AUT Journal of Modeling and Simulation

AUT J. Model. Simul., 55(1) (2023) 17-38
DOI: 10.22060/miscj.2023.21474.5288

Improved Equilibrium Optimizer using Density-based Population and Entropy Operator for Feature Selection

M. Amiri Ebrahimabadi, B. Mohammad Hasani Zade, N. Mansouri*

Department of Computer Science, Shahid Bahonar University of Kerman, Kerman, Iran

ABSTRACT: As the number of features in practical applications grows, the feature selection process becomes increasingly important. Since the feature selection problem is NP-hard, no exact algorithm can determine the optimal subset over a reasonable period of time. However, traditional feature selection methods are time-consuming and tend to get stuck in local optima. Unlike traditional search techniques, metaheuristics are more effective at exploring and exploiting the search domain because they use several operators. Besides these behaviors of metaheuristics, we present an improved Equilibrium Optimizer algorithm using the density of population and entropy operator, which has proven a good exploration ability to provide a promising candidate solution. After that, a new feature selection model is developed based on the improved Equilibrium Optimizer algorithm. K-nearest neighbors is used as an evaluator for the new solutions. In order to test the performance of the proposed algorithm, simulation experiments are conducted on a set of 14 standard test functions containing both unimodal and multimodal functions. To evaluate the effectiveness of the proposed algorithm, 15 UCI benchmark datasets and five metaheuristics, GA, CS, GSA, RDA, and BBA are applied. The experimental results revealed the effectiveness of our approach in terms of accuracy performance for the feature selection process.

Review History:

Received: Jun. 07, 2022
Revised: Jul. 18, 2023
Accepted: Oct. 07, 2023
Available Online: Oct. 10, 2023

Keywords:

Feature selection
equilibrium optimizer
classification
metaheuristic

1- Introduction

A pre-processing method like Feature Selection (FS) becomes an essential and challenging part of data mining when the volume of data increases significantly [1]. Machine learning tasks require feature selection to select a subset of essential features for optimizing models in terms of accuracy and generality [2]. An illustration of the feature selection diagram can be found in Fig. 1. There are three main types of feature selection methods: filter, wrapper, and embedded. By using a filter model, the relevancy of features is taken into account without using a learning algorithm. As a result, these methods are generally fast. According to this model, features are organized and evaluated according to information-theoretical measures, and those with the highest ranks are selected. The wrapper approach evaluates a set of prominent features using a classifier. To determine which features are the most accurate, the wrapper model uses a given learning model to assess a subset of features. An embedded approach considers feature selection as a part of the machine-learning process.

Traditionally, optimization problems have been solved by deterministic mathematical methods, which suffer from one significant limitation: local optimization traps. Real-life optimization problems cannot be solved using these techniques, so stochastic optimization strategies are becoming increasingly popular [3]. Several practical real-world optimization problems have been solved using non-deterministic methods, also known as metaheuristic algorithms. For example, feature selection, engineering, medical, and economics are among the inheritors [4]. In solving optimization problems, metaheuristic algorithms are preferred over other methods due to four important factors. i) Simple ideas in nature inspired metaheuristic algorithms, which are easy to implement. These algorithms are also easy to learn. ii) Since these algorithms are flexible, they can be used to solve a variety of optimization problems without altering their structure. iii) Metaheuristic methods rarely require derivation. As a result, we don't need to calculate the derivative of the search space to find the optimal solution. iv) The metaheuristic algorithm avoids local optimality in comparison with conventional methods [5]. According to the No Free Lunch theorem (NFL), no optimization method is suitable for every problem. A metaheuristic technique is therefore a promising research topic for solving feature selection problems.

This study contributed the following points:

Present an improved version of EO based on the density of population and entropy operator to overcome the slow convergence rate and the entrapment in local optima problems and to achieve an effective and robust search mechanism.

Develop a new feature selection method using improved EO to locate the best feature subset.

*Corresponding author's email: n.mansouri@uk.ac.ir

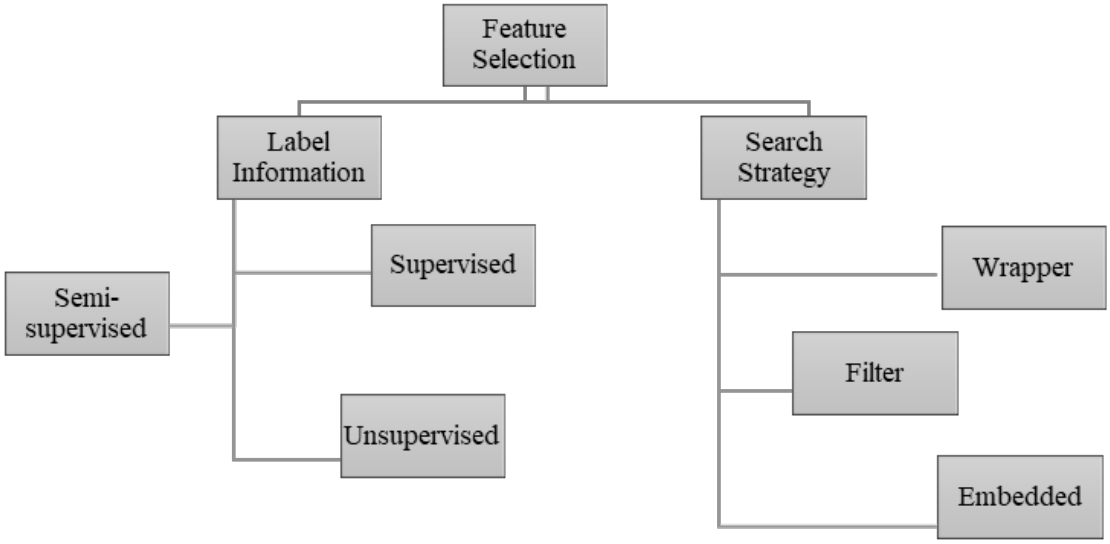

Fig. 1. Feature selection diagram.

Evaluate the improved EO algorithm using 14 well-known functions.

Compare the proposed feature selection approach with other well-known and recent meta-heuristic algorithms, Genetic Algorithm (GA), Cuckoo Search (CS), Gravitational Search Algorithm (GSA), Red Deer Algorithm (RDA), and Binary Bat Algorithm (BBA).

The rest of the paper is organized into the following sections: Section 2 presents a review of the related works. Section 3 describes the Equilibrium Optimizer. Section 4 explains the improved EO structure. Section 5 explains the proposed method in FS. Section 6 illustrates the parameter setup and performance results. Finally, conclusions and future work are presented in Section 7.

2- Related Works

Metaheuristic algorithms are used extensively to select features, mainly due to their ease of application and high accuracy [6]. In recent years, optimization algorithms have been used to solve high-dimensional feature selection problems in many fields. Optimization algorithms enhanced classification accuracy and reduced selected features by enhancing their efficiency. Some of these works are Particle Swarm Optimization (PSO) [7], GA [7], GSA [8], and Bat Algorithm (BA) [9]. There are two steps in the search behavior of optimization algorithms: exploration and exploitation. An optimization algorithm searches the appropriate scopes of a search space during exploration and exploitation. This fine-tuning determines the goodness of an algorithm. The stochastic nature of these elements makes it difficult to compensate for them [10]. Table 1 summarizes some of the meta-heuristic algorithm-based work done in the feature selection area.

Maleki *et al*. [11] used K-Nearest Neighbor (KNN) and GA for efficient feature selection to lower the dataset dimensions and enhance the classifier rate in exploring the step of patients' disease. The best value for k is determined using a practical procedure to improve the algorithm's accuracy. Huang and Wang [12] simultaneously optimized the parameters and feature subset without harming the Support Vector Machine (SVM) classification accuracy. A training process for SVM affects the classification accuracy by selecting kernel parameters. A GA-based approach improves classification accuracy dramatically over the Grid algorithm. SVMs have fewer input features.

Alotaibi [13] proposed a stock market model that includes some techniques like feature extraction and prediction. Additionally, the indexed data are computed using standard indicators like the Average True Range. Furthermore, stock movement is predicted using the selected features. Based on the optimized neural network, the final results are formulated. With the proposed Red Deer Adopted Wolf Algorithm (RDAWA), the training of Neural Networks (NN) can be made more accurate. Nagpal *et al*. [8] investigated a relatively new evolutionary computation method called Gravitational Search Algorithm (GSA) for feature selection. Due to the numerous features in medical data, efficient feature selection methods are required for disease prediction. In the proposed wrapper-based approach, the number of features is reduced and the prediction accuracy is improved by using GSA and KNN. Tarade *et al*. [14] proposed the Hybrid GSA (HGSA) method to solve the feature selection problem. To enhance exploitative and exploitative abilities, the authors applied GA operators (i.e., crossovers and mutations). The feature selection problems have been solved using several evolutionary algorithms.

Table 1. Summary of different feature selection algorithms based on meta-heuristic.

