst function values of all classifiers, in the optimization of 100% over the 18 datasets, and 100% of average accuracy for LR and C4.5.

Table 13 Statistical comparison of Wilcoxon rank sum test (p-vlaue, h) and Friedman test based (Rank), in which Iman-Davenport's procedure is used as a post hoc procedure ($\alpha = 0.05$) between BWOA and the other four algorithms for 50–50 training-validation test

	Best			Mean			Std			Worst			AvgPerf		
	p values	h	Rank	p values	h	Rank	p values	h	Rank	p values	h	Rank	p values	h	Rank
LR															
BWOA	1	≅	**1.5**	1	≅	**1.2**	1	≅	2.9	1	≅	**1.3**	1	≅	**1.6**
BBA	1.91E−02	+	2.6	4.55E−04	+	2.6	5.00E−01	≅	3.0	7.38E−04	+	2.7	9.88E−03	+	2.7
bGWO	2.93E−04	+	4.4	5.36E−04	+	4.4	7.11E−01	≅	**2.0**	1.85E−03	+	3.9	4.55E−04	+	2.2
WOA-CM	8.61E−03	+	2.8	3.86E−04	+	3.2	2.47E−03	+	4.3	2.93E−04	+	3.8	2.76E−04	+	3.7
WOASA	1.96E−04	+	5.4	1.96E−04	+	5.4	1.96E−04	+	5.8	1.96E−04	+	5.6	1.96E−04	+	5.7
BPSOGSA	7.38E−04	+	4.3	1.96E−04	+	4.2	8.79E−01	≅	2.9	2.14E−03	+	3.8	1.96E−04	+	5.2
C4.5															
BWOA	1	≅	**2.0**	1	≅	**1.5**	1	≅	**2.2**	1	≅	**1.4**	1	≅	**1.3**
BBA	2.15E−02	+	2.6	3.86E−04	+	2.7	9.36E−02	≅	3.0	8.46E−04	+	2.9	4.21E−E−04	+	2.3
bGWO	3.77E−03	+	3.1	1.56E−02	+	2.7	6.47E−01	≅	**2.2**	6.42E−02	+	2.4	2.87E−01	≅	4.5
WOA-CM	7.85E−02	≅	2.5	3.78E−03	+	3.3	1.59E−03	+	4.1	3.86E−04	+	3.6	2.85E−03	+	3.3
WOASA	1.96E−04	+	5.7	1.96E−04	+	5.7	2.33E−04	+	5.7	1.96E−04	+	5.7	1.96E−04	+	5.4
BPSOGSA	3.27E−04	+	5.1	2.76E−04	+	5.1	2.79E−02	+	3.8	2.76E−04	+	4.9	2.33E−04	+	4.3
NB															
BWOA	1	≅	2.7	1	≅	**1.9**	1	≅	**1.3**	1	≅	**1.5**	1	≅	**2.1**
BBA	1.00E+00	≅	2.8	6.10E−04	+	2.7	1.22E−04	+	2.8	1.22E−04	+	2.7	1.22E−03	+	3.0
bGWO	1.95E−03	+	3.9	1.96E−04	+	4.2	3.27E−04	+	3.3	1.96E−04	+	3.9	1.22E−04	+	3.5
WOA-CM	4.89E−01	≅	**2.6**	7.17E−01	≅	2.6	4.38E−04	+	4.3	3.53E−02	+	2.9	6.05E−01	≅	2.9
WOASA	9.59E−01	≅	3.5	2.48E−01	≅	3.9	1.96E−04	+	4.9	1.08E−02	+	4.6	2.78E−01	≅	3.8
BPSOGSA	6.10E−05	+	5.6	1.96E−04	+	5.7	1.96E−04	+	4.4	1.96E−04	+	5.4	6.10v05	+	5.6

