Expert Systems With Applications 139 (2020) 112824

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection

Mohamed Abdel-Basset[a,*], Doaa El-Shahat[b], Ibrahim El-henawy[b], Victor Hugo C. de Albuquerque[c], Seyedali Mirjalili[d]

[a] Faculty of Computers and Informatics, Department of Operations Research, Zagazig University, Egypt
[b] Computer Science Department, Faculty of Computers and Informatics, Zagazig University, Egypt
[c] University of Fortaleza, Fortaleza, Ceará, Brazil
[d] Torrens University Australia, 90 Bowen Terrace, Fortitude Valley, Queensland 4006, Australia

ARTICLE INFO

Article history:
Received 6 April 2019
Revised 17 July 2019
Accepted 18 July 2019
Available online 27 July 2019

Keywords:
Feature selection
Grey wolf optimization algorithm
Wrapper method
Classifier, accuracy
Cross-validation
Mutation

ABSTRACT

Because of their high dimensionality, dealing with large datasets can hinder the data mining process. Thus, the feature selection is a pre-process mandatory phase for reducing the dimensionality of datasets through using the most informative features and at the same time maximizing the classification accuracy. This paper proposes a new Grey Wolf Optimizer algorithm integrated with a Two-phase Mutation to solve the feature selection for classification problems based on the wrapper methods. The sigmoid function is used to transform the continuous search space to the binary one in order to match the binary nature of the feature selection problem. The two-phase mutation enhances the exploitation capability of the algorithm. The purpose of the first mutation phase is to reduce the number of selected features while preserving high classification accuracy. The purpose of the second mutation phase is to attempt to add more informative features that increase the classification accuracy. As the mutation phase can be time-consuming, the two-phase mutation can be done with a small probability. The wrapper methods can give high-quality solutions so we use one of the most famous wrapper methods which called k-Nearest Neighbor (k-NN) classifier. The Euclidean distance is computed to search for the k-NN. Each dataset is split into training and testing data using K-fold cross-validation to overcome the overfitting problem. Several comparisons with the most famous and modern algorithms such as flower algorithm, particle swarm optimization algorithm, multi-verse optimizer algorithm, whale optimization algorithm, and bat algorithm are done. The experiments are done using 35 datasets. Statistical analyses are made to prove the effectiveness of the proposed algorithm and its outperformance.

1. Introduction

Recently, data mining has become an indispensable field. It has been at the forefront of many applications in the areas of telecommunications, financial data, biological data analysis, etc. The purpose of data mining is to acquire and use the knowledge that flows through handling huge dimensions of datasets. Dimensionality is one of the biggest problems that can hinder the process of data mining due to the high computational costs it needs. Each dataset consists of a set of samples which represents information about a specific case given in the form of features. The problem of the dataset is not limited to its massive dimensionality but also containing irrelevant and redundant features. In addition, the collected datasets may be accompanied by a high level of noise. The traditional machine learning methods are not able to handle this huge number of features. It can suffer from local optima. Hence the need for feature selection has become a necessity as a preprocessing phase to reduce the dimensionality of data and eliminate the duplication and unwanted features in the data. The main goal of feature selection is to find the best subset of informative features while preserving high classification accuracy in representing the original dataset features. The process of assigning each sample to a certain class is known as classification.

There are many feature selection methods that contribute to selecting the most effective features from the original dataset features. Feature selection methods can be categorized into wrapper, filter, and embedded. In wrapper methods, a learning algorithm or classification algorithm is used to evaluate the selected subset of features. Filter method evaluates the selected subset of features de-

* Corresponding author.
E-mail addresses: mohamed.abdelbasset@fci.zu.edu.eg (M. Abdel-Basset), doaazidan@zu.edu.eg (D. El-Shahat), ielhenawy@zu.edu.eg (I. El-henawy), victor.albuquerque@unifor.br (Victor Hugo C. de Albuquerque).

https://doi.org/10.1016/j.eswa.2019.112824
0957-4174/

pending on the characteristics of the data. Compared to the wrapper method, filter methods are faster as the wrapper method waste time in measuring the classifier accuracy. On the other hand, the wrapper method can obtain a good subset of features that suit the classifier than the filter method. Hybrid methods have the advantage of low computational cost but are usually specific to machine learning. Feature selection has to satisfy two objectives: minimizing the number of features selected and maximizing the classification accuracy rate.

Several limitations can face us when trying to find the best subset of features. Many search techniques such as random search, breadth search, depth search, or hybrid are used to find the best selected subset of features. But the exhaustive search is not acceptable as for large datasets, if we assume that the size of features is d, it would be practically impossible to choose the best subset of features among 2^d solutions. In recent years, we have witnessed the emergence of metaheuristics that mostly imitate the collective behavior of organisms. These algorithms have made great progress especially in the area of optimization. Metaheuristic algorithms can be the best choice as it can obtain good solutions in a reasonable time. They can be a good alternative to alleviate the drawbacks of an exhaustive search that needs high computational cost and may be time-consuming. On the other hand, most of the metaheuristic algorithms suffer from local optimum, lack of diversity of the search, and imbalance between the explorative or the exploitive capabilities of the algorithm.

For the aforementioned challenges that can be a stumbling block in the way of finding a good solution to the feature selection. This motivates us to propose a method for tackling the feature selection problems based on grey wolf optimizer algorithm. The grey wolf optimizer algorithm showed a resounding success in many applications Faris, Aljarah, Al-Betar, and Mirjalili (2018a); Heidari and Pahlavani (2017); Lu, Gao, Li, and Xiao (2017); Sanjay, Jayabarathi, Raghunathan, Ramesh, and Mithulananthan (2017); Venkatakrishnan, Rengaraj, and Salivahanan (2018). In order to rectify the problem of local optima and lack of diversity, a two-phase mutation operation is effectively incorporated with the proposed algorithm (TMGWO). Also, the mutation operation is characterized by its ability to improve the exploitation of this algorithm. The main contributions of this paper are:

- A new modified version of the grey wolf algorithm is adapted. Starting from we deal with a binary problem, two different transformation functions are used.
- The effect of the two transformation functions is investigated on thirty-five datasets.
- A bi-phase mutation operator is employed for enhancing the performance of the proposed algorithm.
- The mutation operator is done with a probability. The results revealed that if the mutation probability increases, the classification accuracy raises.
- TMGWO algorithm is compared with other metaheuristics and the experimental results have revealed that TMGWO presented superior results compared to other metaheuristic algorithms.
- A number of comparisons and statistical analyses are done such as best fitness, mean fitness, worst fitness, classification accuracy, standard deviation (std), time, and drawing charts and boxplots.

This paper is organized as follows. Section 2 explores an overview of the related work done in feature selection. Section 3 provides a description of the standard grey wolf optimizer algorithm. In Section 4, the proposed algorithm (TMGWO) is more investigated and elaborated. The numerical and comparison results are given in Section 5. Finally, some conclusions are introduced and some future works are suggested in Section 6.

2. Related work

Metaheuristics have demonstrated their effectiveness in solving many applications. Feature selection is a multi-objective optimization problem that aims to maximize the classification accuracy and minimize the number of selected features. Several metaheuristics are adopted that solve the feature selection problems, and some of them will be surveyed. Sharawi, Zawbaa, and Emary (2017)) suggested a version of Whale Optimization Algorithm (WOA) that makes use of the wrappers method to find the optimal subset of features. Subsequently, Eid (2018) introduced the sigmoid function to WOA for solving the feature selection problems. With the simulated annealing, the whale algorithm is hybridized to enhance the exploitation capability of WOA as in Mafarja and Mirjalili (2017). The simulated annealing was integrated into WOA in order to improve the best solution found after each iteration. The same authors (Mafarja & Mirjalili, 2018) provided two variants of the WOA algorithm. In the first variant, the roulette wheel and tournament selection are used instead of the random operator. To increase the performance of the proposed algorithm, the crossover and mutation operators are used. Furthermore, Sayed, Darwish, and Hassanien (2018) used the chaotic search in conjunction with WOA to combat the local optima stagnation and slow convergence speed problems that encounter the feature selection problems. Moreover, a filter algorithm based Pearson correlation coefficient and correlation distance is used with WOA (Zheng et al., 2018).

Mafarja et al. (2019) proposed binary variants of grasshopper algorithm (BGOA) based on two mechanisms. The first mechanism employs the sigmoid function and V-shaped function as transformation functions. The second mechanism uses the mutation operator to support the exploration phase of BGOA. Mafarja et al. (2018) improved the performance of the conventional grasshopper algorithm through evolutionary population dynamics and selection operators. Zakeri and Hokmabadi (2019) supplemented the grasshopper algorithm with statistical measures to replace the duplicated features with the most promising features.

Emary, Zawbaa, and Hassanien (2016a) investigated two binary variants of the lion algorithm that use either the S-shaped function or the V-shaped function. After that, Mafarja, Eleyan, Abdullah, and Mirjalili (2017) were interested in studying the effect of different transformation function on the lion algorithm. Six variants of lion algorithm are proposed based on S-shaped function and the V-shaped function. Also, a new binary version of the butterfly algorithm based on the previous transformation functions is proposed as in Arora and Anand (2019). A binary version of the grey wolf optimizer is provided as can be found in Emary, Zawbaa, and Hassanien (2016b) and Emary, Zawbaa, Grosan, and Hassenian (2015). Chen, Zhou, and Yuan (2019) have utilized the logistic map sequence with particle swarm optimization algorithm to increase the search space diversity. De Souza, dos Santos Coelho, De Macedo, and Pierezan (2018) introduced a crow search algorithm using a V-shaped transformation function.

Sayed, Nabil, and Badr (2016) integrated the flower algorithm with the clonal selection search. The accuracy of the optimum path forest is used to evaluate the performance of the solutions. In addition, the flower algorithm is hybridized with a rough set (Zawbaa, Hassanien, Emary, Yamany, & Parv, 2015). Sayed, Khoriba, and Haggag (2018) and Sayed, Hassanien, and Azar (2017) introduced the chaos maps to crow search and the salp swarm algorithms in which the performance of 10 chaos maps was investigated as well. Hegazy, Makhlouf, and El-Tawel (2018) added a new control parameter to adjust the best solution and the KNN classifier is employed.

Zhang, Mistry, Lim, and Neoh (2018) incorporated the firefly algorithm with the simulated annealing to enhance the quality of solutions and escape from the local optima. The return-cost at-

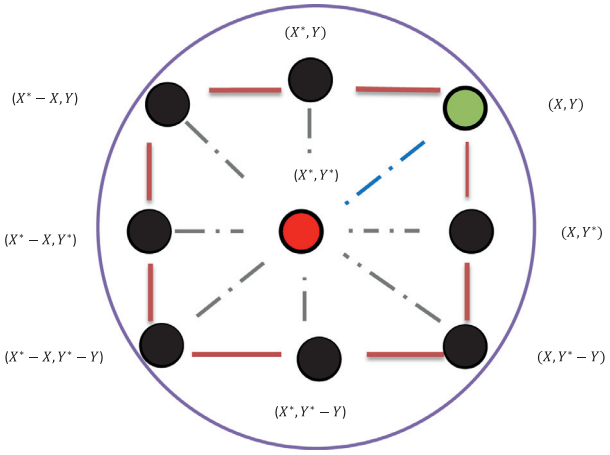

Fig. 1. 2D position vector with possible next locations.

tractiveness, the binary movement with the adaptive jump and the Pareto dominance-based selection are used by Zhang, Song, and Gong (2017) to effectively solve the feature selection problem. Faris, Hassonah, Ala'M, Mirjalili, and Aljarah (2018b) proposed a multi-verse optimizer algorithm based on Support vector machine (SVM) classifier. A new variant of particle swarm optimization algorithm (Gu, Cheng, & Jin, 2018) is suggested for tackling high dimensional feature selection. A binary version of brain storm optimization algorithm is introduced for solving the feature selection problems (Papa, Rosa, de Souza, & Afonso, 2018).

