S-shaped versus V-shaped transfer functions for binary Manta ray foraging optimization in feature selection problem

Kushal Kanti Ghosh1 • Ritam Guha1 • Suman Kumar Bera1 • Neeraj Kumar2•3•4•5 • Ram Sarkar1

Received: 15 February 2020 / Accepted: 23 November 2020 / Published online: 3 January 2021
© Springer-Verlag London Ltd., part of Springer Nature 2021

Abstract
Feature selection (FS) is considered as one of the core concepts in the areas of machine learning and data mining which immensely impacts the performance of classification model. Through FS, irrelevant or partially relevant features can be eliminated which in turn helps in enhancing the performance of the model. Over the years, researchers have applied different meta-heuristic optimization techniques for the purpose of FS as these overcome the limitations of traditional optimization approaches. Going by the trend, we introduce a new FS approach based on a recently proposed meta-heuristic algorithm called Manta ray foraging optimization (MRFO) which is developed following the food foraging nature of the Manta rays, one of the largest known marine creatures. As MRFO is apposite for continuous search space problems, we have adapted a binary version of MRFO to fit it into the problem of FS by applying eight different transfer functions belonging to two different families: S-shaped and V-shaped. We have evaluated the eight binary versions of MRFO on 18 standard UCI datasets. Of these, the best one is considered for comparison with 16 recently proposed meta-heuristic FS approaches. The results show that MRFO outperforms the state-of-the-art methods in terms of both classification accuracy and number of features selected. The source code of this work is available in https://github.com/Rangerix/MetaheuristicOptimization.

Keywords Manta ray foraging optimization • Feature selection • Optimization • Classification • Transfer functions • Metaheuristic

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s00521-020-05560-9) contains supplementary material, which is available to authorized users.

Suman Kumar Bera
sumankumarbera.cse.rs@jadavpuruniversity.in
Kushal Kanti Ghosh
kushalkanti1999@gmail.com
Ritam Guha
ritamguha16@gmail.com
Neeraj Kumar
neeraj.kumar@thapar.edu
Ram Sarkar
rsarkar@ieee.org

1 Department of Computer Science and Engineering, Jadavpur University, Kolkata, India

2 Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala, Punjab, India

3 Department of Computer Science and Information Engineering, Asia University, Taichung City, Taiwan

4 School of Computer Science, University of Petroleum and Energy Studies (UPES), Dehradun, Uttarakhand 248007, India

5 King Abdul Aziz University, Jeddah, Saudi Arabia
1 Introduction

In recent times, due to the huge amount of data collected every minute and the need to convert such data into useful information, data mining is considered one of the fastest growing fields of Information Technology [1–4]. With increasing applications of traditional machine learning [5] as well as deep learning [6–8], data mining and knowledge extraction techniques have gained huge popularity over the years as most of these learning techniques are hungry for informative data. Data mining involves: pre-processing, knowledge representation and pattern evaluation [9]. One of the main data pre-processing steps is feature selection (FS). FS can be defined as the process of finding the optimal subset of features that can retain suitably high classification accuracy in representing the original feature set [9]. Through FS, we can remove irrelevant and redundant features from the datasets. In the learning process, irrelevant and redundant features not only mislead the learning algorithm and reduce the performance but also result in increased computational complexity and high storage requirement [10]. FS techniques are broadly classified into two categories [11–15]: filter and wrapper. Filter methods do not require any learning algorithm, rather they use statistical methods to evaluate a feature subset or determine correlation between variables. On the other hand, wrapper models use learning algorithm, like classifiers for the evaluation of feature subsets. Filters are much faster than wrappers, but wrappers are capable to achieve higher classification accuracies than filters [16].

Over the last two decades, several meta-heuristic optimization techniques have been proposed in the field of FS to overcome the limitations of traditional optimization approaches [17]. Nature inspired algorithms have shown high performance in solving the search problems in general [17]. In the literature, many methods have been proposed to mimic the behaviours of animals, birds, fish, wolves, etc. The first proposed FS technique using meta-heuristic approach is Genetic Algorithm (GA) [18]. A chaotic genetic feature selection optimization method (CGFSO) is proposed by the authors in [19], Particle Swarm Optimization (PSO) is proposed in [20] by mimicking the social synergy of a flock of birds. Many FS approaches based on PSO are proposed in [21–23]. Ant Colony Optimization (ACO) [24] is another meta-heuristic algorithm that mimics the behaviour of real ants when searching for the shortest path to a food source. ACO-based FS approaches are proposed in [25, 26]. Following the biological behaviours of bees, Artificial Bee Colony (ABC) optimization algorithm [27] and its FS counterpart [28] are also proposed. Social Mimic Optimization (SMO) Algorithm [13] is proposed by following the human behavior of mimicking someone who is more intelligent, more esteemed, more powerful, and its FS counterpart is proposed in [29]. Some recent optimization algorithms and their FS counterparts are Grey Wolf Optimizer (GWO) [16], Whale Optimization Algorithm (WOA) [30], Ant Lion Optimizer (ALO) [17], Gravitational Search Algorithm (GSA) [31], Social Ski-driver (SSD) Algorithm [32], Sailfish Optimizer (SFO) [33], Ring Theory-based Evolutionary Algorithm (RTEA) [34].