Authors	Techniques	Classifier	Comments	Domain
(Maleki et al., 2021)	GA	Decision Tree, KNN	• Lung cancer dataset: 1000 samples, each with 23 features • Dataset is classified into three levels low, medium, high	• lung cancer/Medical
(Huang and Wang, 2021)	GA, Grid Algorithm	SVM	• Classified using 10 k-fold cross-validation	• No specific domain • Tested on 11 datasets
(Alotaibi, 2021)	RDA, GWO, RDAWA	RF, SVM, Optimized Neural Network	• Data for testing the proposed method was collected from three companies • The classifiers were trained with the extracted features.	• Saudi Stock Market
(Nagpal et al., 2017)	GSA, PSO, GA	KNN, CFS	• 3 datasets were used for the experiment • Accuracy increased after FS	• Biomedical Data
(Mafarja et., 2019)	BGSA, HGSA, GWO, PSO, GA	Decision Tree, KNN	• Proposed GSA-based optimizer for the first time with evolutionary crossover and mutation schemas • Proposed methods merit in terms of exploitation and exploration	• No specific domain • Tested on 18 datasets

3- Equilibrium Optimizer

The physical laws of nature are used to develop an Equilibrium Optimizer (EO). It uses the uncertain condition mass balance equation to track the closest nonreactive element concentration. In general, mass balance can be expressed as a first-order differential equation [15].

$$V\frac{dC}{dt} = QC_{eq} - QC + G \tag{1}$$

which V is the control volume, C is the concentration of the control volume, $V\frac{dC}{dt}$ is the control volume changing speed, Q is the flow speed, C_{eq} is the concentration inside the control volume at an equilibrium state in which no generation is available, and G is the mass generation speed. A steady equilibrium state can be achieved when $V\frac{dC}{dt}$ leaning toward zero, and Eq. (1) can be rearranged to solve for C as a function of t [15].

$$\frac{dC}{\lambda C_{eq} - \lambda C + \frac{G}{V}} = dt \tag{2}$$

In addition, $\lambda = Q/V$ as the turn-over rate [1].

$$\int_{c_0}^{c} \frac{dC}{\lambda C_{eq} - \lambda C + \frac{G}{V}} = \int_{t_0}^{t} dt \tag{3}$$

where C_0 and C are the concentration levels at t_0 and t.

$$C = C_{eq} + (C_0 - C_{eq})F + \frac{G}{\lambda V}(1-F) \tag{4}$$

In Eq. (4), F plays an important role in balancing exploration and exploitation which is calculated as follows [1]:

$$F = e^{-\lambda(t-t_0)} \tag{5}$$

3- 1- Initialization

Like most meta-heuristic algorithms, EO commences the optimization process with the initial population. Based on the number of particles and dimensions, the initial concentrations are determined as follows [1]:

$$C_i^{initial} = C_{\min} + rand_i(C_{\max} - C_{\min})\ i = 1,2,...,n \quad (6)$$

$C_i^{initial}$ is the initial concentration vector of the *i-th* particle, $C_{\min}$ and $C_{\max}$ represents the minimum and maximum values for the dimensions, is a random vector in the interval of [0,1], and n is the number of particles as the population. In order to select equilibrium candidates, particles are estimated for their fitness functions.

3- 2- Equilibrium Pool (C_{eq})

Some equilibrium candidates are bound when the optimization process begins to provide a pattern for finding particles. There are five candidates based on experiments under varying problem situations. Thus, four of these candidates represent the best particles identified during optimization, and the fifth represents the mean of the other four. These four candidates help the EO have better exploration, and the fifth one helps it have better exploitation. It's also possible to use any number of candidates in experiments, but we use five candidates and group them in a vector called equilibrium pool in this paper:

$$\vec{C}_{eq,pool} = \{\vec{C}_{eq(1)}, \vec{C}_{eq(2)}, \vec{C}_{eq(3)}, \vec{C}_{eq(4)}, \vec{C}_{ave}\} \quad (7)$$

3- 3- Exponential Term (F)

A reasonable balance between exploration and exploitation will be made possible with the exponential term. As a result, turnover rates are time-dependent and are subject to change, so they are treated as random numbers in the range [0, 1]. The formula is shown below [1]:

$$\vec{F} = e^{-\vec{\lambda}(t-t_0)} \quad (8)$$

F is a function of t and changes (decreases) with the number of iterations [1]:

$$t = (1 - \frac{Iter}{Max\_Iter})^{(a_2 \frac{Iter}{Max\_Iter})} \quad (9)$$

In Eq. (9), *Iter* is the current iteration and *Max_Iter* is the maximum iteration. Where a_2 is the constant to manage exploration power. Parameter t_0 is used to avoid convergence at local minima using slowing down the routine [1].

$$\vec{t}_0 = \frac{1}{\vec{\lambda}} \ln(-a_1 sign(\vec{r} - 0.5)[1 - e^{-\vec{\lambda}t}]) + t \quad (10)$$

So a_1 handles the exploration abilities, r is a random number in the range of [0, 1] and the sign executes the direction of the search.

Eq. (10) takes place in Eq. (8):

$$\vec{F} = a_1 sign(\vec{r} - 0.5)[e^{-\vec{\lambda}t} - 1] \quad (11)$$

3- 4- Generation Rate (G)

As a result of the generation rate, the EO can explore the search domain more easily. In EO, the generation rate (G) is as follows [1]:

$$\vec{G} = \vec{G}_0 e^{-\vec{\lambda}(t-t_0)} = \vec{G}_0 \vec{F} \quad (12)$$

$$\vec{G}_0 = \overrightarrow{GCP}(\vec{C}_{eq} - \vec{\lambda}\vec{C}) \quad (13)$$

$$\overrightarrow{GCP} = \begin{cases} 0.5r_1 & r_2 \geq GP \\ 0 & r_2 < GP \end{cases} \quad (14)$$

The *GCP* is the generation rate control and is computed using Eq. (14). In Eq. (14), r_1 and r_2 are two random vectors in the range of [0, 1].

The updating rule of EO is as follows [1]:

$$\vec{C} = \vec{C}_{eq} + (\vec{C} - \vec{C}_{eq}).\vec{F} + \frac{\vec{G}}{\vec{\lambda}V}(1 - \vec{F}) \quad (15)$$

where F is supposed to be an exponential term, G is the generated rate, C_{eq} is a random candidate from the equilibrium pool, and V is a constant unit. Figure 2 shows the pseudo-code for the EO algorithm.

4- The Proposed Method (IMEO2)

Varzaneh et al. [16] proposed Improved Equilibrium Optimization (IMEO), which increases the exploration capability of the original EO by using entropy-based operators. The operation is based on chemistry-physics principles. By using this operator, they claim that the algorithm can be automatically explored and exploited. Hence, if an agent's fitness does not improve after k consecutive generations, it may be stuck in local optima. It is possible to calculate the improvement of an agent by using the following formula:

$$IMP(it) = \frac{fit(it)}{fit(it - k)} \quad (16)$$

Defining the particle fitness at the time of (*it*) and (*it* -

Equilibrium Algorithm Pseudo Code

Initialize the particle's populations i=1,…,n
Assign equilibrium candidates' fitness a large number
Assign free parameters a1=2; a2=1; GP=0.5;
While Iter < Max_iter
For i=1: number of particles (n)
Calculate fitness of i[th] particle
If fit($\vec{C}_i$) fit($\vec{C}_{eq\,1}$)
Change with $\vec{C}_i$ and fit ($\vec{C}_{eq1}$) with fit ($\vec{C}_i$)
Elseif fit($\vec{C}_i$) > fit($\vec{C}_{eq1}$) & fit($\vec{C}_i$) < fit($\vec{C}_{eq2}$)
Change $\vec{C}_{eq2}$ with $\vec{C}_i$ and fit ($\vec{C}_{eq2}$) with fit ($\vec{C}_i$)
Elseif fit($\vec{C}_i$) > fit($\vec{C}_{eq1}$) & fit($\vec{C}_i$) > fit($\vec{C}_{eq2}$) & fit($\vec{C}_i$) < fit($\vec{C}_{eq3}$)
Change $\vec{C}_{eq3}$ with $\vec{C}_i$ and fit ($\vec{C}_{eq3}$) with fit ($\vec{C}_i$)
Elseif fit($\vec{C}_i$) > fit($\vec{C}_{eq1}$) & fit($\vec{C}_i$) > fit($\vec{C}_{eq2}$) & fit($\vec{C}_i$) > fit($\vec{C}_{eq3}$) & fit($\vec{C}_i$) < fit($\vec{C}_{eq4}$)
Change $\vec{C}_{eq4}$ with $\vec{C}_i$ and fit ($\vec{C}_{eq4}$) with fit ($\vec{C}_i$)
Endif
Endfor
$\vec{C}_{ave} = (\vec{C}_{eq1} + \vec{C}_{eq2} + \vec{C}_{eq3} + \vec{C}_{eq4})/4$
Construct the equilibrium pool $\vec{C}_{eq,pool} = \{\vec{C}_{eq1}, \vec{C}_{eq2}, \vec{C}_{eq3}, \vec{C}_{eq4}, \vec{C}_{ave}\}$
If (Iter > 1) => Accomplish memory saving
Assign $t = (1 - \frac{Iter}{Max\_Iter})^{(a_2 \frac{Iter}{Max\_Iter})}$ *Eq. (9)*
For i=1: number of particles(n)
Randomly choose one candidate from the equilibrium pool (vector)
Generate random vectors of $\vec{r}$, $\vec{\lambda}$ Eq. (11)
Construct $\vec{F} = a_1 sign(\vec{r} - 0.5)[e^{-\vec{\lambda}t} - 1]$ Eq. (11)
Construct $\overrightarrow{GCP} = \begin{cases} 0.5r_1 & r_2 \geq GP \\ 0 & r_2 < GP \end{cases}$ Eq. (14)
Construct $\vec{G}_0 = \overrightarrow{GCP}(\vec{C}_{eq} - \vec{\lambda}\vec{C})$ Eq. (13)
Construct $\vec{G} = \vec{G}_0 \vec{F}$ Eq. (12)
Update concentration $\vec{C} = \vec{C}_{eq} + (\vec{C} - \vec{C}_{eq}).\vec{F} + \frac{\vec{G}}{\vec{\lambda}V}(1 - \vec{F})$ Eq. (15)
Endfor
Iter=Iter+1
Endwhile