To increase the effectiveness of the grey wolf algorithm, three different strategies are incorporated (Tu, Chen, & Liu, 2019). Also, the authors (Tu et al., 2019) suggested dividing the grey wolves into two groups dominant and omega. Two different strategies are applied to each group. But on the other side, for the two the proposed algorithms, training the parameters is required in advance to get the optimal setting. Moreover, the grey wolf algorithm is used with different classifiers: decision-tree, KNN, convolutional neural network classifier and has achieved high accuracy for detecting Alzheimer as in Shankar et al. (2019).

Other metaheuristic algorithms are proposed for feature selection as discussed by Hafez, Zawbaa, Emary, and Hassanien (2016); Jain and Purohit (2017); Kashef and Nezamabadi-pour (2015); Li, Dong, and Sun (2019); Tabakhi, Moradi, and Akhlaghian (2014); Wang, Jing, and Niu (2017); Xue, Zhang, and Browne (2014). The hybridization of the different algorithms was a trend for feature selection to benefit from the advantages and the strength of different algorithms. Many papers have discussed this as in Al-Tashi, Kadir, Rais, Mirjalili, and Alhussian (2019); Arora, Singh, Sharma, Sharma, and Anand (2019); Cho, Lee, Park, and Chun (2013); Ghamisi and Benediktsson (2015); Hafez, Hassanien, Zawbaa, and Emary (2015); Tawhid and Dsouza (2018b); Tawhid and Dsouza (2018a).

3. Grey wolf algorithm

Grey wolves are one of the most fantastic predators that have high skills in catching their prey. This is because they live in a strict and organized pack. A new metaheuristic has recently been proposed to simulate the behaviors of the grey wolves in hunting, searching, and encircling their prey, known as Grey Wolf Optimizer (GWO) algorithm (Mirjalili, Mirjalili, & Lewis, 2014a). Given the social hierarchy of the wolves' community, there are four types of wolves with different level of dominance and leadership (Alpha (α), Beta (β), Delta (δ), and Omega (ω)). In the GWO algorithm, the alpha, beta, and delta represent the first, second, third best solutions respectively. The remaining candidate solutions are regarded

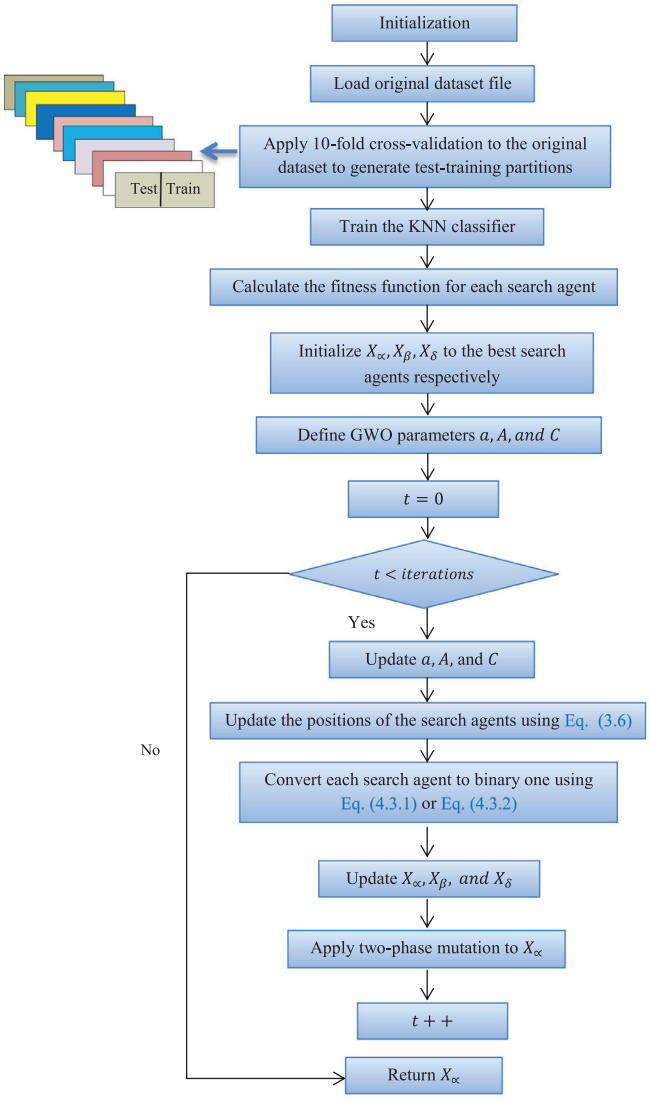

Fig. 2. Flowchart of the proposed TMGWO algorithm.

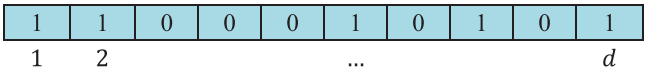

Fig. 3. Solution representation.

as omega. The wolves can encircle their prey during the hunting process in a manner that can be mathematically modeled as:

$$\vec{D} = \left| \vec{C} \cdot \vec{X_p} - \vec{X}(t) \right| \tag{3.1}$$

$$\vec{X}(t+1) = |\vec{X_p}(t) - \vec{A} \cdot \vec{D})| \tag{3.2}$$

where $\vec{X_p}$ and $\vec{X}$ represent the positions of the prey and grey wolf respectively at an iteration (t). $\vec{A}$ and $\vec{C}$ are coefficient vectors that can be formulated as follows:

$$\vec{A} = \left| 2\vec{a} \cdot \overrightarrow{rand1} - \vec{a} \right| \tag{3.3}$$

$$\vec{C} = 2 \cdot \overrightarrow{rand2} \tag{3.4}$$

where $\overrightarrow{rand1}$ and $\overrightarrow{rand2}$ are random vectors in [0, 1]. $\vec{a}$ is linearly decreasing from 2 to zero over iterations as follows:

$$a = 2 - t * \frac{2}{iterations} \tag{3.5}$$

The parameter *iterations* determines the maximum number of iterations. The position of the grey wolf (X, Y) can be updated according to the position of the prey (X^*, Y^*). The position of the best grey wolf can be updated by adjusting the vectors $\vec{A}$ and $\vec{C}$. Fig. 1 shows a 2D position vector and the next possible positions of the grey wolf.

In order to mimic the hunting behavior, the alpha, beta, and delta are aware of the potential locations of the prey. As alpha, beta, and delta indicate the best three solutions, the rest of other wolves update their positions according to the best three solutions $(\vec{X_1}, \vec{X_2}, and\ \vec{X_3})$. This can be expressed as follows:

$$\vec{X}(t+1) = \left(\vec{X_1} + \vec{X_2} + \vec{X_3}\right)/3 \tag{3.6}$$

$$\vec{X_1} = \vec{X_\alpha} - \vec{A_1} \cdot \left(\vec{D_\alpha}\right), \vec{D_\alpha} = \left|\vec{C} \cdot \vec{X_\alpha} - \vec{X}\right| \tag{3.7}$$

$$\vec{X_2} = \vec{X_\beta} - \vec{A_2} \cdot \left(\vec{D_\beta}\right), \vec{D_\beta} = \left|\vec{C_2} \cdot \vec{X_\beta} - \vec{X}\right| \tag{3.8}$$

$$\vec{X_3} = \vec{X_\delta} - \vec{A_3} \cdot \left(\vec{D_\delta}\right), \vec{D_\delta} = \left|\vec{C_3} \cdot \vec{X_\delta} - \vec{X}\right| \tag{3.9}$$

When attacking the prey, each wolf updates its position between its current position and the position of the prey so that $|A| < 1$. In searching for the prey, the alpha, beta, delta wolves go away from each other to search for the prey and converge when attacking the prey. $\vec{A_1}$ can take random values less than -1 or greater than 1 to force the wolves to diverge from the prey. Last but not least, the pseudocode for GWO is present in Algorithm 1.

Algorithm 1 The standard GWO.

```
Initialize a population of n grey wolves X_i, i = 1, ..., n
Initialize the parameters a, A, and C
Calculate the fitness of each grey wolf
Assign the best three grey wolves to X_α, X_β, and X_δ respectively
Define t = 0
while (t < iterations)
    for each grey wolf
        Update the position of the current grey wolf using Eq. (3.6)
    end for
    Update a, A, and C
    Calculate the fitness of all the grey wolves
    Update the first three grey wolves X_α, X_β, and X_δ
    t++
end while
return the best grey wolf X_α
```

Table 1
Description of the datasets.

No.	Dataset	#F	#samples	#classes	Area
1	Australian	14	690	2	Financial
2	Breast cancer Coimbra	9	116	2	Medical
3	Breast cancer tissue	9	106	6	Medical
4	Climate	20	540	2	Physical
5	Diabetic	19	1151	2	Life
6	Fri_c0_500_10	10	500	2	N/A
7	Fri_c0_1000_10	10	1000	2	N/A
8	Fri_c1_1000_10	10	1000	2	N/A
9	German	24	1000	2	Financial
10	Glass	9	214	7	Physical
11	HeartEW	13	270	5	Life
12	IonosphereEW	34	351	2	Physical
13	Kc1	21	2109	2	N/A
14	Kc2	21	522	2	N/A
15	Lung cancer	21	226	2	Medical
16	Lymphography	18	148	4	Medical
17	Meta all	62	71	6	Life
18	Page blocks	10	5473	2	Computer
19	Parkinsons	22	195	2	Life
20	Pc1	21	1109	2	N/A
21	Robot-failures-lp1	90	88	4	Physical
22	Robot-failures-lp2	90	47	5	Physical
23	Robot-failures-lp3	90	47	4	Physical
24	Robot-failures-lp4	90	117	3	Physical
25	Robot-failures-lp5	90	164	5	Physical
26	Segment	19	2310	7	Life
27	SonarEW	60	208	2	Physical
28	Space-ga	6	3107	2	Space
29	SpectEW	22	267	2	Medical
30	Stock	9	950	2	Business
31	Vehicle	18	846	4	Life
32	Vowel	10	990	2	Life
33	WineEW	13	178	3	Physical
34	WDBC	30	569	2	Life
35	Zoo	16	101	7	Life

Table 2
Parameter setting for the proposed algorithm.

Parameter	Value
Number of independent runs	20
Number of iterations	30
Number of search agents n	5
Dimension d	Number of features
K-neighbors	5
K-fold cross-validation	10
$\propto$	0.01
β	0.99
A	[0,2]
M_p	0.5

4. The proposed algorithm

In this section, the proposed algorithm TMGWO is explored for tackling the feature selection problems (see Fig. 2). The main steps of TMGWO are initialization, evaluation, transformation function, and two-phase mutation. For each of these steps, a detailed explanation will be presented in the next subsections.