The authors of the work reported in [35] have proposed two binary variants of ALO using S-Shaped and V-Shaped transfer functions. In [36], the authors have proposed six variants of ALO using three S-Shaped and three V-Shaped transfer functions. In [37], authors have proposed binary version of Grasshopper Optimization Algorithm (BGOA) using S-Shaped and V-Shaped functions. In [38], the authors proposed six transfer functions and eight binary versions of PSO.

Since there already exists so many optimization algorithms in literature that perform quite well, the question is whether we need another new optimization approach. According to No Free Lunch theorem for optimization [39], there cannot exist a single algorithm to solve all optimization problems. This, in turn, implies that currently proposed algorithms for FS are not able to solve all FS problems. This motivated us to propose a new FS approach based on a recently proposed meta-heuristic algorithm, Manta ray foraging optimization (MRFO) [40]. We have proposed the binary version of MRFO for FS problems. We have proposed eight different versions of the FS technique based on four S-Shaped and four V-Shaped transfer functions. The methods are evaluated on 18 standard datasets and compared with different recently proposed meta-heuristic FS techniques to validate the performance of the same.

2 Present work

2.1 Manta Ray Foraging Optimization: a brief overview

MRFO is first proposed in [40], inspired by Manta rays, one of the largest known marine creatures. Manta rays feed on plankton, mostly made of microscopic animals from the water. They have different types of foraging strategies:
1. **Chain foraging** [41]: Manta rays observe the plankton position and swim towards it by forming an orderly line. So, the plankton that is missed by the previous manta rays will be devoured by the following manta rays. The higher the concentration of plankton in a position, the better that position is. Now, except the first, all manta rays move towards both the best position and the one in front of it. The mathematical model of chain foraging is given by Eq. (12).

\[
x^d_i(t + 1) = \begin{cases} 
 x^d_i(t) + r \cdot (x^d_{best}(t) - x^d_i(t)) + \varsigma \cdot (x^d_{best}(t) - x^d_i(t)) & \text{if } i = 1 \\
 x^d_i(t) + r \cdot (x^d_{i-1}(t) - x^d_i(t)) + \varsigma \cdot (x^d_{best}(t) - x^d_i(t)) & \text{if } i = 2, \ldots, N 
\end{cases}
\]

(1)

where \( x^d_i(t) \) is the position of \( i^{th} \) individual in \( d^{th} \) iteration at \( d^{th} \) dimension. \( r \) is a random number, \( r \in [0, 1] \). \( \varsigma \) is a weight coefficient. \( x^d_{best} \) is the position with highest plankton concentration. Although the actual best solution is not known, MRF0 assumes the position with highest concentration found so far as the best solution.

2. **Cyclone foraging** [42]: When a school of manta rays finds a plankton with high concentration in deep water, they form a spiral by creating head-to-tail links. In this formation, each manta ray moves towards the plankton position as well as the one in front of it. The mathematical model of cyclone foraging is given by Eq. (34).

\[
x^d_i(t + 1) = \begin{cases} 
 x^d_{best}(t) + r \cdot (x^d_{best}(t) - x^d_i(t)) + \beta \cdot (x^d_{best}(t) - x^d_i(t)) & \text{if } i = 1 \\
 x^d_i(t) + r \cdot (x^d_{i-1}(t) - x^d_i(t)) + \beta \cdot (x^d_{best}(t) - x^d_i(t)) & \text{if } i = 2, \ldots, N 
\end{cases}
\]

(3)

where \( \beta = 2 \epsilon^r \cdot (\epsilon^2 - 1) \cdot \sin(2\pi r_1) \) and \( \epsilon = e^{1/T} \) is the weight coefficient, \( T \) is the maximum number of iterations, \( r_1 \) is a random number, and \( r_1 \in [0, 1] \). All individuals perform a random search considering the best so far plankton position as the reference position. So, it works as a good exploitation for the region found best so far. This can also be modified to improve exploration capability of the MRFO algorithm. We can force the manta rays to look for a new position, by assigning a new random position as their reference position. This results in extensive global search. The mathematical model is given by:

\[
x^d_{rand} = Lb^d + r \cdot (Ub^d - Lb^d)
\]

(5)

\[
x^d_i(t + 1) = \begin{cases} 
 x^d_{rand}(t) + r \cdot (x^d_{rand}(t) - x^d_i(t)) + \beta \cdot (x^d_{best}(t) - x^d_i(t)) & \text{if } i = 1 \\
 x^d_i(t) + r \cdot (x^d_{i-1}(t) - x^d_i(t)) + \beta \cdot (x^d_{best}(t) - x^d_i(t)) & \text{if } i = 2, \ldots, N 
\end{cases}
\]

(6)

where \( x^d_{rand} \) is a random position in the search space, \( Lb^d \) and \( Ub^d \) are lower and upper limits of the \( d^{th} \) dimension, respectively.