Fig. 2. Pseudo code of EO algorithm [1].

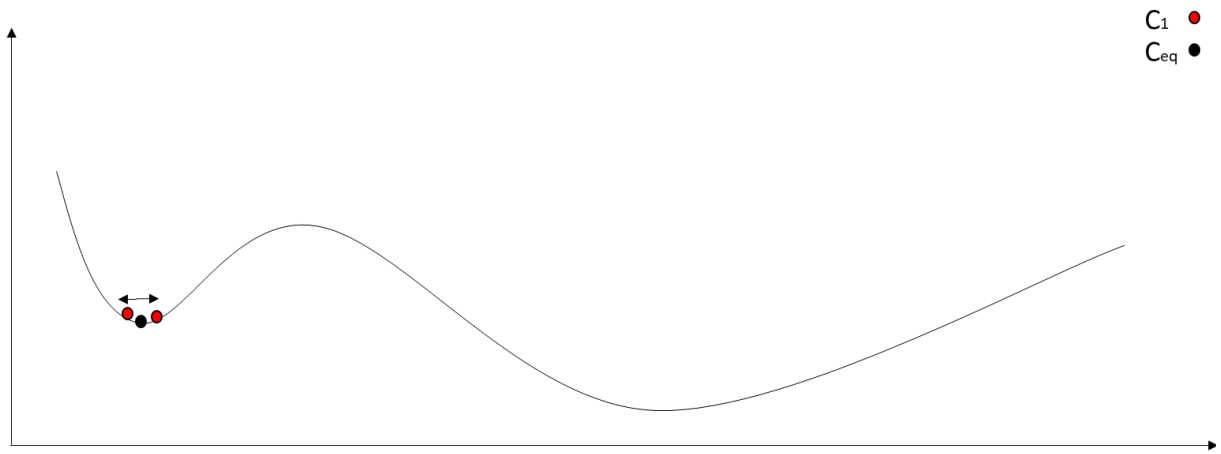

Fig. 3. An example of entropy-based operator performance.

k) as *fit (it)* and *fit (it - k)*, respectively. It is obvious if the value of IMP for some agent tends to zero (below epsilon), this agent is not capable of exploring new areas in the search space. As a result, the new position for this agent is:

$$C_{new} = \begin{cases} C_{eq} \pm shift & IMP < \varepsilon \\ C_{old} & Otherwise \end{cases} \quad (17)$$

Where C_{new} represents the agent's new position and C_{old} represents its previous position. C_{eq} indicates the best position so far, and *shift* indicates how far it is from the current position. Based on the Eq. (18), we can determine the shift value.

$$shift = \frac{|LB| + |UB|}{p} \quad (18)$$

The lower bound and upper bound are determined by $|LB|$ and $|UB|$, respectively.

Nevertheless, this approach can enhance the search ability of the agents, especially for unimodal functions (functions with a single global optima). It often fails to find global optima in multimodal functions (functions that contain more than one local optimum and one global optima). The best agent cannot escape a local optimum by changing position with an entropy-based operator.

Figure 3 shows an example of using an entropy-based operator to solve a local optima problem for agent C_1. The shift value changes the position of C_1 only slightly, while the agent remains in local optima. To overcome the previous drawback, we propose a new Improved Equilibrium Optimization (IMEO2) based on a density-based population operator. According to this operator, if the average distance between all agents in local optima is smaller than a value, then the stuck agent (such as C_1 in Fig. 3) will move toward a random agent in the search space. The average distance between agents and the stuck agent's new position can be found in Eq. (19) and Eq. (20).

$$D_{Avg} = \frac{1}{N} \sum_{\substack{i=1 \\ j=1 \\ j \neq i}}^{N} \sqrt{(C_i - C_j)^2} \quad (19)$$

$$C_{new} = C_{old} + rand \times (C_{old} - C_{rand}) \quad (20)$$

In this case, N is the size of the population and C_{rand} is a random agent in the search space.

5- Proposed Feature Selection Method

Figure 4 illustrates the steps involved in selecting features using IMEO2. The proposed IMEO2 seeks to identify the most informative set of features. Experimental results indicate that the IMEO2 algorithm is highly efficient and performs well.

The high efficiency of IMEO2 comes from its balancing strength in exploration and exploitation ability. The IMEO2 algorithm updates concentrations by selecting a random solution from the equilibrium pool. Then, the IMEO2 can achieve the optimal solution while avoiding stuck in local optima using the exponential term (F) and generation rate (G) [17]. To balance the number of selected features in each

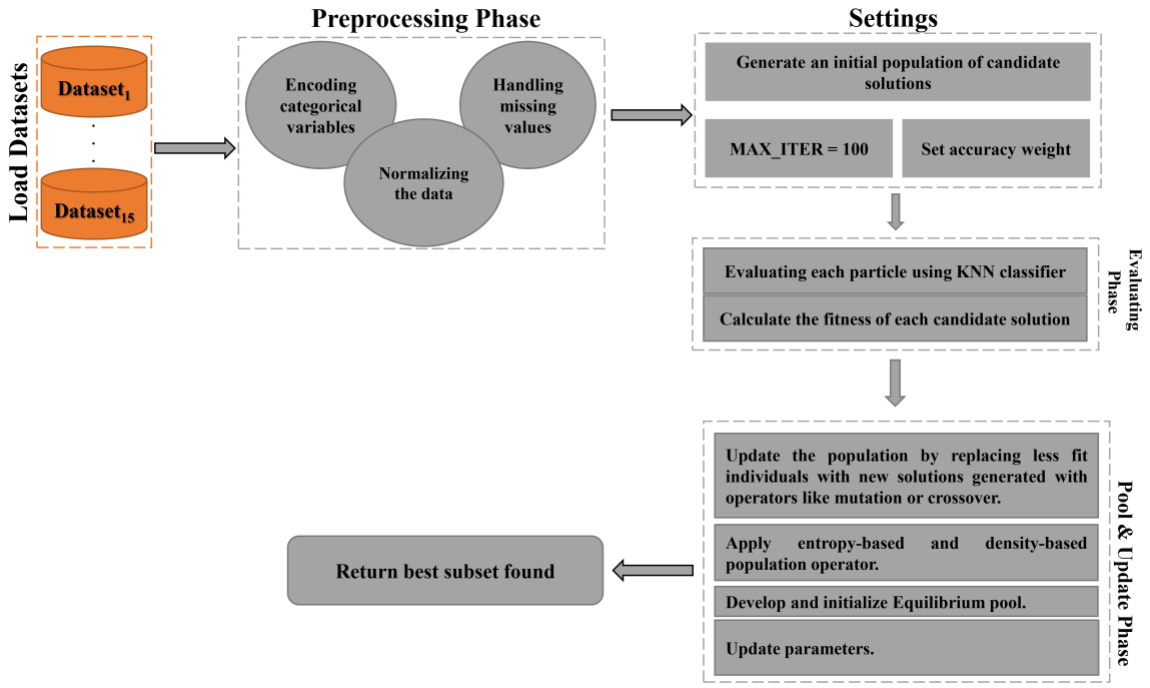

Fig. 4. Proposed feature selection framework

solution and the classification accuracy, the fitness function in Eq. (21) is used in the IMEO2 algorithm to estimate search agent's performance.

$$Fitness = \alpha\gamma_R + \beta\frac{|R|}{|N|} \tag{21}$$

The definition of each parameter in Eq. (21) is listed here:

$\alpha \in [0,1]$: constant to find a balance between classification and the number of reduced features.