4.1. Initialization

The first step is the initialization phase, in which we initialize a population of randomly generated n wolves or search agents. Each search agent can represent a possible solution and has a dimension d equal to the number of features in the original dataset. Fig. 3 shows a possible solution for a dataset that contains ten features. The problem of feature selection for the classification purpose is summarized by selecting a few of the possible features that maximize the classification accuracy. Therefore, we want to specify some features (one value) and reject other features (zero value). Initially, each solution is configured with binary values (0′s or 1′s).

4.2. Evaluation

When there is more than one objective that the problem must satisfy to get the best solution, this problem is known as a multi-objective problem. Based on this, the problem of feature selection is considered to be a multi-objective problem as it must achieve the following:

- Minimize the number of selected features.
- Maximize the classification accuracy of a given classifier.

Based on the above, the fitness function for evaluating the solutions is designed to achieve the balance between the two objectives as:

$$fitness = \propto \gamma_R(D) + \beta \frac{|s|}{|d|} \tag{4.2.1}$$

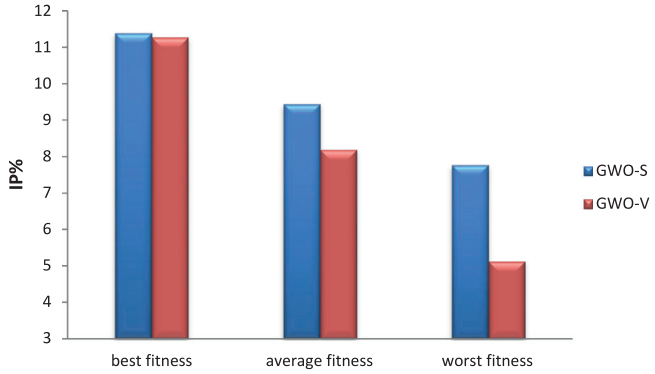

Fig. 4. Comparison based on total improvement percentage for the fitness values.

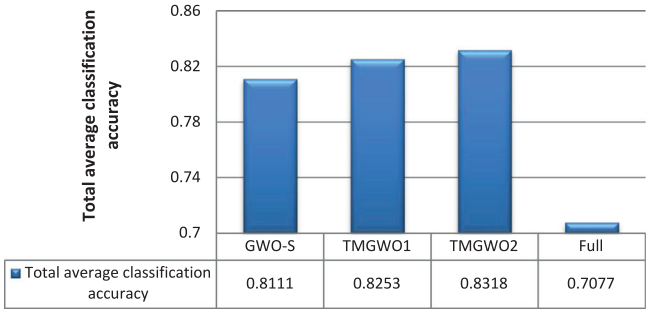

Fig. 5. Total average classification accuracy of the algorithms averaged over all the datasets.

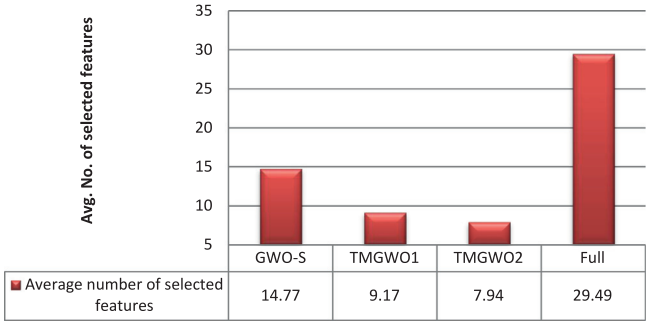

Fig. 6. Average number of selected features for the best classification accuracy over all the datasets.

where $\gamma_R(D)$ is the classification error rate of condition attribute set R relative to decision D, calculated using the classifier KNN (Altman, 1992). $|s|$ represents the cardinality of the length of the selected feature subset. $|d|$ is the cardinality of all the features found in each sample in the original dataset. α and β are the weight parameters that correspond to the importance of the classification accuracy and the selected feature subset length, $\alpha \in [0,1]$ and $\beta = 1 - \alpha$. The significant impact and weight is given to the classification accuracy rather than the number of selected features. If the evaluation function only takes the classification accuracy into consideration, the result will be the neglect of the solutions that may have the same accuracy but have fewer chosen features that act as a main factor in decreasing the dimensionality issue. In KNN, each sample can be classified into a specific class label to which the majority its K neighbors belongs to. KNN classifier is characterized by its simple implementation. The quintessence of classification is to classify new incoming samples for which the class label is not known. A common practice is to divide the dataset into training and testing data. Each sample in the testing data needs to determine its K nearest neighbors from the training data. Accordingly, in order to determine the nearest K neighbors to this sample, the Euclidean distance is therefore calculated as:

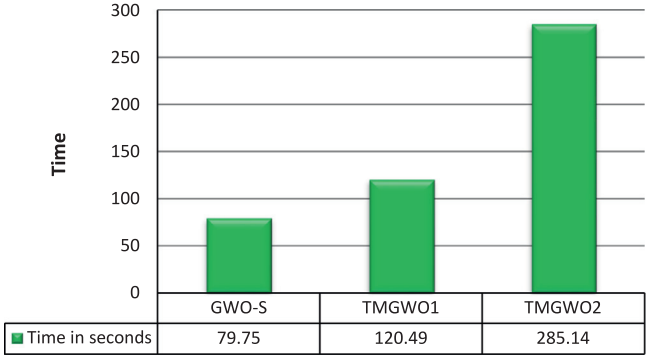

Fig. 7. Average time in seconds for all datasets.

$$Euc_D = \sqrt{\sum_{h=1}^{d}(train\_f_h - test\_f_h)} \tag{4.2.2}$$

where $train\_f_h$ and $test_f_h$are a single feature in a specific sample of the training and testing data respectively. h is a variable and $h = 1, \ldots, d$ where d is the number of features in each sample. To train the classifier, a common practice is to save part of the data to be test data and the remaining data is used to train the classifier. But when we are doing that, we may face the overfitting problem. This problem happens when the classifier does much better accuracy on the training data than on the test data. One way to lessen the overfitting problem is to use the cross-validation. In this paper, K-fold cross-validation is used with $K = 10$. The idea behind the cross-validation is dividing the dataset into K folds or partitions of roughly equal size. The classifier will be trained on $K-1$ partitions combined together and then applied to the remaining partition as testing data to predict the class label of each sample. After that, the percentage of the incorrect prediction of the class labels which known as the classification percentage error rate is calculated. The results from different rounds of data are averaged to have reliable statistical results.

4.3. Transformation function

The positions of the search agents generated from the standard GWO are continuous values. So, it cannot be directly applied to our problem due to its contradiction with the binary nature of the feature selection. Since the problem of feature selection comes from selecting or not selecting (0 or 1) the most beneficial features that maximize the classification accuracy. A transformation function is responsible for mapping the continuous search space of the standard GWO to a binary one. The performance of two different transformation functions is deeply investigated in the results section. The different two transformation functions are used here. The first one is the sigmoidal function which is considered as an example of the S-shaped function (Mirjalili & Lewis, 2013). The second one is *tanh* function which is considered an example of the V-Shaped function (Mafarja et al., 2017). The sigmoidal and *tanh* functions can convert any continuous value to be binary one by using:

$$x_{s_i} = \frac{1}{1+e^{-x_i}}, \; x_{binary} = \begin{cases} 0 & if \; rand < x_{s_i} \\ 1 & if \; rand \geq x_{s_i} \end{cases} \tag{4.3.1}$$

$$x_{v_i} = |tanh(x)|, \; x_{binary} = \begin{cases} 0 & if \; rand < x_{v_i} \\ 1 & if \; rand \geq x_{v_i} \end{cases} \tag{4.3.2}$$

where each of $x_{s_i}$and x_{v_i} is a continuous value (feature) in the search agent for the S-shaped and V-shaped functions respectively

Algorithm 2 The proposed two-phase mutation.

Input: the best grey wolf X_α from each iteration
$fitness$=Calculate the fitness of X_α
// start the first phase
Define vector $one\_positions$ to store the locations of the selected features in X_α
Define $X_{mutated1} = X_\alpha$
for $i = 1$ to length of $one\_positions$ //for each selected feature in X_α
  Generate a random number r
  if $(r < M_p)$
    $X_{mutated1}[one\_positions[i]] = 0$ while keeping the other features
    $fitness\_mutated =$ the fitness of $X_{mutated1}$
    if $(fitness\_mutated < fitness)$
      $fitness = fitness\_mutated$
      $X_\alpha = X_{mutated1}$
    end if
  end if
end for
// start the second phase
Define vector $zero\_positions$ to store the locations of the unselected features in X_α
Define Define $X_{mutated2}$
for $j = 1$ to length of $zero\_positions$ // for each unselected feature in X_α
  Generate a random number r
  if $(r < M_p)$
    $X_{mutated2}[zero\_positions[j]] = 1$ while keeping the other features
    $fitness\_mutated =$ the fitness of $X_{mutated2}$
    if $(fitness\_mutated < fitness)$
      $fitness = fitness\_mutated$
      $X_\alpha = X_{mutated2}$
    end if
  end if
end for
output: the improved X_α

Algorithm 3 The proposed TMGWO.

Initialize a population of n grey wolves $X_i, i = 1, \ldots . n$
Each grey wolf is a vector has a d dimension equal to the problem size
Initialize the parameters a, A, and C
Calculate the fitness of each grey wolf
Assign the best three grey wolves to X_α, $X_\beta, and\ X_\delta$ respectively
Define $t = 0$
while $(t < iterations)$
  for each grey wolf
    Update the position of the current grey wolf using Eq. (3.6)
  end for
  Update a, A, and C
  Convert the position of grey wolves into binary ones using Eq. (4.3.1) or Eq. (4.3.2)
  Calculate the fitness of all the grey wolves in the population
  Update the first three grey wolves X_α, $X_\beta, and\ X_\delta$
  $t++$
  $X_\alpha =$ apply two-phase mutation to (X_α)
  Update the fitness of X_α
end while
return the best grey wolf X_α

and $i = 1, \ldots, d$. x_{binary} value can be 0 or 1 according to the value of a random number *rand* compared to the values of x_{s_i} and x_{v_i}. The two functions have the ability to convert the continuous search space into a binary one. The effect of using each of them with the proposed algorithm TMGWO will be investigated later.

4.4. Two-phase mutation

To enhance the exploitation of the proposed algorithm, a bi-phase mutation operator is used. The purpose of the first mutation phase is to lessen the number of selected features while preserving the high classification accuracy. The second phase is interested in adding more informative features that increase the classification accuracy. The two-phase mutation can be done with a probability M_p to overcome the time-consuming problem that accompanies the mutation operator. The pseudocode of the proposed mutation can be described in Algorithm 2. Finally, the overall pseudocode of TMGWO can be found in Algorithm 3.

5. Experimental results and discussion

We have conducted all the experiments in Windows 10 Ultimate 64-bit operating system; processor Intel Core i5-3317 U CPU, processor speed of 1.70 GHz; 4 GB of RAM. We have implemented all the algorithms in the Java programming language

Table 3
Comparison between GWO-S and GWO-V based on the fitness value.