3. **Somersault foraging** [43]: This is a random, frequent, local and cyclical movement that helps manta rays optimize plankton intake. The best so far plankton position is used as a pivot and each individual swims to and fro around the pivot and somersaults to a new position. The mathematical model is given as:

\[
x^d_i(t + 1) = x^d_i(t) + S \cdot (r_2 \cdot x^d_{best} - r_3 \cdot x^d_i(t)), i = 1, 2, \ldots, N
\]

(7)

where \( S \) is the Somersault factor, \( r_2 \) and \( r_3 \) are random numbers in \([0, 1]\). As per [40], we have fixed \( S = 2 \).
2.2 Proposed binary MRFO

2.2.1 Manta ray representation

FS is a binary optimization problem [44], since while selecting a feature subset, each feature belonging to the original feature vector has two possibilities: either to be included in that subset or discarded. Hence, a vector of 0’s and 1’s is needed to represent any solution in a FS problem, where 0 represents the corresponding feature not selected and 1 represents selected. The length of the vector is equal to the feature dimension of the dataset under consideration. Accordingly, a manta ray is represented as a binary vector. Problems with continuous real search space can be converted to binary problems by converting their variables to binary variables. In the continuous version of MRFO, manta rays can move around the search space using position vectors within the continuous real domain. In discrete binary search space, update of positions implies switching between 0 and 1. Therefore, we have converted the real position of manta rays to binary values by applying a transfer function to the real position [22].

Transfer function defines the probability of updating the values of binary solution from 0 to 1 and vice-versa. Transfer functions force manta rays to move in a binary space. There are two different types of transfer functions [36]: S-Shaped and V-Shaped. In S-Shaped functions, the positions are updated based on Eq. (8).

\[
X_i^d(t+1) = \begin{cases} 
1 & \text{if rand} < S\text{function}(x_i^d(t+1)) \\
0 & \text{else}
\end{cases}
\tag{8}
\]

where \text{rand} is a random number in [0, 1]. In case of V-Shaped functions, the positions are updated based on Eq. (9).

\[
X_i^d(t+1) = \begin{cases} 
\sim x_i^d(t) & \text{if rand} < V\text{function}(x_i^d(t+1)) \\
x_i^d(t) & \text{else}
\end{cases}
\tag{9}
\]

Table 1 S-Shaped and V-Shaped families of transfer functions

| Name | Transfer function | Name | Transfer function |
|------|-------------------|------|-------------------|
| S1   | \(T(x) = \frac{1}{1+e^{-x}}\) | V1   | \(T(x) = |\tanb(x)|\) |
| S2   | \(T(x) = \frac{1}{1+e^{-x}}\) | V2   | \(T(x) = |\text{erf}(\sqrt{2}x)|\) |
| S3   | \(T(x) = \frac{1}{1+e^{-x}}\) | V3   | \(T(x) = |\frac{x}{\sqrt{1+x^2}}|\) |
| S4   | \(T(x) = \frac{1}{1+e^{x/2}}\) | V4   | \(T(x) = |\frac{x}{\sqrt{1+x^2}}|\) |

Algorithm 1 Pseudocode of MRFO algorithm

while Stop Criterion not Satisfied do
  for \(i = 1...N\) do
    if rand < 0.5 then
      //Cyclone Foraging
      if \(t/T < \text{rand}\) then
        \(x_{\text{rand}} = x_i + \text{rand} \cdot (x_u - x_i)\)
        \(x_i^{d+1} = \begin{cases} 
          x_i^{d+1} & \text{if } x_i^d + x_{\text{rand}}^d - x_i^d \geq 0 \\
          x_i^d & \text{else}
        \end{cases}
        \tag{22}
      \end{cases}
    \end{cases}
  \end{cases}
  \end{cases}
end if
else
  //Chain Foraging
  \(x_i^{d+1} = \begin{cases} 
    x_i^d + \text{rand} \cdot (x_{\text{best}}^d - x_i^d) & \text{if } x_i^d + \text{rand} \cdot (x_{\text{best}}^d - x_i^d) \\
    x_i^d & \text{else}
  \end{cases}
  \tag{22}
end if
end for
//Sommersault Foraging
for \(i = 1...N\) do
  \(x_i^{d+1} = x_i^d + \text{rand} \cdot (x_{\text{best}}^d - x_i^d)\), \(i = 1, 2, ..., N\)
  Compute fitness for \(i\)th individual \(f(x_i(t+1))\)
  if \(f(x_i(t+1)) < f(x_{\text{best}})\) then
    \(x_{\text{best}} = x_i(t+1)\)
  end if
end for
end while
In this work, we have studied the impacts of different transfer functions on the performance of MRFO. We have used eight transfer functions [38], four S-Shaped and four V-Shaped transfer functions. Table 1 shows the mathematical formulas of the 8 transfer functions used in this work. Figure 1 shows their corresponding graphs.

### Table 2: Description of the datasets used in the present work

| Sl. No. | Dataset          | No. of Attributes | No. of Samples | No. of Classes | Dataset domain     |
|---------|------------------|-------------------|----------------|----------------|--------------------|
| 1       | Breastcancer     | 9                 | 699            | 2              | Biology            |
| 2       | BreastEW         | 30                | 569            | 2              | Biology            |
| 3       | CongressEW       | 16                | 435            | 2              | Politics           |
| 4       | Exactly          | 13                | 1000           | 2              | Biology            |
| 5       | Exactly2         | 13                | 1000           | 2              | Biology            |
| 6       | HeartEW          | 13                | 270            | 2              | Biology            |
| 7       | IonosphereEW     | 34                | 351            | 2              | Electromagnetic    |
| 8       | KrvskpEW         | 36                | 3196           | 2              | Game               |
| 9       | Lymphography     | 18                | 148            | 2              | Biology            |
| 10      | M-of-n           | 13                | 1000           | 2              | Biology            |
| 11      | PenglungEW       | 325               | 73             | 2              | Biology            |
| 12      | SonarEW          | 60                | 208            | 2              | Biology            |
| 13      | SpectEW          | 22                | 267            | 2              | Biology            |
| 14      | Tic-tac-toe      | 9                 | 958            | 2              | Game               |
| 15      | Vote             | 16                | 300            | 2              | Politics           |
| 16      | WaveformEW       | 40                | 5000           | 3              | Physics            |
| 17      | WineEW           | 13                | 178            | 3              | Chemistry          |
| 18      | Zoo              | 16                | 101            | 6              | Artificial         |