: Classification error rate of given classifier (KNN) for subset R

$\beta = (1-\alpha)$: Another constant which calculated based on

R: Length of reduction features

N: Total number of features in the dataset

A fitness function is used to estimate whether a particular solution is a good one compared to other solutions. A classification process is efficient if the classification accuracy is high. In experiments, classification accuracy is crucial. Also, a wrapper approach based on the KNN classifier (where $K= 5$ [18]) is used to generate the best reduction.

6- Experimental Setup and Results

This section evaluates the performance of the proposed method using two series of experiments. The first experiment uses 14 benchmark functions to investigate the efficiency of the proposed algorithm (IMEO2). The second one evaluates the proposed feature selection method using six feature selection algorithms.

6- 1- Experimental setup

The proposed method has been implemented in Google's Colab research environment, and all methods that have been compared to our method have been implemented and tested in the same environment. Python programming language and Py-FS by [18] have also been used to implement the code. Furthermore, each algorithm is executed 5 times, and also available RAM in Google's Colab environment is 12 GB. (https://colab.research.google.com)

There are two categories of datasets used, small and medium. In order to test the method's classification accuracy performance, we used 15 well-known datasets, the information of which can be found in Table 2. Table 3 shows the parameters used in the IMEO2 algorithm.

Figure 5 shows the 15 datasets used along with the characteristics of each, i.e. the number of features and the number of samples in it.

In our experiment, n and *population_size* are set according to [19]. According to [18], features are selected based on accuracy weight. Table 3 shows the values of the provided method settings.

Table 2. Information of benchmark datasets.

No.	Datasets	Number of features	Number of instances
1	Iris	5	150
2	Arrhythmia	279	452
3	Ionsphere	35	351
4	Breastcancer	11	699
5	BreastEW	31	568
6	HeartEw	14	270
7	Exactly	14	1000
8	Exactly2	14	1000
9	Tic-tac-toe	10	958
10	M-of-n	14	1000
11	Digits	65	1797
12	Madelon	501	2600
13	Monk1	7	556
14	Wine	14	178
15	Zoo	17	101

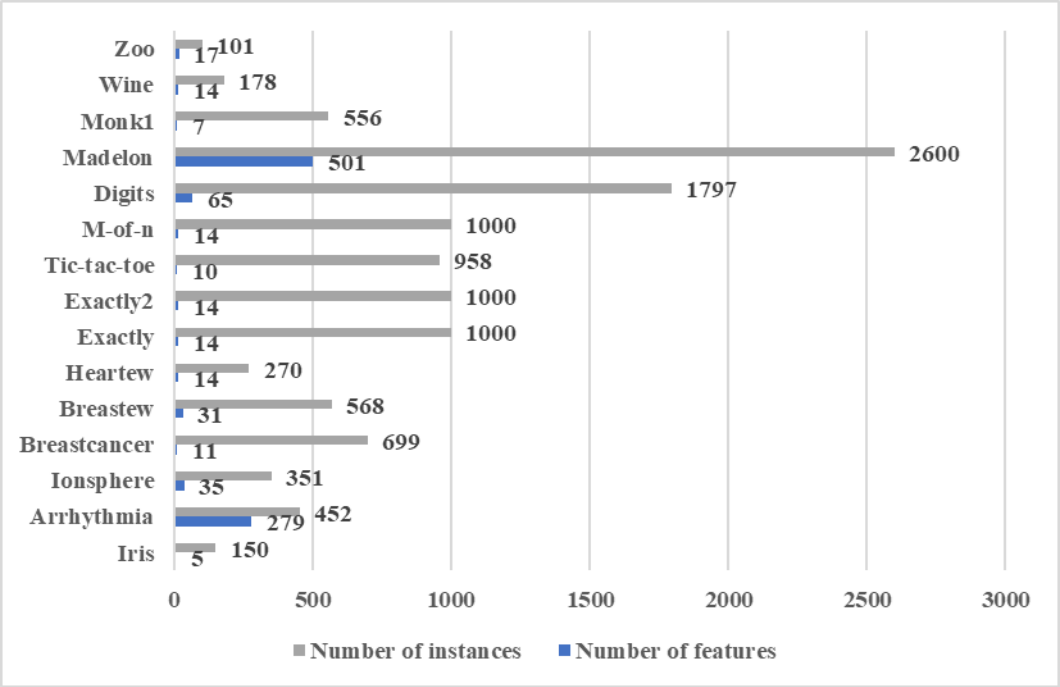

Fig. 5. Overview of the datasets used.

Table 3. Parameter setting of the proposes method

Parameter	Value
Maximum number of Iterations (n)	100
Agents (population size)	10
Number of neighbors in the KNN classifier	5
Number of runs	5
a_1	2
a_2	1
r	∈ [0, 1]
GP	0.5
pool size	4

Table 4. Unimodal benchmark functions.

Function	Dim	Range	f_{min}
$F_1(x)=\sum_{i=1}^{n}x_i^2$	30	[-100, 100]	0
$F_2(x)=\sum_{i=1}^{n}\left(\sum_{j-1}^{i}x_j\right)^2$	30	[-100,100]	0
$F_3(x)=\max_i\left\{\lvert x_i\rvert,1\le i\le n\right\}$	30	[-100, 100]	0
$F_4(x)=\sum_{i=1}^{n-1}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right]$	30	[-30,30]	0
$F_5(x)=\sum_{i=1}^{n}\left(\left[x_i+0.5\right]\right)^2$	30	[-100,100]	0
$F_6(x)=\sum_{i=1}^{n}ix_i^4+random\left[0,1\right)$	30	[-1.28,1.28]	0

6- 2- Experiment 1: Solving global optimization problems

We compare the performance of the proposed algorithm (IMEO2) with the original EO and IMEO [16] algorithms on 14 benchmark test functions. There are three types of functions: Unimolal (F1–F6), Multimodal (F7–F10), and Fixed-Dimensional (F11–F14). Tables 4-6 summarize these benchmark functions [20, 21].

Figure 6 shows the box plot for different benchmark functions for related methods. Results obtained by running each method 30 times. The proposed IMEO2 has the best performance in F2, F5, and F6 and improves the result by 5%, 10%, and 3.2% over the original EO. According to the broad box of IMEO in F2, F4, and F6, this algorithm cannot produce reliable results. On F7, F8, F9, and F14, the proposed IMEO2 has better performance than IMEO and improved the result by 14.6%, 12.3%, 3.1%, and 7.74%, while on F6 and F13, it matched the original EO's result. A density-based population operator is one of the reasons for this superiority, which considers the situation of all agents along with the situation of each agent from the previous iteration. If we consider only

Table 5. Multimodal benchmark functions.

Function	Dim	Range	f_{min}
$F_7(x)=\sum_{i=1}^{n}-x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	30	[-500,500]	−418.9829 × Dim
$F_8(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos\left(2\pi x_i\right)+10\right]$	30	[-5.12,5.12]	0
$F_9(x)=0.1\left\{\sin^2\left(3\pi x_1\right)+\sum_{i=1}^{n}\left(x_i-1\right)^2\left[1+\sin^2\left(3\pi x_i+1\right)\right]+\left(x_n\right.\right.$ $\sum_{i=1}^{n}u\left(x_i,5,100,4\right)$	30	[-50,50]	0

Table 6. Multimodal benchmark functions with fixed dimension.

Function	Dim	Range	f_{min}
$F_{10}(x)=\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}\left(x_i-a_{ij}\right)^6}\right)^{-1}$	2	[-65,65]	1
$F_{11}(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_i\left(b_i^2+b_i x_2\right)}{b_i^2+b_i x_3+x_4}\right]^2$	4	[-5,5]	0.00030
$F_{12}=\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos x_1+10$	2	[-5,5]	0.398
$F_{13}(x)=-\sum_{i=1}^{4}c_i \exp\left(-\sum_{j=1}^{6}a_{ij}\left(x_j-p_{ij}\right)^2\right)$	6	[0,1]	-3.32
$F_{14}(x)=-\sum_{i=1}^{7}\left[\left(X-a_i\right)\left(X-a_i\right)^T+c_i\right]^{-1}$	4	[0,10]	-10.4028