No.	Dataset	Best fitness		Mean fitness		Worst fitness		Full
		GWO-S	GWO-V	GWO-S	GWO-V	GWO-S	GWO-V	
1	Australian	**0.1664**	0.1807	**0.1938**	0.2009	**0.2374**	0.2123	0.3213
2	Breast cancer Coimbra	**0.2654**	0.2744	**0.2788**	0.3438	**0.2924**	0.4522	0.6220
3	Breast cancer tissue	**0.6677**	**0.6677**	**0.6956**	0.7148	**0.7194**	0.7678	0.8020
4	Climate	0.0828	**0.0820**	**0.0865**	0.0897	**0.0878**	0.0938	0.1199
5	Diabetic	0.3017	**0.2917**	**0.3163**	0.3188	**0.3311**	0.3375	0.3440
6	Fri_c0_500_10	**0.1376**	**0.1376**	0.1564	0.1784	0.1713	0.1931	0.2080
7	Fri_c0_1000_10	**0.1436**	**0.1436**	**0.1561**	0.1651	**0.1792**	0.1970	0.2208
8	Fri_c1_1000_10	**0.1059**	0.1267	**0.1244**	0.1424	**0.1347**	0.1613	0.2327
9	German	**0.2643**	0.2722	**0.2715**	0.2809	**0.2769**	0.2937	0.3208
10	Glass	**0.2637**	0.2648	**0.2664**	0.2730	**0.2742**	0.2800	0.3117
11	HeartEW	0.1894	**0.1777**	**0.2083**	0.2255	**0.2340**	0.3029	0.3436
12	IonosphereEW	0.1226	**0.0979**	0.1358	**0.1311**	**0.1452**	0.1546	0.1797
13	Kc1	0.3093	**0.3069**	**0.3115**	0.3148	**0.3149**	0.3253	0.3329
14	Kc2	0.2337	**0.2118**	0.2401	**0.2394**	**0.2455**	0.2617	0.2746
15	Lung cancer	**0.1177**	0.1184	**0.1343**	0.1420	**0.1411**	0.1574	0.1765
16	Lymphography	**0.1421**	0.1480	**0.1558**	0.1742	**0.1823**	0.2029	0.2362
17	Meta all	**0.3150**	0.3432	**0.3430**	0.3690	**0.3731**	0.4013	0.4767
18	Page blocks	0.0466	**0.0402**	**0.0484**	0.0487	**0.0500**	0.0536	0.0563
19	Parkinsons	**0.1452**	0.1316	**0.1668**	0.1855	**0.1930**	0.2240	0.2653
20	Pc1	**0.0887**	0.0910	**0.0908**	0.0930	**0.0933**	0.0955	0.1071
21	Robot-failures-lp1	**0.1298**	0.1393	**0.1521**	0.1528	**0.1647**	0.1758	0.2451
22	Robot-failures-lp2	0.2513	**0.2480**	**0.2750**	0.2882	**0.2789**	0.3267	0.3317
23	Robot-failures-lp3	0.3006	**0.2996**	**0.3095**	0.3232	**0.3265**	0.3753	0.4307
24	Robot-failures-lp4	0.1211	**0.1199**	**0.1448**	0.1552	**0.1590**	0.1846	0.1989
25	Robot-failures-lp5	**0.3703**	0.3762	**0.3889**	0.3891	**0.4015**	**0.4015**	0.4121
26	Segment	**0.0301**	0.0351	**0.0395**	0.0415	**0.0432**	**0.0432**	0.0532
27	SonarEW	0.3962	**0.3379**	0.4173	**0.3890**	**0.4351**	0.4565	0.5000
28	Space-ga	**0.2107**	**0.2107**	**0.2109**	0.2271	**0.2132**	0.2617	0.5497
29	SpectEW	**0.2623**	0.2681	0.2836	**0.2810**	**0.2948**	0.2980	0.3260
30	Stock	**0.0711**	**0.0711**	**0.0796**	0.0870	**0.0879**	0.0890	0.1590
31	Vehicle	0.2794	**0.2778**	**0.2858**	0.3029	**0.2942**	0.3485	0.3494
32	Vowel	0.0189	**0.0169**	**0.0213**	0.0221	**0.0240**	0.0259	0.0369
33	WineEW	**0.0717**	0.0725	**0.0812**	0.1130	**0.0985**	0.1517	0.3419
34	WDBC	**0.0562**	0.0577	**0.0623**	0.0683	**0.0682**	0.0816	0.0842
35	Zoo	**0.0452**	0.0458	**0.0490**	0.0541	**0.0557**	0.0625	0.0693

Bold values indicate the best results.

5.1. Dataset description

The effectiveness and the strength of our proposed algorithm will be thoroughly investigated by using well-known datasets in feature selection. There are 35 datasets which are taken from the UCI machine learning repository (Asuncion & Newman, 2007) and online at https://www.openml.org /search. A brief description of the used datasets is given in Table 1. For each dataset, the number of samples (#samples), features (#F), classes (#classes) and the area to which each dataset belongs are provided.

5.2. Parameter setting

The performance of the proposed algorithm is compared with other well-regarded state-of-the-art feature selection algorithms summarized as:

- Binary Bat Algorithm (BA) (Mirjalili, Mirjalili, & Yang, 2014b).
- Binary Crow Search Algorithm (BCSA) (De Souza et al., 2018).
- Binary Grey Wolf Optimization Algorithm (bGWOA) (Emary et al., 2016b).
- Binary whale Optimization Algorithm (bWOA) (Hussien, Hassanien, Houssein, Bhattacharyya, & Amin, 2019).
- Discrete particle swarm optimization (DPSO) algorithm (Unler and Murat (2010)
- Flower Algorithm (FA) (Yang, 2012).
- Multi-Verse Optimizer (MVO) algorithm (Mirjalili, Mirjalili, & Hatamlou, 2016).
- Non-Linear Particle Swarm Optimization algorithm (NLPSO) (Mafarja, Jarrar, Ahmad, & Abusnaina 2018).
- Particle Swarm Optimization (PSO) algorithm (Kennedy & Eberhart, 1997).

Each algorithm is run 20 independent runs with a random seed. For all the next experiments, the maximum number of iterations is set to 30. The number of search agents in the population is 5. Furthermore, 10-fold cross-validation is adopted here. Several experiments are conducted on different datasets to select the best values for $\propto$ and β based on some values taken from the literature. As a result, the values of $\propto$ and β are set to 0.01 and 0.99 respectively. The parameters of the proposed algorithm are presented in Table 2. The parameters of the algorithms are taken from the literature to ensure a fair comparison among the algorithms. Our main objective is to investigate the performance of the different algorithms for solving the feature selection problems compared to the proposed algorithm. A common wrapper method for feature selection is used here is KNN classifier. KNN is one of the most common supervised learning algorithms that classifies the new instance in the test set based on distance from the new instance to the training set. It is characterized by its simple and easy implementation. In addition, we only tune one parameter K in comparison with other classifiers. Several trials and runs are done on different datasets to determine the best choice for K value. The algorithm gives better results when K=5. 10-fold cross-validation is recommended for real-world datasets as in Friedman, Hastie, and Tibshirani (2001); Kohavi, (1995). Also, we see that the variance of the performance accuracy between the folds is minimized. 10-fold cross-validation is used to split the dataset into 10 folds distributed between training and testing data with ratio 9:1. The training data is used to learn the KNN classifier while the test data is kept far away.

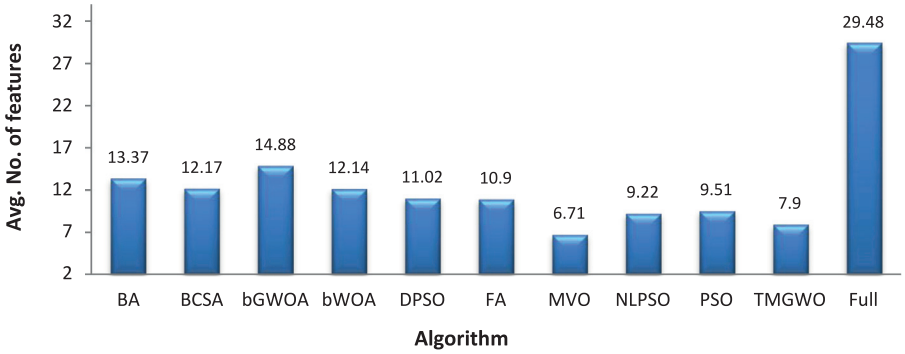

Fig. 8. The average number of features selected by the algorithms.

Table 4
Comparison based on the classification accuracy and number of selected features.

No.	Dataset	GWO-S		TMGWO1		TMGWO2		Full	
		Accuracy	#F	Accuracy	#F	Accuracy	#F	Accuracy	#F
1	Australian	**0.8362**	**6**	0.8318	6	**0.8463**	7	0.6855	14
2	Breast cancer Coimbra	**0.7363**	**4**	0.7363	4	**0.7363**	**4**	0.3818	9
3	Breast cancer tissue	**0.3300**	**4**	0.3300	4	**0.3300**	**4**	0.2000	9
4	Climate	0.9203	8	0.9277	8	**0.9314**	**6**	0.8888	20
5	Diabetic retinopathy	0.7000	9	**0.7121**	**8**	**0.7121**	**8**	0.3440	19
6	Fri_c0_500_10	**0.8660**	**5**	**0.8660**	**5**	**0.8660**	**5**	0.7999	10
7	Fri_c0_1000_10	0.8610	6	**0.8750**	**5**	**0.8750**	**5**	0.7870	10
8	Fri_c1_1000_10	0.8979	5	**0.9010**	**4**	**0.9010**	**4**	0.7750	10
9	German	0.7410	19	**0.7560**	**14**	**0.7560**	**14**	0.6859	24
10	Glass	**0.7380**	**4**	**0.7380**	**4**	**0.7380**	**4**	0.6952	9
11	HeartEW	0.8148	8	**0.8407**	**6**	**0.8407**	**6**	0.6629	13
12	IonosphereEW	0.8799	13	0.9257	5	**0.9314**	**4**	0.8285	34
13	Kc1	.0.6938	13	0.6990	9	**0.7042**	**8**	0.6938	21
14	Kc2	0.7711	15	**0.7961**	**5**	**0.7961**	**5**	0.7326	21
15	Lung cancer	0.8863	11	0.8863	6	**0.9000**	**6**	0.8318	21
16	Lymphography	0.8642	14	0.8928	11	**0.9000**	**12**	0.7714	18
17	Meta all	0.6857	24	0.7000	14	**0.7428**	**11**	0.5285	62
18	Parkinsons	0.8578	10	0.8631	6	**0.8684**	**6**	0.7421	22
19	Page blocks	0.9568	5	**0.9639**	**4**	**0.9639**	**4**	0.9531	10
20	Pc1	0.9145	9	0.9163	6	**0.9172**	9	0.9018	21
21	Robot-failures-lp1	0.8750	55	0.9125	11	**0.9125**	**6**	0.7625	90
22	Robot-failures-lp2	0.7500	35	0.7500	21	**0.7750**	**6**	0.6750	90
23	Robot-failures-lp3	0.7000	33	0.7500	22	**0.7750**	**18**	0.5750	90
24	Robot-failures-lp4	0.8818	37	0.9181	28	**0.9363**	**20**	0.8090	90
25	Robot-failures-lp5	0.6312	48	0.6687	34	**0.6750**	**31**	0.5937	90
26	Segment	0.9688	6	**0.9744**	**6**	**0.9744**	**6**	0.9562	19
27	SonarEW	0.6050	31	0.7049	14	**0.7400**	**6**	0.5050	60
28	Space-ga	**0.7938**	**4**	**0.7938**	**4**	**0.7938**	**4**	0.5497	6
29	SpectEW	0.7423	16	0.7461	9	**0.7615**	**12**	0.6807	22
30	Stock	**0.9326**	**4**	0.9326	4	**0.9326**	**4**	0.8494	9
31	Vehicle	0.7250	13	0.7321	8	**0.7380**	9	0.6571	18
32	Vowel	0.9878	7	0.9888	6	**0.9888**	**6**	0.7353	12
33	WineEW	0.9352	11	0.9470	6	**0.9470**	**6**	0.6647	13
34	WDBC	0.9482	15	0.9482	5	**0.9482**	**4**	0.9250	30
35	Zoo	0.9600	10	0.9600	9	**0.9600**	**8**	0.9400	16

Bold values indicate the best results.