### 2.2.2 Fitness function

One important aspect of FS techniques is evaluation of the selected feature subset. Since the proposed method is a wrapper-based FS technique, a learning algorithm (classifier) is involved in the evaluation process. We have used a well-known classifier, K-Nearest Neighbour (KNN) classifier [45]...
Fig. 2 Convergence graphs for 18 UCI datasets used in present work
Fig. 3  Accuracy versus Population size for 18 UCI datasets used in present work
Table 3 Classification accuracies (in percentage) of the proposed FS method using eight different transfer functions

| Without FS | MRFOs1 | MRFOs2 | MRFOs3 | MRFOs4 | MRFOv1 | MRFOv2 | MRFOv3 | MRFOv4 |
|------------|--------|--------|--------|--------|--------|--------|--------|--------|
| Breastcancer | 94.4 | 99.29 | 100 | 99.29 | 100 | 100 | 99.29 | 99.29 |
| BreastEW | 96.3 | 99.12 | 96.49 | 97.37 | 96.49 | 98.25 | 96.49 | 97.37 |
| CongressEW | 91.7 | 98.85 | 98.85 | 98.85 | 100 | 100 | 98.85 | 100 |
| Exactly | 67.3 | 90.5 | 100 | 99 | 93 | 100 | 99.5 | 100 |
| Exactly2 | 74.3 | 79.5 | 78.5 | 78.5 | 79 | 79 | 79.5 | 79 |
| HeartEW | 81.5 | 87.04 | 87.04 | 88.89 | 85.19 | 88.89 | 94.44 | 92.59 |
| IonosphereEW | 86.6 | 97.14 | 94.29 | 94.29 | 95.71 | 97.14 | 97.14 | 97.14 |
| KrvskpEW | 91.5 | 98.44 | 97.65 | 97.81 | 97.65 | 97.97 | 97.5 | 97.81 |
| Lymphography | 68.3 | 96.67 | 93.33 | 93.33 | 93.33 | 93.33 | 93.33 | 96.67 |
| M-of-n | 84.9 | 100 | 98.5 | 100 | 100 | 100 | 100 | 100 |
| PenglungEW | 95.1 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| SonarEW | 62 | 95.24 | 100 | 92.86 | 95.24 | 92.86 | 97.62 | 97.62 |
| SpectEW | 83.1 | 92.59 | 90.74 | 92.59 | 96.3 | 94.44 | 92.59 | 90.74 |
| Tic-tac-toe | 71.5 | 79.69 | 78.65 | 83.85 | 80.73 | 81.77 | 80.73 | 82.29 |
| Vote | 87.7 | 98.33 | 100 | 100 | 96.67 | 98.33 | 100 | 100 |
| WaveformEW | 76.8 | 84.9 | 84.7 | 83.5 | 84.7 | 84.3 | 85.2 | 84.6 |
| WineEW | 93.2 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| Zoo | 83.1 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| sum of ranks | 79 | 37 | 45 | 44 | 46 | 35 | 35 | 37 |
| avg of ranks | 4.39 | 2.06 | 2.5 | 2.44 | 2.56 | 1.94 | 1.94 | 2.06 |
| overall rank | 7 | 3 | 5 | 4 | 6 | 2 | 2 | 1 |

The highest classification accuracy is highlighted in bold

Table 4 No. of selected features with the proposed FS method using eight different transfer functions