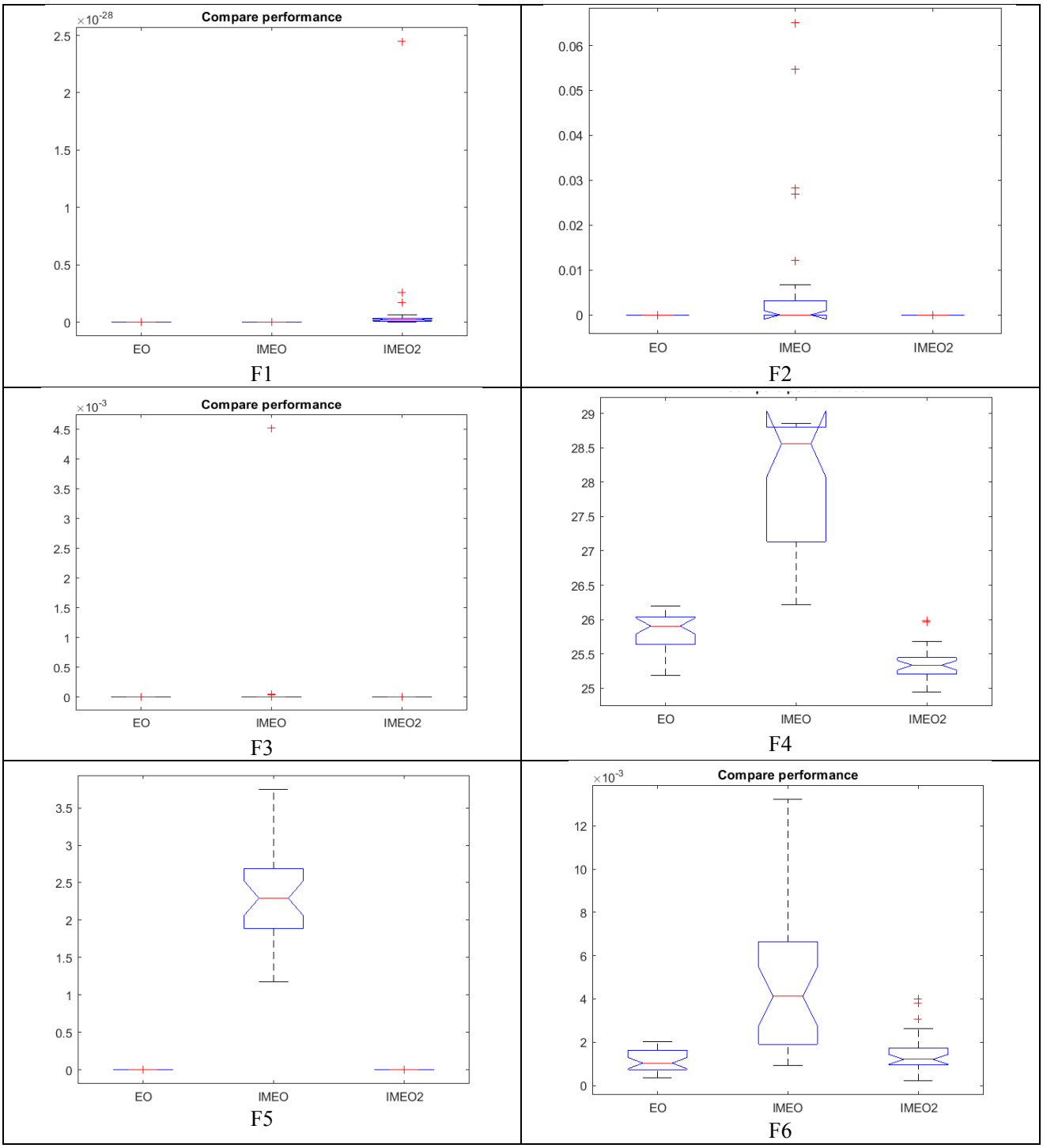

Fig. 6. Best fitness values for various benchmark functions.(Continued)

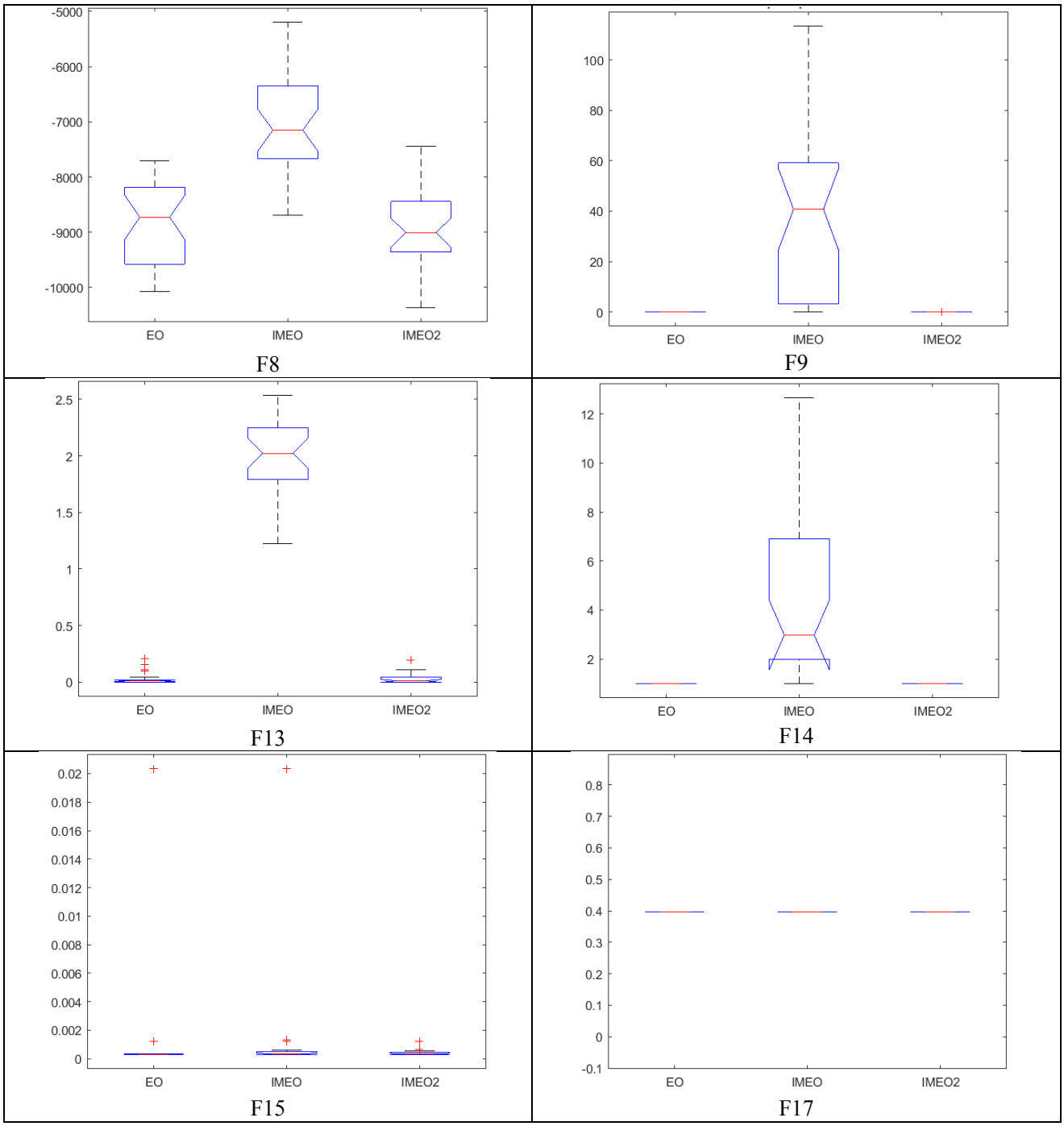

Fig. 6. Best fitness values for various benchmark functions.(Continued)

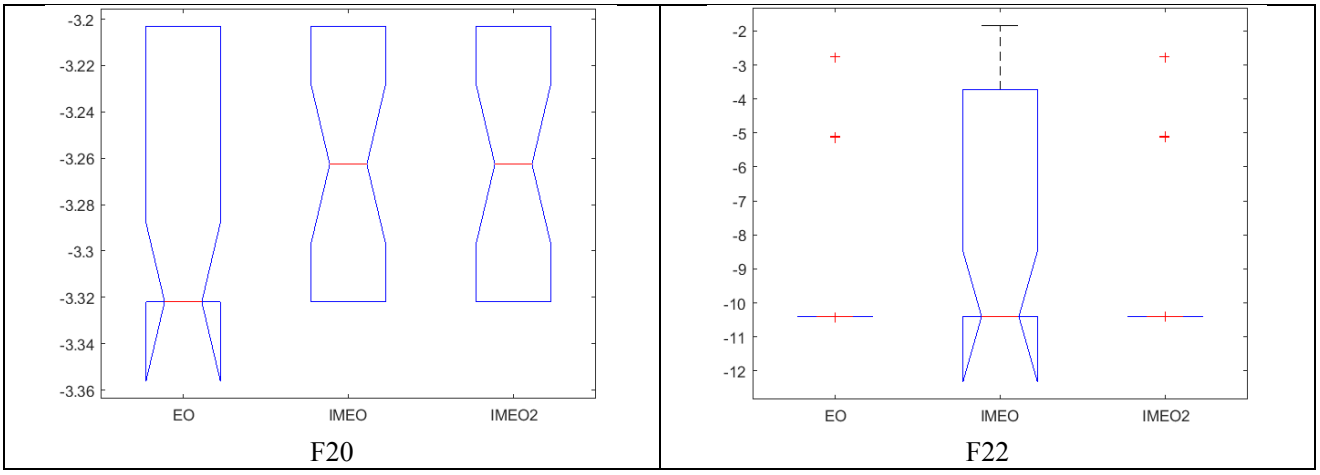

Fig. 6. Best fitness values for various benchmark functions.

Table 7. Results of comparing based on the classification accuracy.