Table 5
Paired samples statistics.

	Mean	N	Std	Std. Error Mean
TMGWO	0.8318	35	0.1270821	0.0214808
GWO-S	0.8111	35	0.1332094	0.0225165

5.3. Comparison based on transformation function

In the first experiment, a binary version of GWO is implemented using the sigmoid function (GWO-S) and the *tanh* function (GWO-V) to the study the effect of using different transformation functions. Additionally, three performance measures are used to compare the performance of GWO-S and GWO-V based on the fitness function value depicted as:

- The best fitness value is the minimum fitness value obtained by running each of the two algorithms 20 times as in Eq. (5.3.1). f_i is the final fitness value obtained at the end of each run.

$$Best\ \ fitness = \min_{i=1}^{20} f_i \tag{5.3.1}$$

- The mean fitness value is the average of the fitness values obtained from running the algorithm 20 times.

$$Mean\ \ fitness = \frac{1}{20}\sum_{i=1}^{20} f_i \tag{5.3.2}$$

Table 6
Paired samples test.

	Paired Differences					t	Df	Sig. (2-tailed)
	Mean	Std	Std. Error Mean	95% confidence interval of the difference				
				Lower	Upper			
(TMGWO) – (GWO-S)	0.0207	0.0277	0.0047	0.01123	0.03028	4.428	34	0.000

Table 7
Comparison among algorithms based on the classification accuracy.

No.	Dataset	BA	BCSA	bGWOA	bWOA	DPSO	FA	MVO	PSO	NLPSO	TMGWO	Full
1	Australian	0.8312	0.8304	0.8260	0.8391	0.8318	0.8376	0.8318	0.8333	0.8304	**0.8463**	0.6855
2	Breast cancer Coimbra	**0.7363**	**0.7363**	**0.7363**	**0.7272**	**0.7363**	**0.7363**	**0.7363**	**0.7363**	**0.7363**	**0.7363**	0.3818
3	Breast cancer tissue	**0.3300**	**0.3300**	**0.3300**	**0.3200**	**0.3300**	**0.3300**	**0.3300**	0.3200	**0.3300**	**0.3300**	0.2000
4	Climate	0.9184	**0.9416**	0.9240	0.9222	0.9185	0.9277	0.9277	0.9203	**0.9416**	0.9314	0.8888
5	Diabetic retinopathy	0.9242	0.6973	0.9250	0.7043	0.6932	0.7086	0.7113	0.6947	0.6973	**0.7121**	0.3440
6	Fri_c0_500_10	0.8508	**0.8660**	0.8360	**0.8660**	0.8540	**0.8660**	**0.8660**	**0.8660**	**0.8660**	**0.8660**	0.7999
7	Fri_c0_1000_10	**0.8750**	**0.8750**	**0.8610**	**0.8750**	**0.8750**	**0.8750**	**0.8750**	**0.8750**	**0.8750**	**0.8750**	0.7870
8	Fri_c1_1000_10	**0.9010**	**0.9010**	**0.9010**	**0.8979**	**0.9010**	**0.9010**	**0.9010**	0.8979	**0.9010**	**0.9010**	0.7750
9	German	0.7420	0.7460	0.7440	0.7400	0.7430	0.7460	0.7540	0.7420	0.7460	**0.7560**	0.6859
10	Glass	0.7342	**0.7380**	**0.7380**	**0.7380**	**0.7380**	**0.7380**	**0.7380**	**0.7380**	**0.7380**	**0.7380**	0.6952
11	HeartEW	0.8259	0.8259	**0.8407**	**0.8148**	**0.8407**	**0.8407**	0.8296	0.8222	0.8259	**0.8407**	0.6629
12	IonosphereEW	0.8765	0.9085	0.8828	0.8942	0.9000	0.9057	0.9057	0.9142	0.9085	**0.9314**	0.8285
13	Kc1	0.7004	0.6961	0.7028	0.6976	0.6938	0.6928	0.7033	0.6942	0.6961	**0.7042**	0.6938
14	Kc2	0.7846	0.8076	0.7730	0.7750	0.8038	0.7884	0.7923	**0.7961**	**0.8076**	**0.7961**	0.7326
15	Lung cancer	0.8772	0.8727	0.8863	0.8772	0.8777	0.8772	**0.9000**	0.8772	0.8727	**0.9000**	0.8318
16	Lymphography	0.8775	0.8785	0.8642	0.8714	0.8714	0.8571	0.8857	0.8785	0.8785	**0.9000**	0.7714
17	Meta all	0.6857	0.6857	0.7000	0.6857	0.7285	0.6857	0.7000	0.6571	0.6857	**0.7428**	0.5285
18	Parkinsons	0.8578	**0.8684**	0.8578	0.8578	**0.8684**	0.8578	0.8631	**0.8684**	**0.8684**	**0.8684**	0.7421
19	Page blocks	0.9163	0.9559	0.9639	**0.9643**	0.9612	0.9612	0.9610	**0.9643**	**0.9570**	**0.9639**	0.9531
20	Pc1	0.9163	0.9163	0.9145	0.9172	0.9154	0.9154	**0.9172**	0.9145	0.9163	**0.9172**	0.9018
21	Robot-failures-lp1	0.8924	0.8875	0.8750	0.8750	0.8625	0.8750	**0.9125**	0.8875	0.8875	**0.9125**	0.7625
22	Robot-failures-lp2	0.7500	**0.7750**	0.7250	0.7250	0.7500	0.7500	**0.7750**	0.7500	**0.7750**	**0.7750**	0.6750
23	Robot-failures-lp3	0.72.5	0.7500	0.7250	0.7250	0.7500	0.7000	0.7500	0.7000	0.7500	**0.7750**	0.5750
24	Robot-failures-lp4	0.8750	0.8818	0.8727	0.8818	0.8727	0.8909	0.9272	0.9000	0.8818	**0.9363**	0.8090
25	Robot-failures-lp5	0.6374	0.6250	0.6250	0.6312	0.6312	0.6437	**0.6875**	0.6375	0.6250	0.6750	0.5937
26	Segment	0.9714	0.9679	0.9705	0.9679	0.9683	0.9709	**0.9744**	0.9658	0.9679	**0.9744**	0.9562
27	SonarEW	0.6550	0.6799	0.6200	0.6600	0.6750	0.6600	0.7300	0.6800	0.6799	**0.7400**	0.5050
28	Space-ga	**0.7938**	**0.7938**	**0.7938**	**0.7938**	**0.7938**	**0.7938**	**0.7938**	**0.7938**	**0.7938**	**0.7938**	0.5497
29	SpectEW	0.7423	0.7346	0.7500	0.7230	0.7307	0.7346	0.7576	0.7346	0.7346	**0.7615**	0.6807
30	Stock	**0.9326**	**0.9326**	**0.9326**	**0.9326**	**0.9326**	**0.9326**	**0.9326**	**0.9326**	**0.9326**	**0.9326**	0.8494
31	Vehicle	0.7261	0.7321	0.7297	0.7309	0.7250	0.7273	0.7345	0.7261	0.7321	**0.7380**	0.6571
32	Vowel	**0.9888**	**0.9534**	**0.8888**	**0.9672**	**0.8981**	**0.9888**	**0.9888**	**0.9888**	**0.9756**	**0.9888**	0.7353
33	WineEW	0.9411	0.9352	0.9411	0.9411	0.9411	0.9411	**0.9470**	0.9352	0.9352	**0.9470**	0.6647
34	WDBC	**0.9482**	**0.9464**	**0.9482**	**0.9482**	**0.9482**	**0.9482**	**0.9482**	**0.9482**	**0.9464**	**0.9482**	0.9250
35	Zoo	**0.9600**	**0.9600**	**0.9600**	**0.9600**	**0.9600**	**0.9600**	**0.9600**	**0.9600**	**0.9600**	**0.9600**	0.9400

Bold values indicate the best results.

- The worst fitness value is the maximum fitness value obtained by running the algorithm twenty times.

$$Worst\ \ fitness = \max_{i=1}^{20} f_i \tag{5.3.3}$$

- full is the fitness value of selecting all the features in each dataset.

The results of comparing the two algorithms (GWO-S, GWO-V) are presented in Table 3. The best results are indicated by bold values in the table. As can be seen, GWO-S performs better than GWO-V in most of the datasets. As well as the performance of GWO-V algorithm has been degraded especially when comparing with the worst fitness values. Fig. 4 depicts the total Improvement Percentage (IP) on the performance of the two algorithms compared to the fitness value for all the datasets. Total IP can be formulated as:

$$IP = \sum_{i=1}^{m} \frac{full\ - f_{Algo.}}{full} \times 100 \tag{5.3.4}$$

where $f_{Algo.}$ is the fitness value of GWO-S and GWO-V for the best, mean, and worst case. i is a variable where $i = 1, \ldots, m$ and m is the number of datasets ($m = 35$). The figure shows that the total improvement percentage has significantly increased and the performance has been improved when compared with selecting all the features for each dataset. In addition, we have observed that GWO-S gives much better IP than GWO-V overall the datasets.

5.4. The effect of the two-phase mutation operator on the proposed algorithm

As illustrated above, the performance of two different transformation functions is examined to find the most effective one. The results proved that GWO-S is more proficient than GWO-V. In the second experiment, it is the time to test the effect of adding the two-phase mutation operator to the GWO-S (TMGWO). In this subsection, the mutation probability has been changed as well to test its impact on the algorithm's results. In Table 4, a comparison among the following algorithms is done:

- Grey wolf optimization algorithm with the sigmoid function (GWO-S).

Table 8
Comparison among algorithms based on the average fitness values.