| Entire dimension | MRFOs1 | MRFOs2 | MRFOs3 | MRFOs4 | MRFOv1 | MRFOv2 | MRFOv3 | MRFOv4 |
|------------------|--------|--------|--------|--------|--------|--------|--------|--------|
| Breastcancer | 9 | 2 | 5 | 6 | 6 | 5 | 6 | 2 |
| BreastEW | 30 | 7 | 14 | 3 | 3 | 18 | 7 | 4 |
| CongressEW | 16 | 5 | 6 | 5 | 9 | 2 | 4 | 2 |
| Exactly | 13 | 8 | 7 | 8 | 6 | 7 | 6 | 6 |
| Exactly2 | 13 | 7 | 9 | 8 | 7 | 9 | 5 | 1 |
| HeartEW | 13 | 5 | 4 | 7 | 4 | 5 | 5 | 4 |
| IonosphereEW | 34 | 7 | 2 | 5 | 6 | 10 | 5 | 2 |
| KrvskpEW | 36 | 28 | 30 | 29 | 26 | 28 | 29 | 13 |
| Lymphography | 18 | 8 | 10 | 4 | 5 | 10 | 5 | 12 |
| M-of-n | 13 | 7 | 9 | 7 | 7 | 6 | 6 | 7 |
| PenglungEW | 325 | 13 | 14 | 110 | 71 | 59 | 65 | 134 |
| SonarEW | 60 | 42 | 28 | 24 | 32 | 18 | 33 | 8 |
| SpectEW | 22 | 17 | 14 | 17 | 5 | 13 | 17 | 4 |
| Tic-tac-toe | 9 | 5 | 7 | 5 | 4 | 6 | 5 | 3 |
| Vote | 16 | 6 | 10 | 6 | 6 | 4 | 6 | 1 |
| WaveformEW | 40 | 33 | 35 | 37 | 28 | 27 | 31 | 20 |
| WineEW | 13 | 10 | 3 | 6 | 4 | 4 | 2 | 2 |
| Zoo | 16 | 8 | 11 | 8 | 8 | 7 | 7 | 5 |
| Sum of ranks | 696 | 218 | 218 | 294 | 239 | 232 | 250 | 103 |
| Avg of ranks | 38.67 | 12.11 | 12.11 | 16.33 | 13.3 | 12.9 | 13.8 | 5.7 |
| Overall rank | 7 | 2 | 2 | 8 | 4 | 3 | 5 | 1 |

Lowest number of selected features for each dataset is highlighted in bold
for this purpose. FS, in general, has two objectives [29]: achieving higher classification accuracy and selecting lower number of features. Basically, FS method removes the irrelevant and redundant features. Higher accuracy is achieved by eliminating the irrelevant features, which otherwise mislead the classification model, thereby lowering the classification accuracy. Both the higher classification accuracy and lower number of selected features indicate that the selected subset is better. However, these two objectives are opposing in nature, since the number of features needs to be minimized and the classification accuracy needs to be maximized. Hence, classification error rate has been chosen here instead of classification accuracy. These two are combined into a single objective by Eq. (10) and used as the fitness function in present work.

\[
\text{Fitness} = \omega \gamma + (1 - \omega) \left( \frac{|R|}{|D|} \right)
\]

(10)

where \(\gamma\) is the classification error of the used classifier, \(|R|\) is number of selected features, \(|D|\) is the total number of features (dimension), \(\omega\) is weight [35], \(\omega \in [0, 1] \).

### 3 Experimental results and discussion

#### 3.1 Datasets

The proposed eight binary MRFO algorithms are evaluated on 18 benchmark UCI datasets [46]. The datasets are described in terms of number of attributes, number of samples, number of classes and dataset domain in Table 2.

#### 3.2 Parameter tuning

The convergence graphs for the 18 datasets with all the eight binary MRFO versions are given in Fig. 2. These graphs depict the value of fitness function vs no. of iterations for all eight versions of the proposed method. It can be noticed that all the methods have converged within 30 iterations. Although it is stated that the ultimate motive of any FS procedure is to increase the classification accuracy while using minimum number of features, the importance of increasing the classification accuracy is more than reduction in the number of features. So, the convergence of the algorithm is guided by the changes in classification accuracy instead of number of features. To proceed with the algorithm, the maximum number of iterations is needed to be fixed beforehand. Through experimental observation, we have seen that for every dataset used in the experimentation, there is little to no improvement in classification accuracy after 30 iterations. So, the algorithm is believed to have converged as it cannot improve the accuracy further after that point. So, we have set the number of iterations as 30 for all the subsequent experiments.

We have also observed the effect of population size on the performance of MRFO in terms of classification accuracy. For this, the proposed approaches are evaluated with population sizes of 5, 10, 20, 30 and 50. Figure 3

### Table 5 Achieved classification accuracy and selected number of features using Naive Bayes and Random Forest classifiers compared to the results obtained using KNN classifier