No.	Dataset	IMEO2	EO	GA	GSA	CS	BBA	RDA
1	Iris	98.13	94.3332	96.6	100	100	100	100
2	Arrhythmia	79.2592	75.1851	58.2417	62.6373	63.7362	63.7362	61.5384
3	Ionsphere	90.285	89.9999	90.1408	90.1408	91.5492	84.5070	94.3661
4	Breastcancer	97.129	92.9999	93.5714	93.5714	92.1428	98.57201	96.4285
5	BreastEW	94.832	93.1578	94.7368	94.7368	95.614	92.9824	93.8596
6	HeartEw	74.8147	74.4444	81.4814	81.4814	75.9259	74.0740	77.7777
7	Exactly	99.7	91.8	69	69	69	69	69
8	Exactly2	64.8	62.9	76	76	76	76	76
9	Tic-tac-toe	79.9999	75.6249	65.1041	65.625	65.1041	73.9583	77.0833
10	M-of-n	100	99.99	76.5	79	87	73.5	81
11	Digits	90.337	81.337	95.5	94.4	95	93.3333	93.8888
12	Madelon	57.661	53.9999	73.2692	80.9615	70.5769	68.6538	62.8846
13	Monk1	99.818	99.818	72.3214	87.5	100	81.25	100
14	Wine	88.1	84.4444	94.4	94.4	97.2	94.4444	97.2222
15	Zoo	90.514	72	95.2380	95.2380	95.2380	95.2380	95.2380

the position of one agent at a point in search space from a previous iteration, it may be the best agent at that point and all the other agents have moved toward the local optimal point. In this case, it may not be necessary for the agents to move toward the best choice if the density is calculated at that point. In contrast, if local optima are dense, the density-based population operator enables agents to escape from them.

6- 3- Experiment 2: Feature selection using benchmark datasets

All results from experiments are reported in this section. We compare the IMEO2 method in terms of classification accuracy with other six state-of-the-art methods.

From Table 7, we see the efficiency of the IMEO2 algorithm over the EO, GA, GSA, CS, BBA, and RDA

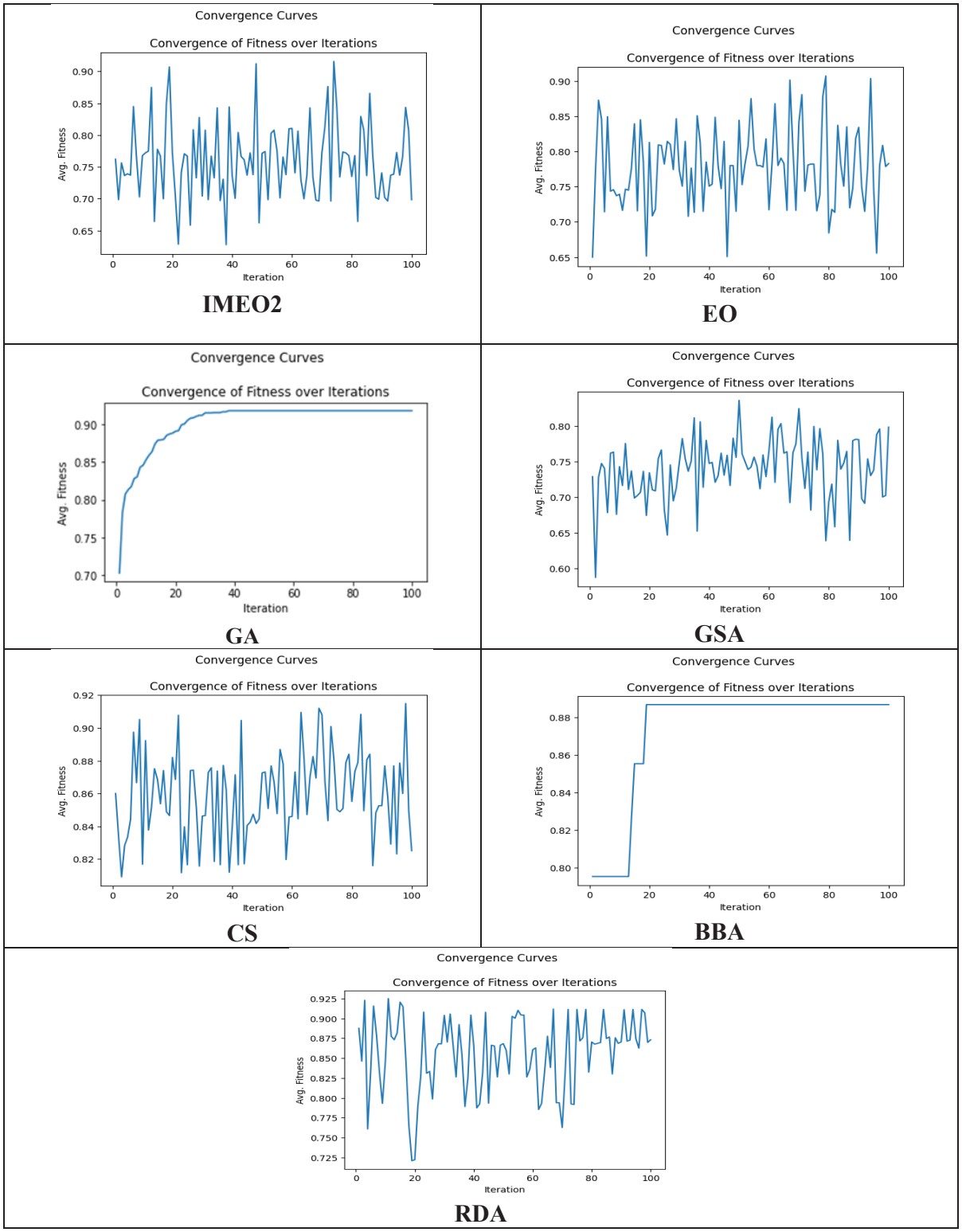

Fig. 7. Convergence curves of IMEO2, EO, GA, GSA, CS, BBA, and RDA (tested on BreastCancer dataset).

algorithms. Also, it can be seen that the IMEO2 algorithm has good accuracy on the Iris (98.13%, 94.3332%, 96.6%, 100%, 100%, 100%, 100%), Arrhythmia (79.2592%, 75.1851%, 64.8352%, 58.2417%, 62.6373%, 63.7362%, 63.7362%, 61.5384%), BreastCancer (97.129%, 92.9999%, 93.5714%, 93.5714%, 92.1428%, 98.5720%, 96.4285%), M-of-n (100%, 99.99%, 76.5%, 79%, 87%, 73.5%, 81%), and Monk1 (99.818%, 99.818%, 72.3214%, 87.5%, 100%, 81.25%, 100%) datasets respectively.