Also, we perform multiple comparisons among all the algorithms by using the Friedman's test at 5% significance level, which is a multiple comparisons test that aims to detect significant differences between the properties of two or more algorithms [69]. It is essential to mention that the Friedman's test is applied in this paper by utilizing the KEEL software [70]. The average rankings obtained by Friedman test can be used as indicators to illustrate how successful the algorithm is. In other words, the lower the rank, the more successful the algorithm is. Based on the results in Tables 8, 9, 10, 11 and 12, the results in Tables 13 and 14 give the mean ranks (Rank) among five algorithms obtained by the Friedman's test with a confidence level of 0.95 ($\alpha = 0.05$), in which Iman-Davenport's procedure is used as a post hoc procedure. In Table 13, BWOA exhibits statistically superior performance than the four compared algorithms in the pair-wise Iman-Davenport's procedure at the 95% significance level. It can also be observed that BWOA ranks first according to the best, mean, worst results using LR, C4.5 and NB except in the 'best' results of NB. In Std, BWOA ranked first of C4.5 with bGWO and NB second of LR with BPSOGSA. As for the classification average accuracy, BWOA algorithm has best summary rank using all classifiers compared to other algorithm. In Table 14, BWOA ranks first in all catogries of this test superior other competitive algorithms. Therefore, we can conclude that BWOA outperforms all of the compared algorithms in all comparing categories with different classifiers.

6 Conclusion and future work

Feature selection is an essential task which can particularly determine the performance of recognition and classification. Recently, various evolutionary algorithms have been developed for FS. In this paper, we propose a new FS algorithm based on binary whale optimization algorithm (BWOA). The proposed algorithm has a strong search ability in the problem space and can efficiently get the minimal feature subset. The proposed algorithm is compared with some powerful algorithms, including five rough set FS algorithms and two binary optimization algorithms, namely, ant colony optimization (AntRSAR), simulated annealing (SimRSAR), a rough set approach to FS based on ACO (RSFSACO), tabu search (TSAR), scatter search (SSAR), binary grey wolf optimization (bGWO) and binary dragonfly algorithm (BDA). Also, we apply our proposed FS algorithm via wrapper approach for FS based on three classifiers LR, C4.5 and NB with two kinds

Table 14 Statistical comparison of Wilcoxon rank sum test (p-vlaue, h) and Friedman test based (Rank), in which Iman-Davenport's procedure is used as a post hoc procedure ($\alpha = 0.05$) between BWOA and the other four algorithms for 10-fold cross-validation test

	Best			Mean			Std			Worst			AvgPerf		
	p values	h	Rank	p values	h	Rank	p values	h	Rank	p values	h	Rank	p values	h	Rank
LR															
BWOA	1	≅	**2.2**	1	≅	**1.2**	1	≅	**1.9**	1	≅	**1.2**	1	≅	**1.4**
BBA	3.80E−01	≅	**2.2**	4.38E−04	+	2.7	7.03E−02	≅	3.2	6.10E−05	+	2.5	5.31E−04	+	2.9
bGWO	4.64E−03	+	3.1	1.96E−04	+	3.2	1.02E−01	≅	2.9	1.96E−04	+	3.2	9.35E−04	+	2.7
WOA-CM	2.45E−02	+	3.2	7.76E−04	+	3.4	3.78E−03	+	4.2	1.22E−04	+	3.5	6.10E−05	+	3.9
WOASA	4.38E−04	+	4.8	1.96E−04	+	4.9	1.85E−03	+	5.3	1.96E−04	+	5.2	4.38E−04	+	4.8
BPSOGSA	4.38E−04	+	5.6	1.96E−04	+	5.6	4.75E−02	+	3.4	1.96E−04	+	5.4	4.38E−04	+	5.4
C4.5															
BWOA	1	≅	**1.9**	1	≅	**1.3**	1	≅	**2.1**	1	≅	**1.2**	1	≅	**1.3**
BBA	4.63E−01	≅	2.3	3.27E−04	+	2.4	4.97E−03	+	3.7	1.93E−03	+	2.7	1.96E−04	+	2.8
bGWO	1.30E−02	+	3.2	1.01E−03	+	3.2	3.47E−02	+	2.8	5.99E−04	+	3.1	6.13E−03	+	2.4
WOA-CM	2.15E−02	+	2.9	1.85E−03	+	3.7	4.55E−04	+	4.5	2.93v04	+	3.9	1.18E−03	+	3.8
WOASA	1.96E−04	+	5.3	1.96E−04	+	5.2	2.76E−04	+	5.0	1.96E−04	+	5.0	1.96E−04	+	5.2
BPSOGSA	1.96E−04	+	5.5	1.96E−04	+	5.3	1.70E−01	≅	2.9	1.96E−04	+	5.1	1.96E−04	+	5.4
NB															
BWOA	1	≅	**1.7**	1	≅	**1.3**	1	≅	**1.7**	1	≅	**1.3**	1	≅	**1.4**
BBA	4.88E−04	+	2.7	4.38E−04	+	2.3	3.20E−03	+	2.9	6.10E−05	+	2.5	4.38E−04	+	2.4
bGWO	2.15E−02	+	3.3	5.36E−04	+	3.6	7.40E−03	+	3.2	1.96E−04	+	3.4	1.61E−03	+	3.1
WOA-CM	1.34E−02	+	3.1	9.35v04	+	3.4	2.71E−03	+	4.0	1.22E−04	+	3.2	6.43E−04	≅	3.6
WOASA	4.38E−04	+	5.0	1.96E−04	+	5.2	7.38E−04	+	5.0	1.96E−04	+	5.5	4.38E−04	≅	5.1
BPSOGSA	6.10E−05	+	5.3	2.93E−04	+	5.4	1.68E−02	+	4.1	2.93E−04	+	5.2	4.38E−04	+	5.4