|               | Naive Bayes                      | Random forest | KNN  |
|---------------|---------------------------------|---------------|------|
|               | Accuracy | #Features | Accuracy | #Features | Accuracy | #Features |
| Breastcancer  | 97.87    | 3         | 97.86    | 3         | 99.29    | 2         |
| BreastEW      | 97.36    | 2         | 95.61    | 2         | 97.37    | 4         |
| CongressEW    | 98.85    | 1         | 98       | 2         | 100      | 2         |
| Exactly2      | 76       | 1         | 79       | 2         | 100      | 6         |
| HeartEW       | 92.6     | 4         | 88.89    | 3         | 92.59    | 4         |
| IonosphereEW  | 92.88    | 8         | 97.14    | 8         | 97.14    | 2         |
| KruskpEW      | 95.31    | 9         | 97.12    | 8         | 97.81    | 13        |
| Lymphography  | 90       | 5         | 93.33    | 8         | 93.33    | 5         |
| M-of-a        | 98.5     | 6         | 100      | 6         | 100      | 6         |
| PenglungEW    | 94.7     | 120       | 93.33    | 140       | 100      | 15        |
| SonarEW       | 80.95    | 23        | 92.86    | 13        | 97.62    | 8         |
| SpectEW       | 90.44    | 4         | 90.74    | 5         | 90.74    | 4         |
| Tic-tac-toe   | 82.7     | 6         | 85.94    | 6         | 82.29    | 3         |
| Vote          | 100      | 3         | 98.33    | 3         | 100      | 1         |
| WaveformEW    | 82.5     | 15        | 83       | 15        | 84.6     | 20        |
| WineEW        | 100      | 3         | 100      | 4         | 100      | 2         |
| Zoo           | 100      | 5         | 100      | 5         | 100      | 5         |
Fig. 4 Comparison between MRFO and state-of-the-art methods based on accuracy.
Fig. 5 Comparison between MRFO and state-of-the-art methods based on no. of selected features
| MRFO | bALO-QR | bALOS-QR | bALOV-QR | ALO | bALO-S | bALO-V | bALO-CE | bALOS-CE | bALOV-CE | BGSA | HGSA | HSGW | RSGW | ASGW |
|------|---------|----------|----------|-----|--------|--------|---------|----------|----------|------|-------|------|------|------|
| MRFO | 1       | 1        | 1        | 1   | 1      | 1      | 1       | 1        | 1        | 1    | 1     | 1    | 1    | 1    |
| bALO-QR | 1   | 1        | 0        | 1   | 0      | 0      | 1       | 0        | 1        | 0    | 0     | 1    | 0    | 0    |
| bALOS-QR | 1  | 1        | 1        | 1   | 1      | 1      | 1       | 1        | 0        | 1    | 1     | 1    | 1    | 1    |
| bALOV-QR | 1  | 0        | 1        | 1   | 0      | 1      | 0       | 1        | 0        | 1    | 0     | 0    | 0    | 0    |
| ALO   | 1       | 1        | 1        | 1   | 1      | 1      | 1       | 1        | 1        | 1    | 1     | 1    | 1    | 1    |
| bALO-1 | 1   | 0        | 1        | 0   | 1      | 1      | 1       | 0        | 1        | 0    | 1     | 1    | 1    | 1    |
| bALO-S | 1   | 1        | 1        | 1   | 0      | 1      | 1       | 0        | 1        | 1    | 1     | 1    | 1    | 1    |
| bALO-V | 1   | 0        | 1        | 0   | 1      | 0      | 1       | 0        | 1        | 0    | 1     | 0    | 0    | 1    |
| bALO-CE | 1  | 0        | 1        | 0   | 0      | 0      | 0       | 0        | 1        | 0    | 0     | 1    | 0    | 1    |
| bALOS-CE | 1 | 1        | 0        | 1   | 1      | 1      | 1       | 1        | 1        | 1    | 1     | 1    | 1    | 1    |
| bALOV-CE | 1 | 1        | 1        | 1   | 1      | 1      | 1       | 1        | 1        | 1    | 1     | 1    | 1    | 1    |
| BGSA  | 1       | 0        | 1        | 0   | 0      | 1      | 0       | 0        | 1        | 0    | 0     | 0    | 0    | 0    |
| HGSA  | 1       | 0        | 1        | 0   | 1      | 1      | 1       | 0        | 1        | 1    | 0     | 0    | 0    | 0    |
| HSGW  | 1       | 1        | 1        | 0   | 1      | 0      | 1       | 1        | 1        | 1    | 0     | 0    | 0    | 0    |
| RSGW  | 1       | 0        | 1        | 0   | 1      | 1      | 1       | 0        | 1        | 0    | 0     | 0    | 0    | 0    |
| ASGW  | 1       | 0        | 1        | 0   | 1      | 1      | 1       | 1        | 1        | 1    | 0     | 0    | 0    | 0    |
depicts the impact of different population sizes on the attained classification accuracy for all the said datasets. It can be noticed that increasing population size does not always increase classification accuracy. Besides, increased population size means higher run time. So, for all our subsequent experiments, the value of the population size is set to 20 as a trade-off between the classification accuracy and the computational cost of the algorithm. All the proposed approaches are implemented in Python3 [47] and experimented on Intel Core-i3 CPU with 4 GB of RAM.

Table 3 represents the classification accuracies achieved (in percent) by the proposed eight FS methods. Without FS indicates the classification accuracy of the dataset without applying any FS method. Without applying any FS method.

In Table 3, we have ranked the proposed methods for each dataset according to the achieved classification accuracy and from this, we have obtained average rank of each proposed method. MRFOv3 has achieved rank one i.e., it achieves higher accuracy than other seven proposed methods for most of the datasets used in this work and MRFOv3 performs in a consistent way i.e., does not fluctuate in achieving higher accuracy.

Table 4 shows the number of features selected using the proposed eight FS methods. Entire dimension column indicates the total number of features in the dataset.

In Table 4, we have ranked the proposed methods for each dataset according to the number of selected features and from this, we have obtained average rank of each method. MRFOv3 has achieved rank one i.e., it selects lower number of features in comparison with other seven methods for most of the datasets used in this work and MRFOv3 performs in a consistent way i.e., does not fluctuate in selecting lower number of features.

Observing both of the tables, on an average, MRFOv3 gives highest classification accuracy and selects the least number of features, and it performs consistently. So, amongst the proposed eight methods, MRFOv3 is considered as the best and we have used for comparing with different state-of-the-art methods.

To prove the effectiveness of the selected KNN classifier, its performance is compared with the performances of other two popular classifiers: Naive Bayes and Random Forest. The results are generated by using different classifiers for computing accuracy used in the fitness function (mentioned in Eq. (10)). Table 5 shows the classification accuracy achieved and the number of features selected, putting all the three classifiers individually in the fitness function of the proposed FS approach. The results prove that KNN works best w.r.t. the datasets considered here. Hence, for further experiments, only KNN classifier is used.