No.	Dataset	BA	BCSA	bGWOA	bWOA	DPSO	FA	MVO	NLPSO	PSO	TMGWO	Full
1	Australian	0.1781	0.2222	0.2007	0.1882	0.1813	0.1819	0.1759	0.1813	0.1900	**0.1686**	0.3213
2	Breast cancer Coimbra	0.2812	0.3253	0.2713	0.2811	0.2769	0.2808	0.2707	0.2783	0.2799	**0.2672**	0.6220
3	Breast cancer tissue	0.6955	0.7313	0.6866	0.7007	0.6909	0.6844	**0.6788**	**0.6893**	0.7014	0.6817	0.8020
4	Climate	0.0849	**0.0774**	0.0870	0.0845	0.0854	0.0827	0.0844	0.0686	0.0861	0.0804	0.1199
5	Diabetic retinopathy	0.3175	0.3276	0.3160	0.3161	0.3175	0.3114	0.3013	0.3128	0.3159	**0.3011**	0.3440
6	Fri_c0_500_10	0.1531	0.1728	0.1703	0.1579	0.1584	0.1405	0.1513	0.1548	0.1546	**0.1376**	0.2080
7	Fri_c0_1000_10	0.1628	0.1828	0.1574	0.1518	0.1476	0.1585	0.1372	0.1510	0.1560	**0.1287**	0.2208
8	Fri_c1_1000_10	0.1148	0.1594	0.1280	0.1238	0.1190	0.1199	0.1037	0.1134	0.1224	**0.1033**	0.2327
9	German	0.2733	0.2835	0.2711	0.2713	0.2699	0.2688	0.2642	0.2681	0.2701	**0.2592**	0.3208
10	Glass	0.2691	0.2741	0.2682	0.2683	0.2672	0.2678	0.2652	0.2680	0.2687	**0.2647**	0.3117
11	HeartEW	0.1988	0.2318	0.2051	0.2089	0.1874	0.2047	0.1955	0.2009	0.2155	**0.1760**	0.3436
12	IonosphereEW	0.1257	0.1202	0.1339	0.1190	0.1128	0.1146	0.1118	0.1029	0.1175	**0.0906**	0.1797
13	Kc1	0.3088	0.3190	0.3103	0.3101	0.3098	0.3120	0.3040	0.3094	0.3115	**0.3029**	0.3329
14	Kc2	0.7846	0.2333	0.2385	0.2339	0.2259	0.2296	0.2187	**0.2169**	0.2253	0.2235	0.2746
15	Lung cancer	0.1360	0.4715	0.1352	0.1369	0.1334	0.1342	0.1211	0.1335	0.1370	**0.1172**	0.1765
16	Lymphography	0.1587	0.1796	0.1608	0.1531	0.1477	0.1577	0.1441	0.1502	0.1615	**0.1400**	0.2362
17	Meta all	0.3397	0.3723	0.3419	0.3453	0.3187	0.3409	0.3195	0.3318	0.3111	**0.3102**	0.4767
18	Parkinsons	0.1815	0.1945	0.1642	0.1555	0.1536	0.1651	0.1479	0.1497	0.1615	**0.1404**	0.0563
19	Page blocks	0.0458	0.0509	0.0475	0.0485	0.0482	0.0472	**0.0455**	**0.0485**	0.0469	**0.0455**	0.2653
20	Pc1	0.0896	0.0925	0.0918	0.0904	0.0902	0.0899	0.0878	0.0890	0.0908	**0.0869**	0.1071
21	Robot-failures-lp1	0.1464	0.2014	0.1532	0.1487	0.1439	0.1412	0.1138	0.1448	0.1403	**0.0909**	0.2451
22	Robot-failures-lp2	0.2739	0.2903	0.2812	0.2835	0.2741	0.2759	0.2489	0.2671	0.2657	**0.2415**	0.3317
23	Robot-failures-lp3	0.3026	0.3255	0.3043	0.3061	0.2821	0.3058	0.2716	0.2902	0.3056	**0.2493**	0.4307
24	Robot-failures-lp4	0.1464	0.1563	0.1442	0.1400	0.1372	0.1405	0.0912	0.1296	0.1303	**0.0826**	0.1989
25	Robot-failures-lp5	0.3860	0.4043	0.3893	0.3866	0.3888	0.3849	0.3460	0.3858	0.3874	**0.3430**	0.4121
26	Segment	0.0381	0.0464	0.0405	0.0419	0.0401	0.0413	0.0311	0.0396	0.0415	**0.0284**	0.0532
27	SonarEW	0.3957	0.3914	0.4161	0.3876	0.3584	0.4055	0.3347	0.3616	0.3795	**0.3337**	0.5000
28	Space-ga	**0.2107**	**0.2327**	**0.2108**	**0.2107**	**0.2107**	**0.2107**	**0.2107**	**0.2107**	0.2740	0.2109	0.5497
29	SpectEW	0.2858	0.3053	0.2835	0.2880	0.2816	0.2876	0.2727	0.2826	0.2833	**0.2644**	0.3260
30	Stock	0.07757	0.0919	0.0787	0.0775	0.0754	0.0786	**0.0723**	0.0747	0.0773	0.0730	0.1590
31	Vehicle	0.2885	0.3074	0.2866	0.2833	0.2884	0.2868	0.2808	0.2830	0.2855	**0.2731**	0.3494
32	Vowel	**0.0109**	0.0437	0.1012	0.0313	0.1001	**0.0109**	**0.0109**	**0.0109**	0.0224	**0.0109**	0.2720
33	WineEW	0.0824	0.1211	0.0771	0.0800	0.0706	0.0742	0.0706	0.0788	0.0798	**0.0638**	0.0843
34	WDBC	0.0602	0.0700	0.0607	0.0598	0.0587	0.0604	0.0559	0.0596	0.0605	**0.0535**	0.3419
35	Zoo	0.0541	0.0570	**0.0452**	**0.0517**	0.0505	0.0518	0.0488	0.0515	0.0529	0.0485	0.0842

Bold values indicate the best results.

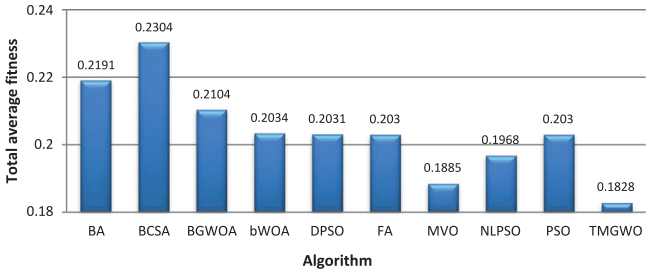

Fig. 9. The total average fitness values among the algorithms.

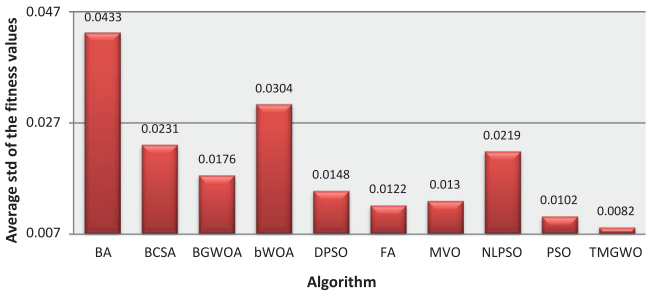

Fig. 10. Comparison among algorithms total average standard deviation for the mean fitness values.

- Grey wolf optimization algorithm with the sigmoid function (GWO-S) and integrated with a two-phase mutation operator with $M_p = 0.1$ (TMGWO1).
- Grey wolf optimization algorithm with the sigmoid function (GWO-S) and integrated with a two-phase mutation operator with $M_p = 0.5$ (TMGWO2).

The best classification accuracy and the number of selected features (#F) that accompany this accuracy for each dataset are the two main performance measures used in our comparison. It is notable that adding the two-phase mutation operator has increased the efficacy of GWO-S. The increase in the mutation probability has resulted in unprecedented superiority over both algorithms (GWO-S and TMGWO1) for all datasets.

A paired sample *t*-test is conducted to the best classification accuracy of the 35 datasets before performing the two-phase mutation operation and after performing the two-phase mutation operator with $M_p = 0.5$ as in Tables 5 and 6. This test is performed in order to prove that the improvement done by the two-phase mutation operation is statistically significant. From the paired samples *t*-test, we find that there is a significant difference in the classification accuracies of the datasets after applying the two-phase mutation operator (Mean = 0.8318, std = 0.1270) than before applying it (Mean = 0.8111, std = 0.1332), where t (34) = 4.428 and p-value = 0.000 which is less than 0.05. Thus, the two-phase mutation operation is integrated with the proposed algorithm with $M_p = 0.5$.

Fig. 5 illustrates a comparison among GWO-S, TMGWO1, and TMGWO2 based on the total average of the best classification accuracy for all the datasets. The proposed algorithm outperforms other algorithms with a total average value of 0.8318. Moreover, the average of the number of the selected features for the best classification accuracy over all the datasets is calculated as shown in Fig. 6. The TMGWO2 algorithm has reduced the average num-

Table 9
Wilcoxon signed ranks test results.

Dataset	Algorithm	No. R+	No. R-	No. ties	Sum_R+	Sum_R-	*p*-value
Fri_c0_1000_10	BA	17	0	3	153	0	0.000
	BCSA	20	0	0	210	0	0.000
	bGWOA	20	0	0	210	0	0.000
	bWOA	18	0	2	171	0	0.000
	DPSO	13	0	7	91	0	0.001
	FA	18	0	2	171	0	0.000
	MVO	6	0	14	21	0	0.023
	NLPSO	18	0	2	171	0	0.000
	PSO	16	0	4	136	0	0.000
HeartEW	BA	20	0	0	210	0	0.000
	BCSA	20	0	0	210	0	0.000
	bGWOA	17	3	0	201	9	0.000
	bWOA	20	0	0	210	0	0.000
	DPSO	15	4	1	160	29.50	0.008
	FA	16	2	2	158	13	0.002
	MVO	14	2	4	131	5	0.001
	NLPSO	18	2	0	207	3	0.000
	PSO	18	2	0	207	3	0.000
Parkinsons	BA	18	2	0	207	3	0.000
	BCSA	18	2	0	203	7	0.000
	BGWOA	15	5	0	195	15	0.001
	bWOA	20	0	0	210	0	0.000
	DPSO	11	6	3	114	38.50	0.071
	FA	18	0	2	171	0	0.000
	MVO	8	4	8	60	18	0.088
	NLPSO	18	2	0	191	19	0.001
	PSO	16	2	2	154	17	0.003
Vehicle	BA	18	2	0	207	3	0.000
	BCSA	20	0	0	210	0	0.000
	BGWOA	17	3	0	201	9	0.000
	bWOA	16	3	1	178	12	0.001
	DPSO	17	3	0	199	11	0.000
	FA	16	4	0	194	16	0.001
	MVO	12	8	0	167	43	0.020
	NLPSO	19	1	0	208	2	0.000
	PSO	15	5	0	180	30	0.005
Wine	BA	17	3	0	203	7	0.000
	BCSA	20	0	0	210	0	0.000
	BGWOA	17	3	0	185	25	0.003
	bWOA	18	2	0	199	11	0.000
	DPSO	15	3	2	146	25	0.007
	FA	15	1	4	122	13.5	0.004
	MVO	14	4	2	129	42	0.057
	NLPSO	18	2	0	203	7	0.000
	PSO	14	2	4	120	16	0.007

ber of the selected features to 7.94 while the total average of the fully selected features is 29.49. Henceforth, we will deal with TMGWO with $M_p = 0.5$ in the next experiments and comparisons. In Fig. 7, the average time for running overall the datasets 20 times is shown.