of validation tests (50–50 training-validation and 10-fold cross-validation) on datasets and compare with recent existing algorithms in literature such as BBA, bGWO, WOA with crossover and mutation operators (WOA-CM), the hybrid WOA combining with SA (WOASA) and binary PSO combining with GSA (BPSOGSA).

Moreover, we evaluate the performance of the proposed algorithm by performing our experiments using 32 datasets from the UCI machine learning repository. The experimental results affirm our algorithm and give clear evidence to conclude that our algorithm attains a better feature set regarding the number of selected features and classification accuracy. Finally, BWOA is competitive and promising compared to other algorithms and can serve as a most-suitable preprocessing tool to optimize the feature selection process. As a future work, we would like to focus on the following directions:

- Our proposed algorithm can be applied for selecting a feature subset of images. Also, BWOA can also be extended to hybridize advanced swarm intelligence algorithms such as PSO, DE, GA and cuckoo search optimization.
- It is known that standard rough set theory has some restrictions [71]. Therefore, the authors in [71] developed probabilistic rough set theory, but this conception has not been performed for dimension reduction by other researchers. A filter dimension reduction algorithm using PSO and a probabilistic rough set is developed in [72], and obtained better performance than using PSO and standard rough set. Nevertheless, the proposed algorithm requires to define a parameter to balance the relative significance assigned for the classification performance and the number of attributes, which is problem-dependent and hard to determine in advance. At the same time, because of the limitation that rough set theory only works on discrete data, the datasets used in rough set in recent work [39, 61, 62, 72] only have a small number of attributes. This motivates us to investigate the filter dimension reduction algorithm using BWOA and probabilistic rough set and compare it with the proposed algorithm in this paper.
- The literature is abundant by FS metrics of various nature, for instance, t-score [73] are often utilized in microarrays, and univariate filter metrics P-metric [74, 75]. Nonetheless, they are less effective than multivariate models, some of them are mentioned in the literature as fast correlation-based filter (FCBF) [76], correlation-based feature selection (CFS) [77], ReliefF [78], uncorrelated shrunken centroid (USC) [79] algorithm, minimum

redundancy–maximum relevance (mRMR) [80], etc. We could like to investigate our proposed algorithm based on FS metric such as CSF [77], FCBF [76], USC [79] algorithm, and mRMR [80] on microarray datasets.
– As mentioned by one of the referees it is worth mentioning that we would like to consider the study the analysis and demonstration of space-time complexity. Also, in the first experiment, we would like to study the comparison of classification accuracy.

Acknowledgements We would like to thank the anonymous reviewers for their valuable suggestions and comments to enhance and improve the quality of the paper. The research of the 1st author is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC). The postdoctoral fellowship of the 2nd author is supported by NSERC.