4 Comparison

In this section, we have compared the classification accuracy achieved and number of features selected using the proposed MRFOv3 approach with some methods reported recently in literature. We have compared our method with ALO-based methods: bALO-QR, bALOS-QR, bALOV-QR, ALO, bALO-1, bALO-S, bALO-V, bALO-CE, bALOS-CE, bALOV-CE reported in [17]; GSA-based methods: BQSA, HGSA reported in [48] and hybrid methods using GWO and WOA: HSGW, RSGW, ASGW reported in [49]. In this work we have used 18 UCI datasets and those were used in all the works reported in [17, 48, 49].

Figure 4 shows the classification accuracy achieved by the proposed MRFOv3 method and other methods for each dataset.

From Fig. 4, it is clear that MRFOv3 achieves the highest accuracy in 14 out of 18 datasets. It is also worth mentioning that not only MRFOv3 has outperformed other methods in 77.8% of the datasets, but also it is able to achieve 100% accuracy in 7 cases: CongressEW, Exactly, M-of-n, PenglungEW, Vote, WineEW and Zoo. Other four cases where MRFOv3 fails to achieve the highest accuracy are: BreastEW, Exactly2, IonosphereEW and SpectEW. For SpectEW, MRFOv3 has achieved second highest accuracy. From Fig. 5, we can see that, in case of BreastEW, Exactly2 and IonosphereEW the number of selected features using MRFOv3 is significantly low. Figure 5 shows the number of features selected by the proposed MRFOv3 and other methods for each dataset used in this work.

From Fig. 5, it is clear that MRFOv3 selects the lowest number of features in 15 out of 18 datasets i.e., 83.3% cases. It can also be noted that for BreastEW, IonosphereEW, PenglungEW, SpectEW and WineEW datasets, the number of selected features by MRFOv3 is quite low compared to other methods. Now, the three cases where MRFOv3 fails to select the lowest number of features are: Exactly, M-of-n and WaveformEW. But for all these datasets, MRFOv3 has achieved the highest accuracy. Moreover, for Exactly and M-of-n, MRFOv3 is able to achieve 100% accuracy. So, observing Figs. 4 and 5, we can conclude that no other method outperforms MRFOv3 w.r.t. both in terms of classification accuracy and number of features even for a single dataset.

Taken together, this comparative study establishes that the proposed method outperforms the state-of-the-art methods considered here for comparison. Again, it justifies the No Free Lunch theorem by showing that there can always be a better algorithm for an optimization problem such as FS. It also shows the impact of transfer functions
on the meta-heuristic algorithm used for FS. In Table 6, we have provided the Wilcoxon signed pair test [50] for all possible pair of the methods. Table 6 shows whether the difference between results (in terms of classification accuracy) of two algorithms is statistically significant or not according to Wilcoxon signed rank test. A p value less than 0.05 is denoted by 1 (0 otherwise) which implies that difference between results is statistically significant, 0 means it is not.

5 Conclusion and future work

FS is considered as an important pre-processing step in the domain of Machine Learning and Data Mining. In this article, we have proposed the binary version of MRFO for selection of optimal subset of features. Since MRFO is reported to be suitable for continuous search space problems, we have modified MRFO for the binary search space to fit it into the problem of FS using eight different transfer functions which belong to two different families: S-shaped and V-shaped. We have applied the eight binary versions of MRFO methods on 18 standard UCI datasets. We have shown the impact of transfer functions on the classification accuracy obtained and number of features selected.

We have considered both the classification accuracy and number of features selected while designing the fitness function. To measure the effectiveness of the proposed approaches, we have considered the best among the eight versions of MRFO and compared it with 16 recently proposed meta-heuristic FS approaches. The results show that MRFO outperforms the state-of-the-art methods in terms of both classification accuracy and number of selected features.

In future, we plan to propose an embedded FS method where binary MRFO can be united with a compatible filter method. We can also aim to hybridize MRFO with other meta-heuristic FS methods. Besides, as the proposed FS version of MRFO is a general approach, so this can easily be applied to other standard pattern classification problems where usually large dimension of feature vectors are used.