Figure 7 and Fig. 8 indicate the convergence curves of

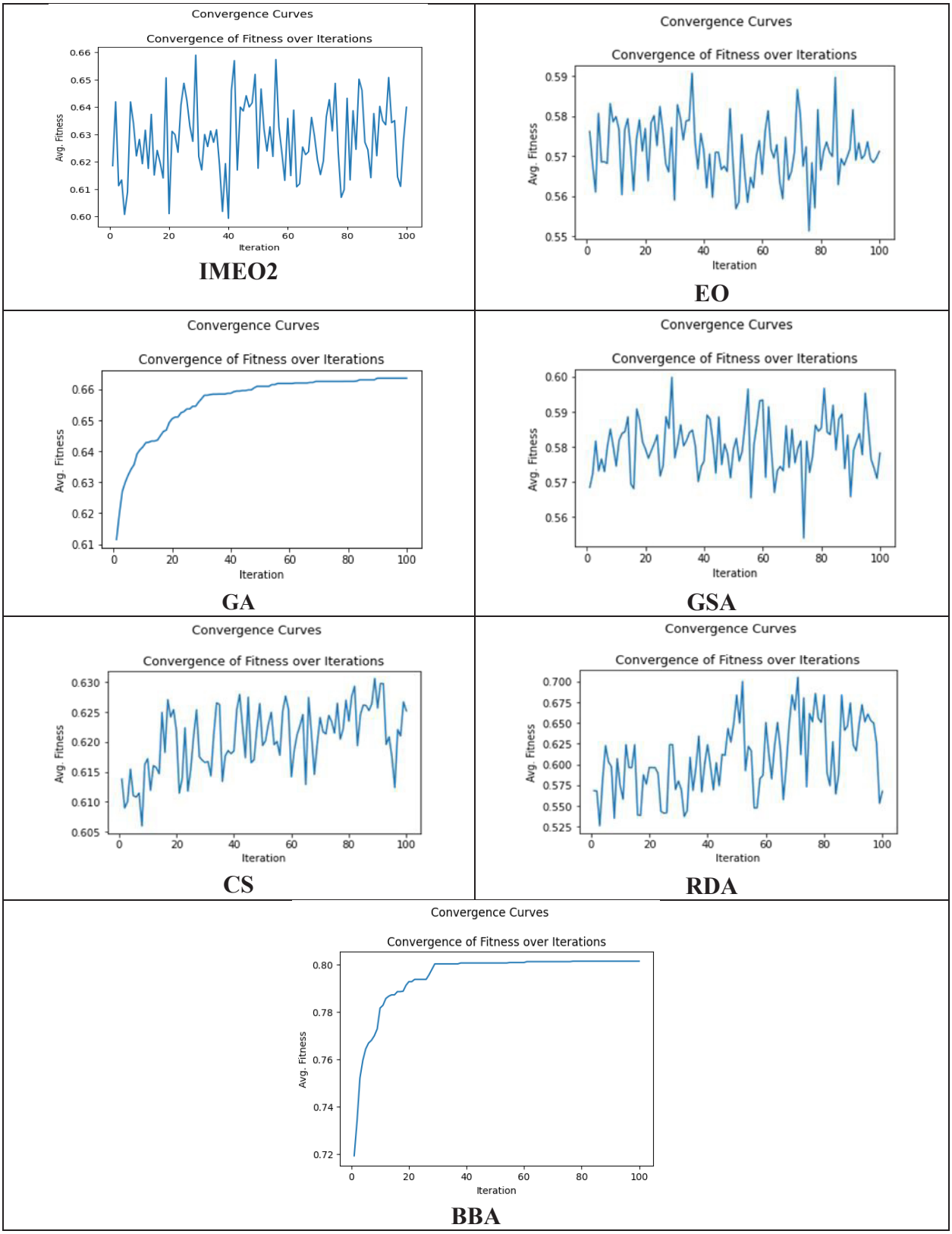

Fig. 8. Convergence curves of IMEO2, EO, GA, GSA, CS, RDA, and BBA (tested on Madelon dataset).

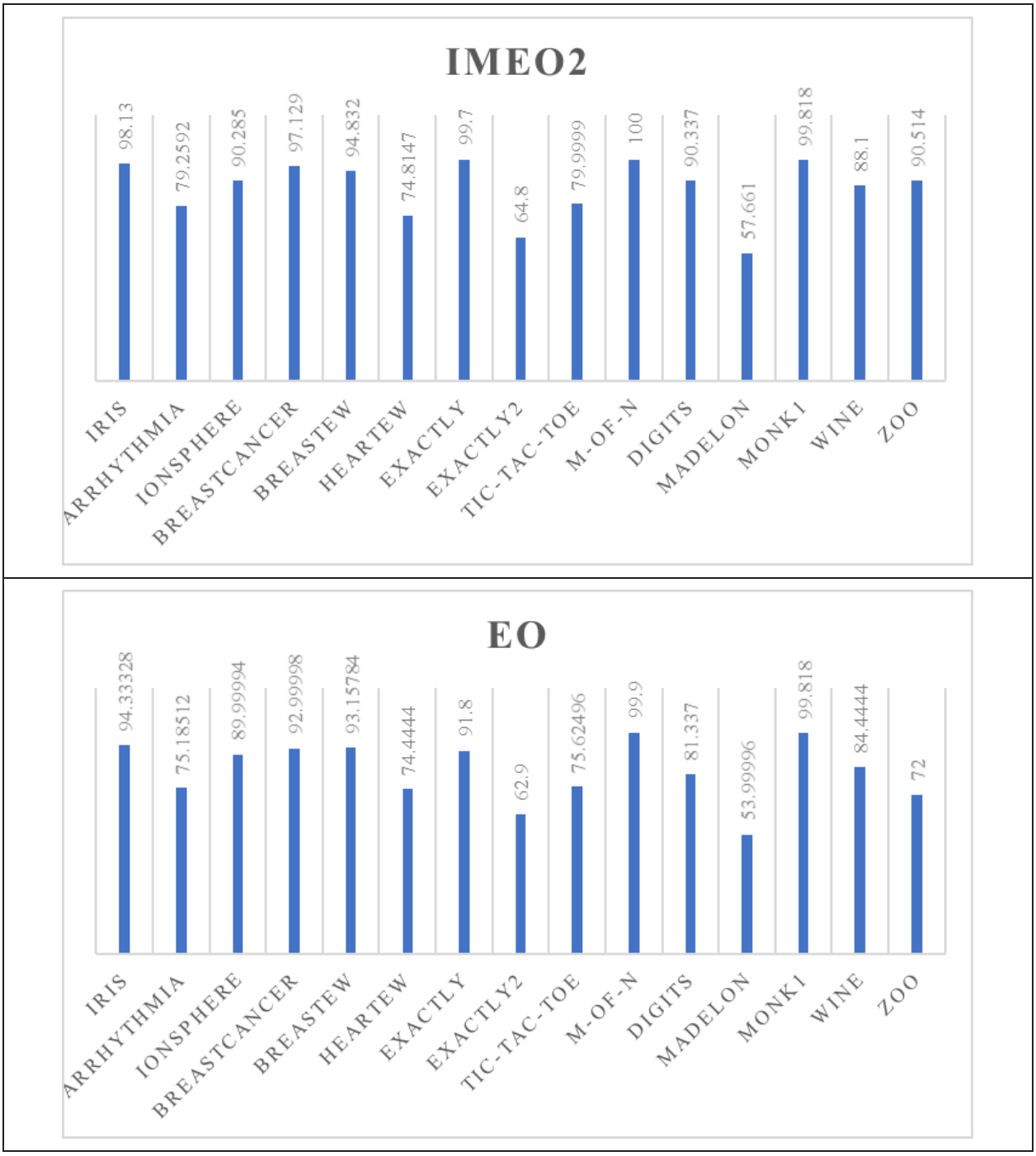

Fig. 9. Diagrams of IMEO2, and other compared algorithms based on classification accuracy.(Continued)

IMEO2, EO, GA, GSA, CS, RDA, and BBA for BreastCancer and Madelon, respectively. Figure 9 shows the classification accuracy test results of the IMEO2, EO, GA, GSA, CS, RDA, and BBA algorithms on the 15 datasets tested. According to Fig. 9, the IMEO2 algorithm performed better on the BreastCancer dataset against EO, GA, GSA, and on the Monk1 dataset against GA, GSA, and BBA. Also, the overall performance of the proposed IMEO2 is superior to all compared methods on Arrhythmia, Exactly, Tic-tac-toe, and M-of-n datasets. Also, according to Fig. 9, the genetic algorithm was less accurate in datasets with medium-dimension. This performance discrepancy is due to a genetic algorithm defect in population loss due to the rare occurrence of premature convergence [22]. The CS algorithm also has low classification accuracy [22], and we can see that it performs worse than the IMEO2 in the Madelon, Digits, and