5.5. Comparison of the proposed algorithm and other metaheuristic algorithms

In the third experiment, we perform a comparison among our proposed algorithm and other nine existing algorithms (BA, BCSA, bGWOA, bWOA, DPSO, FA, MVO, NLPSO, and PSO) for results verification. Some performance measures are used to evaluate the performance of the algorithms: classification accuracy, number of selected features, mean fitness, and standard deviation. In Table 7, a comparison is conducted based on the best classification accuracy achieved by each algorithm. TMGWO precedes the other algorithms in most of the datasets. Furthermore, Fig. 8 shows the average number of selected features accompanied by the best classification accuracy for all the datasets. We can see that the proposed algorithm comes in the second rank with value of 7.9 features. MVO algorithm achieves the lowest average number of features. This is explained by the great priority given to the classification accuracy rather than the number of selected features. However, the difference isn't large between MVO and TMGWO, TMGWO is better as we give more attention to the classification accuracy rather than the number of features selected. Thus, the fitness values are affected by the value of the classification accuracy more than the number of selected features.

Table 8 presents the results obtained by ten algorithms based on the mean fitness values. Inspecting the results, TMGWO has lower fitness values compared to other algorithms. Also, Fig. 9 illustrates the total average fitness over all the datasets. We can see that TMGWO has the minimum average fitness value with a value of 0.1828. From all these experiments, we can conclude that the proposed algorithm's robustness and performance are improved by using the sigmoid function. In addition, using the two-phase mutation enhances the exploitation of TMGWO. K-fold cross-validation is a good choice to avoid the overfitting problem. KNN classifier is a wrapper method that gives high quality solutions and can effectively learn from the training data. The algorithm performance is compared to other nine algorithms. The statistical analyses show the superiority of our proposed algorithm in terms of fitness values, classification accuracy, and number of selected features. In Fig. 10, TMGWO has the smallest value of total average standard deviation, which means that the difference among the fitness val-

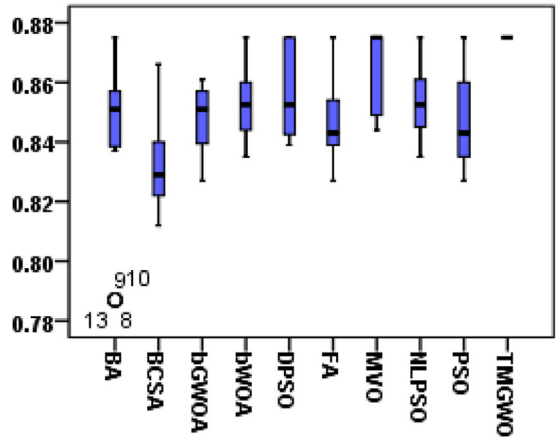

Fig. 11. Boxplot for fri_c0_1000_10 dataset.

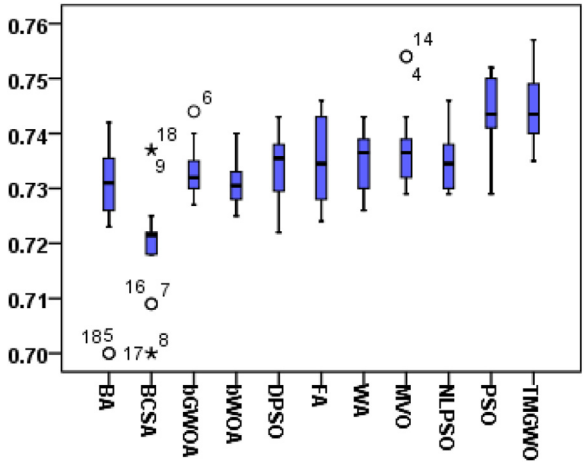

Fig. 12. Boxplot for German dataset.

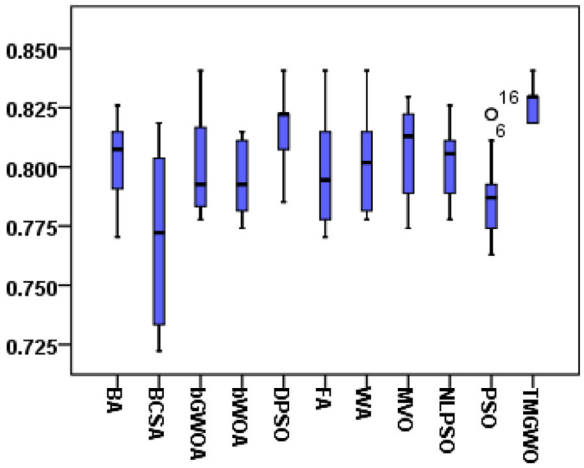

Fig. 13. Boxplot for heartEW dataset.

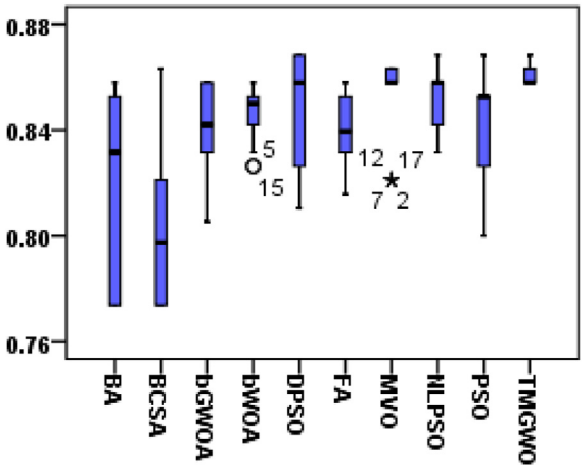

Fig. 14. Boxplot for Parkinson's dataset.

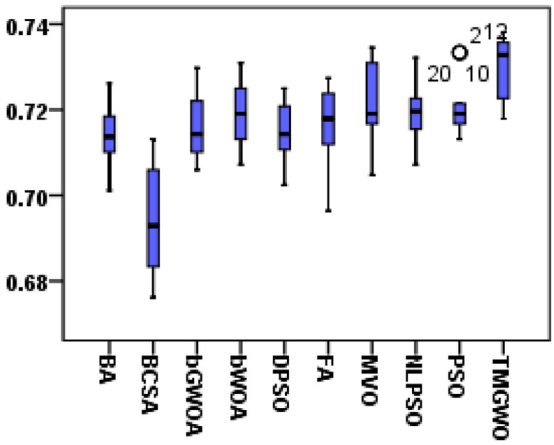

Fig. 15. Boxplot for vehicle dataset.

ues of the datasets is smaller than the other algorithms. A low standard deviation indicates that the fitness values of dataset tend to be close to the mean fitness value. As aforementioned, TMGOWO comes in the first rank according to the obtained mean fitness value. We can also conclude that the worst and the best fitness values are close to the mean value.

In order to detect a statistically significant difference between two sample means, Wilcoxon signed ranks test (Derrac, García, Molina, & Herrera, 2011) is used here. Table 9 presents the Wilcoxon signed ranks test for five randomly selected datasets (fri_c0_1000_10, heartEW, parkinsons, vehicle, and wine). *No.* $R+$ refers to the number of positive ranks in which the proposed algorithm outperforms the compared algorithm which can be BA, BCSA, bGWOA, bWOA, DPSO, FA, MVO, NLPSO, or PSO. On the other side, *No.* $R-$ represents the number of negative ranks in which the proposed algorithm fails to precede the compared algorithm. No. ties are the number of equal ranks for the two algorithms. *Sum_R+* and *Sum_R−* represent the sum of positive and negative ranks respectively. As we can see that *Sum_R+* is greater than *Sum_R−* for the used datasets. Hence, TMGWO shows a significant improvement over DPSO and MVO in parkinsons and for MVO in wine dataset with a level of significance $\alpha = 0.1$. It also shows an improvement over BA, BCSA, bGWOA, bWOA, DPSO, FA, MVO, and PSO with a level of significance $\alpha = 0.05$ in the remaining datasets. The Wilcoxon signed ranks test proves that there is a statistical difference in the performance between the proposed algorithm and the other algorithms as $p - value < \alpha$.

Figs. 11–16 show the boxplots for six datasets, in which the average performance of algorithms can be compared visually. Five elements can be determined from each boxplot: minimum, maximum, median, first quartile, and third quartile of the data. The line inside the box indicates the median value. Note that the boxplots are drawn after running each algorithm 20 times, and they reflect the classification accuracy. It can be seen that TMGWO has higher boxplots compared to the other algorithms. Also, the median of the

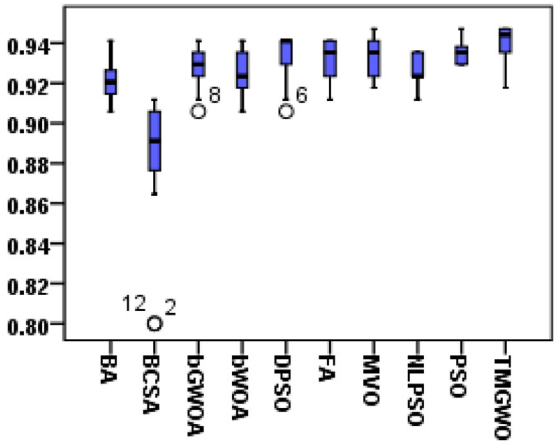

Fig. 16. Boxplot for wine dataset.

proposed algorithm has a greater value compared to the remaining algorithms. Overall, the second-best algorithm changes in different datasets. For instance, the MVO algorithm is the second-best algorithm in Parkinson's dataset, while PSO is the runner up in vehicle dataset. Overall, these box plots allow us to observe that TMGWO is competitive and often superior on the majority of datasets.

From all these experiments, we can conclude that the proposed algorithm's robustness and performance are improved by using the sigmoid function. In addition, using the two-phase mutation enhances the exploitation of TMGWO. K-fold cross-validation is a good choice to avoid the overfitting problem. KNN classifier is a wrapper method that gives high-quality solutions and can effectively learn from the training data. The algorithm performance is compared to the other four algorithms. The statistical analyses show the superiority of our proposed algorithm in terms of fitness values, classification accuracy, number of selected features, and std.

6. Conclusions and future work

This paper proposes a novel algorithm for solving the feature selection problems using the grey wolf algorithm. The algorithm is incorporated with a two-phase mutation operator. The results show how the mutation operator enhances the algorithm capability and efficacy. The first mutation phase seeks to diminish the number of selected features taking into account the high classification accuracy. The second phase seeks to add more features that increase the classification accuracy. Both phases of mutation are done using a small probability. One great merit of the fitness function is choosing the solution with the least number of features if there is more than one solution that has the same highest accuracy. The k-NN classifier is used to train our proposed algorithm. The dataset is split into training and testing data using 10-fold cross-validation. The cross-validation helps us to lessen from the overfitting problem. It is worth mentioning that the algorithm takes much time which is spent during the evaluation process. As the data is split into ten folds, one fold is taken as testing data and the remaining folds are taken as training data. This operation is repeated ten rounds which is considered time-consuming. TMGWO succeeds in obtaining the highest classification accuracy accompanied by a minimum number of selected features compared to FA, WA, MVO, and PSO algorithms. In the future, a parallel version of our proposed algorithm can reduce the time consumption due to the advance of the computing resources as the task can be distributed among multiple processors. Hybridization can be a trend to improve the exploration of the algorithm. We expect that integrating the quantum computing concept with our proposed algorithm can give great results.

Declaration of Competing Interest

None.

References

Al-Tashi, Q., Kadir, S. J. A., Rais, H. M., Mirjalili, S., & Alhussian, H. (2019). Binary optimization using hybrid grey wolf optimization for feature selection. *IEEE Access, 7*, 39496–39508.