Compliance with ethical standards

Conflict of interest We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

References

1. Han J, Pei J, Kamber M (2011) Data mining: concepts and techniques. Elsevier, Amsterdam
2. Ghosh KK et al (2020) A two-stage approach towards protein secondary structure classification. Med Biol Eng Comput 58(8):1723–1737. https://doi.org/10.1007/s11517-020-02194-w
3. Kowal M, Skobel M, Nowicki N (2018) The feature selection problem in computer-assisted cytology. Int J Appl Math Comput Sci 28(4):759–770. https://doi.org/10.2478/amcs-2018-0058
4. Borowik G (2018) Optimization on the complementation procedure to- wards efficient implementation of the index generation function. Int J Appl Math Comput Sci 28(4):803–815. https://doi.org/10.2478/amcs-2018-0061
5. Kumar N, Makkar A (2020) Machine learning in cognitive IoT. CRC Press, Boca Raton
6. Garg S et al (2019) Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in SDN: a social multimedia perspective. IEEE Trans Multimed 21(3):566–578. https://doi.org/10.1109/tmm.2019.2893549
7. Garg S et al (2019) A hybrid deep learning-based model for anomaly de- tection in cloud datacenter networks. IEEE Trans Netw Serv Manag 16(3):924–935. https://doi.org/10.1109/tnsm. 2019.2927886
8. Migliani A, Kumar N (2019) Deep learning models for traffic ow prediction in autonomous vehicles: a review, solutions, and challenges. Veh Commun 20:100184. https://doi.org/10.1016/j. vehcom.2019.100184
9. Liu H, Motoda H (2012) Feature selection for knowledge discovery and data mining, vol 454. Springer, Berlin
10. Guha R et al. (2020) “Mutually informed correlation coefficient (MICC)—a new filter based feature selection method. In: 2020 IEEE Calcutta Conference (CALCON), pp. 54–58. https://doi.org/10.1109/CALCON49167.2020.9106516
11. Ghosh M et al (2018) Genetic algorithm based cancerous gene identification from microarray data using ensemble of filter methods. Med Biol Eng Comput 57(1):159–176. https://doi.org/10.1007/s11517-018-1874-4
12. Blachnik M (2019) Ensembles of instance selection methods: a comparative study. Int J Appl Math Comput Sci 29(1):151–168. https://doi.org/10.2478/amcs-2019-0012
13. Balochian S, Baloochian H (2019) Social mimic optimization algorithm and engineering applications. Expert Syst Appl 134:178–191. https://doi.org/10.1016/j.eswa.2019.05.035
14. Guha R et al (2020) Embedded chaotic whale survival algorithm for filter-wrapper feature selection. Soft Comput 24(17):12821–12843. https://doi.org/10.1007/s00500-020-03183-1
15. Guha R et al (2020) Introducing clustering based population in binary gravitational search algorithm for feature selection. Appl Soft Comput 93:106341. https://doi.org/10.1016/j.asoc.2020.106341
16. Al-Tashi Q et al (2019) Binary optimization using hybrid grey wolf optimization for feature selection. IEEE Access 7:39496–39508
17. Mafarja MM, Mirjalili S (2019) Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection. Soft Comput 23(15):6249–6265
18. Il-Seok O, Lee J-S, Moon B-R (2004) Hybrid genetic algorithms for feature selection. IEEE Trans Pattern Anal Mach Intell 26(11):1424–1437. https://doi.org/10.1109/tpami.2004.105
19. Chen H et al (2013) A heuristic feature selection approach for text categorization by using chaos optimization and genetic algorithm. Math Probl Eng. https://doi.org/10.1155/2013/524017
20. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN’95-international conference on neural networks, vol 4. IEEE, pp 1942–1948
21. Chakraborty B (2008) Feature subset selection by particle swarm optimization with fuzzy fitness function. In: 2008 3rd international conference on intelligent system and knowledge engineering, vol 1, pp 1038–1042. https://doi.org/10.1109/ISKE.2008.4731082
22. Lee S et al (2008) Modified binary particle swarm optimization. IEEE Trans Evol Comput 9:1–14
23. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN’95-international conference on neural networks, vol 4. IEEE, pp 1942–1948
24. Dorigo M, Maniezzo V, Colorni A (1996) Ant system optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern Part B (Cybern) 26(1):29–41
25. Ke L, Feng Z, Ren Z (2008) An efficient ant colony optimization approach to attribute reduction in rough set theory. Pattern Recognit Lett 29(9):1351–1357
26. Chen Y, Miao D, Wang R (2010) A rough set approach to feature selection based on ant colony optimization. Pattern Recognit Lett 31(3):226–233. https://doi.org/10.1016/j.patrec.2009.10.013
27. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Tech. rep. Technical report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department
28. Wang X et al (2007) Feature selection based on rough sets and particle swarm optimization. Pattern Recognit Lett 28(4):459–471
29. Ghosh KK et al (2020) Improved binary sailfish optimizer based on adaptive β-Hill climbing for feature selection. IEEE Access 8:83548–83560. https://doi.org/10.1109/access.2020.2991543
30. Mafarja M, Mirjalili S (2018) Whale optimization algorithms for wrapper feature selection. Appl Soft Comput 62:441–453
31. Chatterjee B et al (2020) Late acceptance hill climbing based social ski driver algorithm for feature selection. IEEE Access 8:75393–75408. https://doi.org/10.1109/ACCESS.2020.2988157
32. Ghosh KK et al (2020) Improved binary sailfish optimizer based on adaptive β-Hill climbing for feature selection. IEEE Access 8:83548–83560. https://doi.org/10.1109/access.2020.2991543
33. Mafarja M et al (2019) Binary grasshopper optimisation algorithm approaches for feature selection problems. Expert Syst Appl 117:267–286
34. Oh B, Kwon S, Lee S, Lee M, Lee J (2017) Hybrid of harmony search algorithm and ring theory-based evolutionary algorithm for feature selection. IEEE Access 5:102629–102645. https://doi.org/10.1109/ACCESS.2020.2999093
35. Emay E, Zawbaa HM, Hassanien AE (2016) Binary grey wolf optimization approaches for feature selection. Neurocomputing 172:371–381. https://doi.org/10.1016/j.neucom.2015.06.083
36. Mafarja Majdi, et al. (2019) 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, pp 1–7
Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.