Appendix

1. *Logistic regression (LR)* [48, 49, 49] A common method that is used for classification and known as the exponential or log-linear classifiers. LR is a selective learning classifier that directly estimates the parameters of the posterior distribution function $P(c|x)$. This algorithm assumes the distribution $P(c|x)$ is given by Eq. (22),

$$P(c = k|x) = \frac{\exp\left(w_k^T x\right)}{\sum_{j=1}^{K} \exp\left(w_j^T x\right)} \tag{22}$$

where $w_j s$ are the parameters to estimate and K is the number of classes. Then maximum likelihood method is used to approximate $w_j s$. Since the Hessian matrix for the logistic regression model is positive definite, the error function has a unique minimum. In this proposed system, the LR is used as a classification to ensure the goodness of the selected features. The best feature combination is the one with maximum classification performance and minimum number of selected features. Note that we use the package of Pattern Recognition and Machine Learning Toolbox (PRML) in Matlab which provides logistic regression functions for both binary and multiclass classification problems [81].

2. *C4.5 decision tree classifier* [50, 51] The C4.5 technique is one of the decision tree families that can produce both decision tree and rule-sets, and construct a tree to improve prediction accuracy.

C4.5 uses two heuristic criteria to rank possible tests Information gain that uses attribute selection measure, which minimizes the total entropy of the subset S_i, and the default gain ratio that divides information gain by the information provided by the test outcomes. The information gain algorithm is described as the function gain (A), which is shown below:

– Select the attribute with the highest information gain.
– S contains s_i tuples of class C_i for $i = 1, \ldots, m$.
– Information measure or expected information is required to classify any arbitrary tuple:

$$I(S_1, S_2, \ldots, S_m) = -\sum_{i=1}^{m} \frac{S_i}{S} \log_2 \frac{S_i}{S}. \tag{23}$$

– Entropy of attribute A with values $a_1, a_2, \ldots, a_v$:

$$E(A) = \sum_{j=1}^{v} \frac{S_{1j} + \cdots + S_{mj}}{S} I(S_{1j}, \ldots, S_{mj}). \tag{24}$$

– Information gain means how much can be gained by branching on attribute A:

$$Gain(A) = I(S_1, S_2, \ldots, S_m) - E(A). \tag{25}$$

3. *Naïve Bayes (NB)* [52, 53] Naïve Bayes has proven to be a simple, useful, and powerful machine learning approach in classification studies. NB is recognized as a simple Bayesian classification algorithm. NB classifier is highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be employed y calculating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

References

1. Joachims T (1998) Text categorization with support vector machines: learning with many relevant features. European conference on machine learning. Springer, New York, pp 137–142
2. Yang Y, Pedersen JO (1997) A comparative study on feature selection in text categorization. In: Proceedings of the Fourteenth International Conference on Machine Learning, ICML '97. Morgan Kaufmann Publishers Inc., San Francisco, pp 412–420. http://dl.acm.org/citation.cfm?id=645526.657137
3. Jain A, Zongker D (1997) Feature selection: evaluation, application, and small sample performance. IEEE Trans Pattern Anal Mach Intell 19(2):153–158
4. Mitra P, Murthy C, Pal SK (2002) Unsupervised feature selection using feature similarity. IEEE Trans Pattern Anal Mach Intell 24(3):301–312
5. Rui Y, Huang TS, Chang S-F (1999) Image retrieval: current techniques, promising directions, and open issues. J Vis Commun Image Represent 10(1):39–62
6. Saeys Y, Inza I, Larrañaga P (2007) A review of feature selection techniques in bioinformatics. Bioinformatics 23(19):2507–2517
7. Model F, Adorjan P, Olek A, Piepenbrock C (2001) Feature selection for dna methylation based cancer classification. Bioinformatics 17(suppl 1):S157–S164