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician, 46*(3), 175–185.

Arora, S., & Anand, P. (2019). Binary butterfly optimization approaches for feature selection. *Expert Systems with Applications, 116*, 147–160.

Arora, S., Singh, H., Sharma, M., Sharma, S., & Anand, P. (2019). A new hybrid algorithm based on grey wolf optimization and crow search algorithm for unconstrained function optimization and feature selection. *IEEE Access, 7*, 26343–26361.

Asuncion, A., & Newman, D. (2007). UCI machine learning repository. Irvine, CA: University of California, School of Information and Computer Science. URL [http://www.ics.uci.edu/~mlearn/MLRepository.html].

Chen, K., Zhou, F. Y., & Yuan, X. F. (2019). Hybrid particle swarm optimization with spiral-shaped mechanism for feature selection. *Expert Systems with Applications*.

Cho, J. H., Lee, D. J., Park, J. I., & Chun, M. G. (2013). Hybrid feature selection using genetic algorithm and information theory. *International Journal of Fuzzy Logic and Intelligent Systems, 13*(1), 73–82.

De Souza, R. C. T., dos Santos Coelho, L., De Macedo, C. A., & Pierezan, J. (2018, July). A V-Shaped Binary Crow Search Algorithm for Feature Selection. In *In 2018* IEEE Congress on Evolutionary Computation (CEC) (pp. 1–8). IEEE.

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 and Evolutionary Computation, 1*(1), 3–18.

Eid, H. F. (2018). Binary whale optimisation: An effective swarm algorithm for feature selection. *International Journal of Metaheuristics, 7*(1), 67–79.

Emary, E., Zawbaa, H. M., Grosan, C., & Hassenian, A. E. (2015). Feature subset selection approach by gray-wolf optimization. In *Afro-European Conference for Industrial Advancement* (pp. 1–13). Springer.

Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016). Binary grey wolf optimization approaches for feature selection. *Neurocomputing, 172*, 371–381.

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

Faris, H., Aljarah, I., Al-Betar, M. A., & Mirjalili, S. (2018a). Grey wolf optimizer: A review of recent variants and applications. *Neural computing and applications, 30*(2), 413–435.

Faris, H., Hassonah, M. A., Ala'M, A. Z., Mirjalili, S., & Aljarah, I. (2018b). A multi-verse optimizer approach for feature selection and optimizing SVM parameters based on a robust system architecture. *Neural Computing and Applications, 30*(8), 2355–2369.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning*: Vol. 1. New York: Springer Series in Statistics.

Ghamisi, P., & Benediktsson, J. A. (2015). Feature selection based on hybridization of genetic algorithm and particle swarm optimization. *IEEE Geoscience and Remote Sensing Letters, 12*(2), 309–313.

Gu, S., Cheng, R., & Jin, Y. (2018). Feature selection for high-dimensional classification using a competitive swarm optimizer. *Soft Computing, 22*(3), 811–822.

Hafez, A. I., Hassanien, A. E., Zawbaa, H. M., & Emary, E. (2015, December). Hybrid monkey algorithm with krill herd algorithm optimization for feature selection. In *Computer Engineering Conference (ICENCO), 2015 11th International* (pp. 273–277). IEEE.

Hafez, A. I., Zawbaa, H. M., Emary, E., & Hassanien, A. E. (2016, August). Sine cosine optimization algorithm for feature selection. In *2016 International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)* (pp. 1–5). IEEE.

Hegazy, A. E., Makhlouf, M. A., & El-Tawel, G. S. (2018). Improved salp swarm algorithm for feature selection. *Journal of King Saud University-Computer and Information Sciences*. doi:10.1016/j.jksuci.2018.06.003.

Heidari, A. A., & Pahlavani, P. (2017). An efficient modified grey wolf optimizer with Lévy flight for optimization tasks. *Applied Soft Computing, 60*, 115–134.

Hussien, A. G., Hassanien, A. E., Houssein, E. H., Bhattacharyya, S., & Amin, M. (2019). S-shaped binary whale optimization algorithm for feature selection. In *Recent trends in signal and image processing* (pp. 79–87). Singapore: Springer.

Jain, K., & Purohit, A. (2017). Feature selection using modified particle swarm optimization. *International Journal of Computer Applications, 161*(7), 8–12.

Kashef, S., & Nezamabadi-pour, H. (2015). An advanced ACO algorithm for feature subset selection. *Neurocomputing, 147*, 271–279.

Kennedy, J., & Eberhart, R. C. (1997, October). A discrete binary version of the particle swarm algorithm. In *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on: 5* (pp. 4104–4108). IEEE.

Kohavi, R. (1995, August). A study of cross-validation and bootstrap for accuracy estimation and model selection. *IJCAI, 14*(2), 1137–1145.

Li, T., Dong, H., & Sun, J. (2019). Binary differential evolution based on individual entropy for feature subset optimization. *IEEE Access, 7*, 24109–24121.

Lu, C., Gao, L., Li, X., & Xiao, S. (2017). A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry. *Engineering Applications of Artificial Intelligence, 57*, 61–79.
Mafarja, M. M., & Mirjalili, S. (2017). Hybrid Whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing, 260*, 302–312.
Mafarja, M., Aljarah, I., Faris, H., Hammouri, A. I., Ala'M, A. Z., & Mirjalili, S. (2019). Binary grasshopper optimisation algorithm approaches for feature selection problems. *Expert Systems with Applications, 117*, 267–286.
Mafarja, M., Aljarah, I., Heidari, A. A., Hammouri, A. I., Faris, H., Ala'M, A. Z., et al. (2018). Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. *Knowledge-Based Systems, 145*, 25–45.
Mafarja, M., Eleyan, D., Abdullah, S., & Mirjalili, S. (2017, July). S-shaped vs. V-shaped transfer functions for ant lion optimization algorithm in feature selection problem. In *Proceedings of the international conference on future networks and distributed systems* (p. 14). ACM.
Mafarja, M., Jarrar, R., Ahmad, S., & Abusnaina, A. A. (2018, June). Feature selection using binary particle swarm optimization with time varying inertia weight strategies. In *Proceedings of the 2nd International Conference on Future Networks and Distributed Systems* (p. 18). ACM.
Mafarja, M., & Mirjalili, S. (2018). Whale optimization approaches for wrapper feature selection. *Applied Soft Computing, 62*, 441–453.
Mirjalili, S., & Lewis, A. (2013). S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation, 9*, 1–14.
Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Computing and Applications, 27*(2), 495–513.
Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014a). Grey wolf optimizer. *Advances in engineering software, 69*, 46–61.
Mirjalili, S., Mirjalili, S. M., & Yang, X. S. (2014b). Binary bat algorithm. *Neural Computing and Applications, 25*(3–4), 663–681.
Papa, J. P., Rosa, G. H., de Souza, A. N., & Afonso, L. C. (2018). Feature selection through binary brain storm optimization. *Computers & Electrical Engineering, 72*, 468–481.
Sanjay, R., Jayabarathi, T., Raghunathan, T., Ramesh, V., & Mithulananthan, N. (2017). Optimal allocation of distributed generation using hybrid grey wolf optimizer. *Ieee Access, 5*, 14807–14818.
Sayed, G. I., Darwish, A., & Hassanien, A. E. (2018). A New Chaotic Whale Optimization Algorithm for Features Selection. *Journal of Classification, 35*(2), 300–344.
Sayed, G. I., Hassanien, A. E., & Azar, A. T. (2017). Feature selection via a novel chaotic crow search algorithm. *Neural Computing and Applications, 31*(1), 1–18.
Sayed, G. I., Khoriba, G., & Haggag, M. H. (2018). A novel chaotic salp swarm algorithm for global optimization and feature selection. *Applied Intelligence, 48*(10), 3462–3481.
Sayed, S. A. F., Nabil, E., & Badr, A. (2016). A binary clonal flower pollination algorithm for feature selection. *Pattern Recognition Letters, 77*, 21–27.
Shankar, K., Lakshmanaprabu, S. K., Khanna, A., Tanwar, S., Rodrigues, J. J., & Roy, N. R. (2019). Alzheimer detection using Group Grey Wolf Optimization based features with convolutional classifier. *Computers & Electrical Engineering, 77*, 230–243.
Sharawi, M., Zawbaa, H. M., & Emary, E. (2017, February). Feature selection approach based on whale optimization algorithm. In *2017 Ninth International Conference on Advanced Computational Intelligence (ICACI)* (pp. 163–168). IEEE.
Tabakhi, S., Moradi, P., & Akhlaghian, F. (2014). An unsupervised feature selection algorithm based on ant colony optimization. *Engineering Applications of Artificial Intelligence, 32*, 112–123.
Tawhid, M. A., & Dsouza, K. B. (2018a). Hybrid binary dragonfly enhanced particle swarm optimization algorithm for solving feature selection problems. *Mathematical Foundations of Computing, 1*(2), 181–200.
Tawhid, M. A., & Dsouza, K. B. (2018b). Hybrid binary bat enhanced particle swarm optimization Algorithm for solving feature selection problems. *Applied Computing and Informatics*. doi:10.1016/j.aci.2018.04.001.
Tu, Q., Chen, X., & Liu, X. (2019). Hierarchy Strengthened Grey Wolf Optimizer for Numerical Optimization and Feature Selection. *IEEE Access, 7*, 78012–78028.
Unler, A., & Murat, A. (2010). A discrete particle swarm optimization method for feature selection in binary classification problems. *European Journal of Operational Research, 206*(3), 528–539.
Venkatakrishnan, G., Rengaraj, R., & Salivahanan, S. (2018). Grey wolf optimizer to real power dispatch with non-linear constraints. *CMES-Computer Modeling in Engineering & Sciences, 115*(1), 25–45.
Wang, H., Jing, X., & Niu, B. (2017). A discrete bacterial algorithm for feature selection in classification of microarray gene expression cancer data. *Knowledge-Based Systems, 126*, 8–19.
Xue, B., Zhang, M., & Browne, W. N. (2014). Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. *Applied Soft Computing, 18*, 261–276.
Yang, X. S. (2012, September). Flower pollination algorithm for global optimization. In *International conference on unconventional computing and natural computation* (pp. 240–249). Springer.
Zakeri, A., & Hokmabadi, A. (2019). Efficient feature selection method using real-valued grasshopper optimization algorithm. *Expert Systems with Applications, 119*, 61–72.
Zawbaa, H. M., Hassanien, A. E., Emary, E., Yamany, W., & Parv, B. (2015, December). Hybrid flower pollination algorithm with rough sets for feature selection. In *Computer Engineering Conference (ICENCO), 2015 11th International* (pp. 278–283). IEEE..
Zhang, L., Mistry, K., Lim, C. P., & Neoh, S. C. (2018). Feature selection using firefly optimization for classification and regression models. *Decision Support Systems, 106*, 64–85.
Zhang, Y., Song, X. F., & Gong, D. W. (2017). A return-cost-based binary firefly algorithm for feature selection. *Information Sciences, 418*, 561–574.
Zheng, Y., Li, Y., Wang, G., Chen, Y., Xu, Q., Fan, J., et al. (2018). A Novel hybrid Algorithm for Feature Selection Based on Whale Optimization Algorithm. *IEEE Access, 7*, 14908–14923.