8. Dash M, Liu H (2003) Consistency-based search in feature selection. Artif Intell 151(1–2):155–176
9. Jensen R (2005) Combining rough and fuzzy sets for feature selection, Ph.D. thesis, Citeseer
10. Liu H, Motoda H (1998) Feature extraction, construction and selection: a data mining perspective, vol 453. Springer, New York
11. Somol P, Pudil P, Kittler J (2004) Fast branch & bound algorithms for optimal feature selection. IEEE Trans Pattern Anal Mach Intell 26(7):900–912
12. Zhong N, Dong J, Ohsuga S (2001) Using rough sets with heuristics for feature selection. J Intell Inf Syst 16(3):199–214
13. Lai C, Reinders MJ, Wessels L (2006) Random subspace method for multivariate feature selection. Pattern Recogn Lett 27(10):1067–1076
14. Modrzejewski M (1993) Feature selection using rough sets theory. European Conference on Machine Learning. Springer, New York, pp 213–226
15. Neumann J, Schnörr C, Steidl G (2005) Combined svm-based feature selection and classification. Mach Learn 61(1–3):129–150
16. Gasca E, Sánchez JS, Alonso R (2006) Eliminating redundancy and irrelevance using a new mlp-based feature selection method. Pattern Recogn 39(2):313–315
17. Xie Z-X, Hu Q-H, Yu D-R (2006) Improved feature selection algorithm based on svm and correlation. International symposium on neural networks. Springer, New York, pp 1373–1380
18. Dash M, Liu H (1997) Feature selection for classification. Intell Data Anal 1(1–4):131–156
19. Fodor IK (2002) A survey of dimension reduction techniques, Center for Applied Scientific Computing, Lawrence Livermore National. Laboratory 9:1–18
20. Neshatian K, Zhang M (2009) Genetic programming for feature subset ranking in binary classification problems. European conference on genetic programming. Springer, New York, pp 121–132
21. Zhu Z, Ong Y-S, Dash M (2007) Wrapper-filter feature selection algorithm using a memetic framework. IEEE Trans Syst Man Cybern Part B 37(1):70–76
22. Huang J, Cai Y, Xu X (2007) A hybrid genetic algorithm for feature selection wrapper based on mutual information. Pattern Recogn Lett 28(13):1825–1844
23. Chen S-C, Lin S-W, Chou S-Y (2011) Enhancing the classification accuracy by scatter-search-based ensemble approach. Appl Soft Comput 11(1):1021–1028
24. Jue W, Qi Z, Hedar A, Ibrahim AM (2014) A rough set approach to feature selection based on scatter search metaheuristic. J Syst Sci Complex 27(1):157–168. https://doi.org/10.1007/s11424-014-3298-z
25. Lin S-W, Lee Z-J, Chen S-C, Tseng T-Y (2008) Parameter determination of support vector machine and feature selection using simulated annealing approach. Appl Soft Comput 8(4):1505–1512
26. Hedar A-R, Ibrahim A-MM, Abdel-Hakim AE, Sewisy AA (2018) Modulated clustering using integrated rough sets and scatter search attribute reduction. In: Proceedings of the genetic and evolutionary computation conference companion, GECCO '18. ACM, New York, pp 1394–1401. https://doi.org/10.1145/3205651.3208286
27. Tabakhi S, Moradi P, Akhlaghian F (2014) An unsupervised feature selection algorithm based on ant colony optimization. Eng Appl Artif Intell 32:112–123
28. Yusta SC (2009) Different metaheuristic strategies to solve the feature selection problem. Pattern Recogn Lett 30(5):525–534
29. Hedar A, Wang J, Fukushima M (2008) Tabu search for attribute reduction in rough set theory. Soft Comput 12(9):909–918
30. Al-Ani A, Alsukker A, Khushaba RN (2013) Feature subset selection using differential evolution and a wheel based search strategy. Swarm Evol Comput 9:15–26
31. Khushaba RN, Al-Ani A, Al-Jumaily A (2011) Feature subset selection using differential evolution and a statistical repair mechanism. Expert Syst Appl 38(9):11515–11526
32. Rodrigues D, Pereira LA, Nakamura RY, Costa KA, Yang X-S, Souza AN, Papa JP (2014) A wrapper approach for feature selection based on bat algorithm and optimum-path forest. Expert Syst Appl 41(5):2250–2258
33. Yazdani S, Shanbehzadeh J, Aminian E (2013) Feature subset selection using constrained binary/integer biogeography-based optimization. ISA Transa 52(3):383–390. 10.1016/j.isatra.2012.12.005. http://www.sciencedirect.com/science/article/pii/S0019057812001991
34. Chuang L-Y, Yang C-H, Li J-C (2011) Chaotic maps based on binary particle swarm optimization for feature selection. Appl Soft Comput 11(1):239–248
35. Inbarani HH, Azar AT, Jothi G (2014) Supervised hybrid feature selection based on pso and rough sets for medical diagnosis. Comput Methods Programs Biomed 113(1):175–185
36. Emarya E, Zawbaa HM, Hassanien AE (2016) Binary grey wolf optimization approaches for feature selection. Neurocomputing 172:371–381
37. Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Appl 27(4):1053–1073. https://doi.org/10.1007/s00521-015-1920-1
38. Pawlak Z (1991) Rough sets: theoretical aspects of reasoning about data. Kluwer Academic Publishers, Dordrecht
39. Wang X, Yang J, Teng X, Xia W, Jensen R (2007) Feature selection based on rough sets and particle swarm optimization. Pattern Recogn Lett 28(4):459–471. 10.1016/j.patrec.2006.09.003. http://www.sciencedirect.com/science/article/pii/S0167865506002327
40. Polkowski L, Tsumoto S, Lin TY (2000) Rough set methods and applications: new developments in knowledge discovery in information systems, vol 56 of studies in fuzziness and soft computing. Physica-Verlag, Heidelberg
41. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67. 10.1016/j.advengsoft.2016.01.008. http://www.sciencedirect.com/science/article/pii/S0965997816300163
42. Mafarja M, Mirjalili S (2018) Whale optimization approaches for wrapper feature selection. Appl Soft Comput 62:441–453. 10.1016/j.asoc.2017.11.006. http://www.sciencedirect.com/science/article/pii/S1568494617306695
43. Mafarja MM, Mirjalili S (2017) Hybrid whale optimization algorithm with simulated annealing for feature selection. Neurocomputing 260:302–312. 10.1016/j.neucom.2017.04.053. http://www.sciencedirect.com/science/article/pii/S092523121730807X
44. Eid HF (2018) Binary whale optimisation: an effective swarm algorithm for feature selection. Int J Metaheuristics 7(1):67–79. https://doi.org/10.1504/IJMHEUR.2018.091880
45. Ke L, Feng Z, Ren Z (2008) An efficient ant colony optimization approach to attribute reduction in rough set theory. Pattern Recogn Lett 29(9):1351–1357
46. Jensen R, Shen Q (2004) Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches. IEEE Trans Knowl Data Eng 16(12):1457–1471
47. Yumin C, Duoqian M, Ruizhi W (2010) A rough set approach to feature selection based on ant colony optimization. Pattern Recogn Lett 31(3):226–233. 10.1016/j.patrec.2009.10.013. http://www.sciencedirect.com/science/article/pii/S0167865509002888
48. Le Cessie S, Van Houwelingen JC (1992) Ridge estimators in logistic regression. Appl Stat 41:191–201
49. Hosmer D, Lemeshow S, Sturdivant R (2013) Applied logistic regression, Wiley Series in Probability and Statistics, Wiley. https://books.google.ca/books?id=bRoxQBIZRd4C

50. Salzberg SL (1994) C4.5: programs for machine learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. Mach Learn 16(3):235–240. https://doi.org/10.1007/BF00993309
51. Jantan H, Hamdan AR, Othman ZA (2010) Human talent prediction in hrm using c4.5 classification algorithm. Int J Comput Sci Eng 2(8):2526–2534
52. Lewis DD (1998) Naive (bayes) at forty: the independence assumption in information retrieval. In: Nédellec C, Rouveirol C (eds) Machine learning: ECML-98. Springer, Berlin, Heidelberg, pp 4–15
53. Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning. Springer, New York. https://doi.org/10.1007/978-0-387-84858-7
54. Mirjalili SM, Yang X-S (2014) Binary bat algorithm. Neural Comput Appl 25(3):663–681. https://doi.org/10.1007/s00521-013-1525-5
55. Mirjalili S, Wang GG, Coelho LS (2014) Binary optimization using hybrid particle swarm optimization and gravitational search algorithm. Neural Comput Appl 25(6):1423–1435
56. Kaveh A, Ghazaan MI (2017) Enhanced whale optimization algorithm for sizing optimization of skeletal structures. Mech Based Design Struct Mach 45(3):345–362. https://doi.org/10.1080/15397734.2016.1213639
57. Mirjalili S, Lewis A (2013) S-shaped versus v-shaped transfer functions for binary particle swarm optimization. Swarm Evol Comput 9:1–14. 10.1016/j.swevo.2012.09.002. http://www.sciencedirect.com/science/article/pii/S2210650212000648
58. Inbarani H, Bagyamathi M, Azar A (2015) A novel hybrid feature selection method based on rough set and improved harmony search. Neural Comput Appl 26(8):1859–1880. https://doi.org/10.1007/s00521-015-1840-0
59. Swiniarski R, Skowron A (2003) Rough set methods in feature selection and recognition. Pattern Recogn Lett 24(6):833–849. 10.1016/S0167-8655(02)00196-4. http://www.sciencedirect.com/science/article/pii/S0167865502001964
60. Nakamura RYM, Pereira LAM, Costa KA, Rodrigues D, Papa JP, Yang XS (2012) Bba: a binary bat algorithm for feature selection. In: 2012 25th SIBGRAPI conference on graphics, patterns and images, pp 291–297. https://doi.org/10.1109/SIBGRAPI.2012.47
61. Ming H (2008) A rough set based hybrid method to feature selection. Int Symp Knowl Acquis Model 2008:585–588. https://doi.org/10.1109/KAM.2008.12
62. Bae C, Yeh W-C, Chung YY, Liu S-L (2010) Feature selection with intelligent dynamic swarm and rough set. Expert Syst Appl 37(10):7026–7032
63. Pawlak Z (1997) Rough set approach to knowledge-based decision support. Eur J Oper Res 99(1):48–57. 10.1016/S0377-2217(96)00382-7. http://www.sciencedirect.com/science/article/pii/S0377221796003827
64. Manish S (2002) Rough-fuzzy functions in classification. Fuzzy Sets Syst 132:353–369
65. Chen Y, Miao D, Wang R, Wu K (2011) A rough set approach to feature selection based on power set tree. Knowl Based Syst 24(2):275–281. 10.1016/j.knosys.2010.09.004. http://www.sciencedirect.com/science/article/pii/S0950705110001498
66. Kohavi R, Sommerfield D (1995) Feature subset selection using the wrapper method: overfitting and dynamic search space topology. In: KDD, pp 192–197
67. Frank A, Asuncion A (2010) UCI machine learning repository. http://archive.ics.uci.edu/ml/index.php
68. Chen Y, Miao D, Wang R (2010) A rough set approach to feature selection based on ant colony optimization. Pattern Recogn Lett 31(3):226–233
69. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1(1):3–18. 10.1016/j.swevo.2011.02.002. http://www.sciencedirect.com/science/article/pii/S2210650211000034
70. Alcala-Fdez J et al (2011) Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. J Mult Valued Logic Soft Comput 17(2-3):255–287. http://www.keel.es/
71. Yao Y, Zhao Y (2008) Attribute reduction in decision-theoretic rough set models. Inf Sci 178(17):3356–3373
72. Cervante L, Xue B, Shang L, Zhang M (2013) Binary particle swarm optimisation and rough set theory for dimension reduction in classification. IEEE Congr Evol Comput 2013:2428–2435. https://doi.org/10.1109/CEC.2013.6557860
73. Li W, Yang Y (2002) How many genes are needed for a discriminant microarray data analysis. Methods of microarray data analysis. Springer, New York, pp 137–149
74. Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA et al (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286(5439):531–537
75. Hwang K-B, Cho D-Y, Park S-W, Kim S-D, Zhang B-T (2002) Applying machine learning techniques to analysis of gene expression data: cancer diagnosis. Methods of microarray data analysis. Springer, New York, pp 167–182
76. Yu L, Liu H (2003) Feature selection for high-dimensional data: A fast correlation-based filter solution. In: Proceedings of the 20th international conference on machine learning (ICML-03), pp 856–863
77. Hall MA (1999) Correlation-based feature selection for machine learning. University of Waikato, Hamilton
78. Wang Y, Makedon F (2004) Application of relief-f feature filtering algorithm to selecting informative genes for cancer classification using microarray data. In: Computational systems bioinformatics conference. CSB 2004. Proceedings. 2004 IEEE. IEEE, pp 497–498
79. Tibshirani R, Hastie T, Narasimhan B, Chu G (2002) Diagnosis of multiple cancer types by shrunken centroids of gene expression. Proc Nat Acad Sci 99(10):6567–6572
80. Ding C, Peng H (2005) Minimum redundancy feature selection from microarray gene expression data. J Bioinform Comput Biol 3(02):185–205
81. Chen M (2016) Pattern recognition and machine learning toolbox. http://www.mathworks.com/matlabcentral/fileexchange/55826-pattern-recognition-and-machine-learning-toolbox

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.