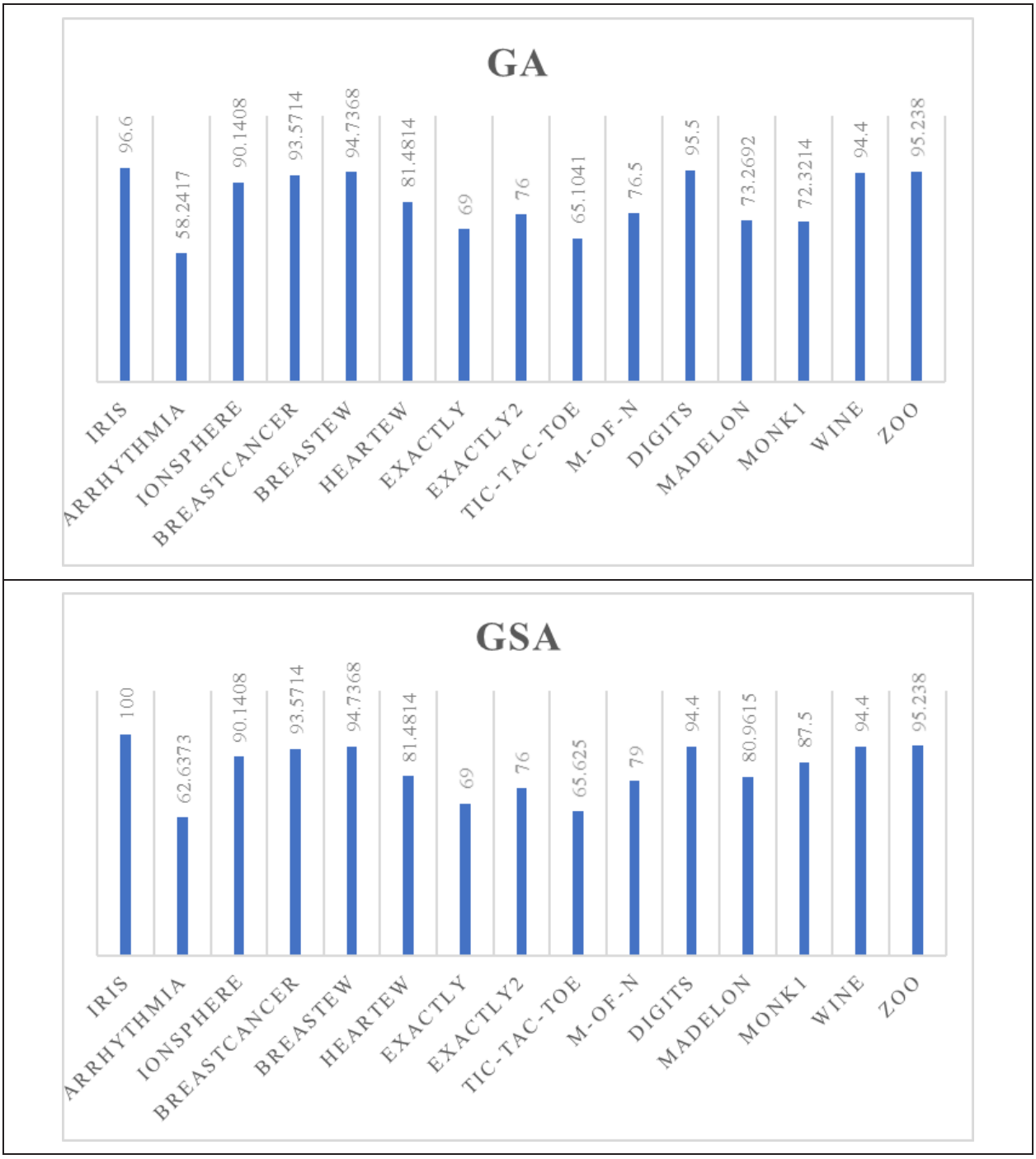

Fig. 9. Diagrams of IMEO2, and other compared algorithms based on classification accuracy.(Continued)

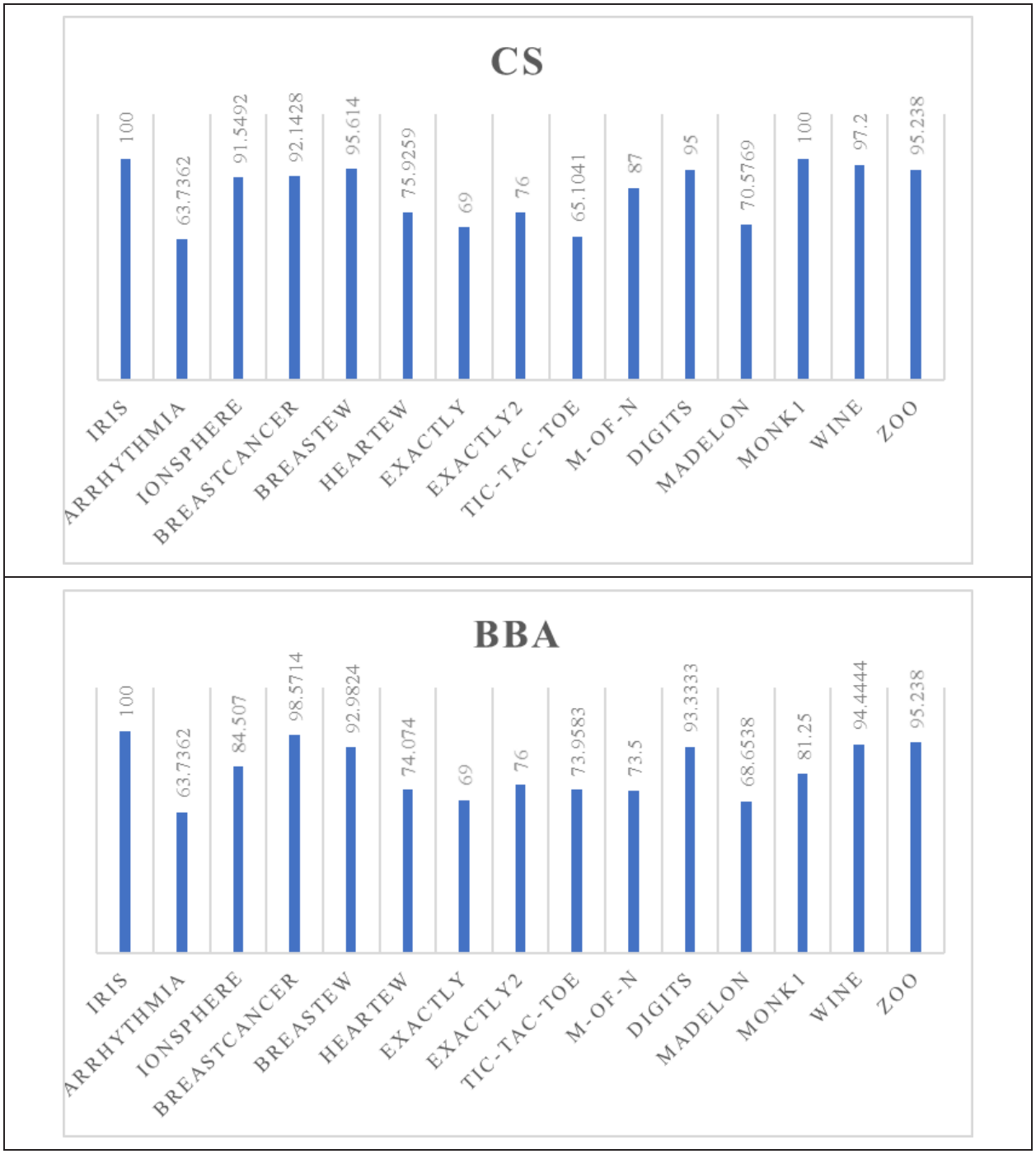

Fig. 9. Diagrams of IMEO2, and other compared algorithms based on classification accuracy.(Continued)

Arrhythmia datasets. Since IMEO2 has suitable exploratory and exploitative search processes and randomly modifies the solutions.

Figure 10 shows the average classification accuracy of IMEO2, EO, GA, GSA, CS, RDA, and BBA algorithms over 15 datasets. As we can see the proposed algorithm IMEO2, outperforms the other 4 algorithms for accuracy rate in the form of IMEO2 (87.0253%), GA (82.1403%), GSA (84.3128%), BBA (82.6165%), and EO (82.7963%).

In this experiment, we find that the IMEO2 algorithm is superior to the mentioned algorithms and has been able to have good results in terms of classification accuracy in the feature selection field.

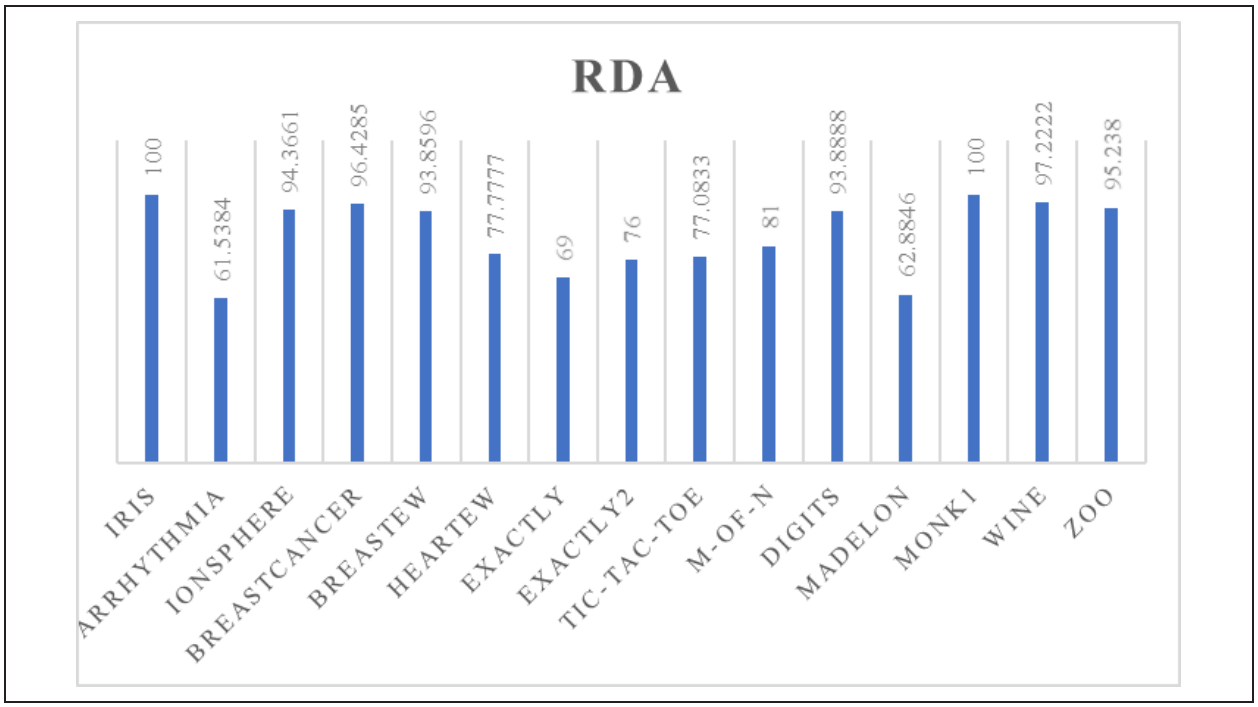

Fig. 9. Diagrams of IMEO2, and other compared algorithms based on classification accuracy.

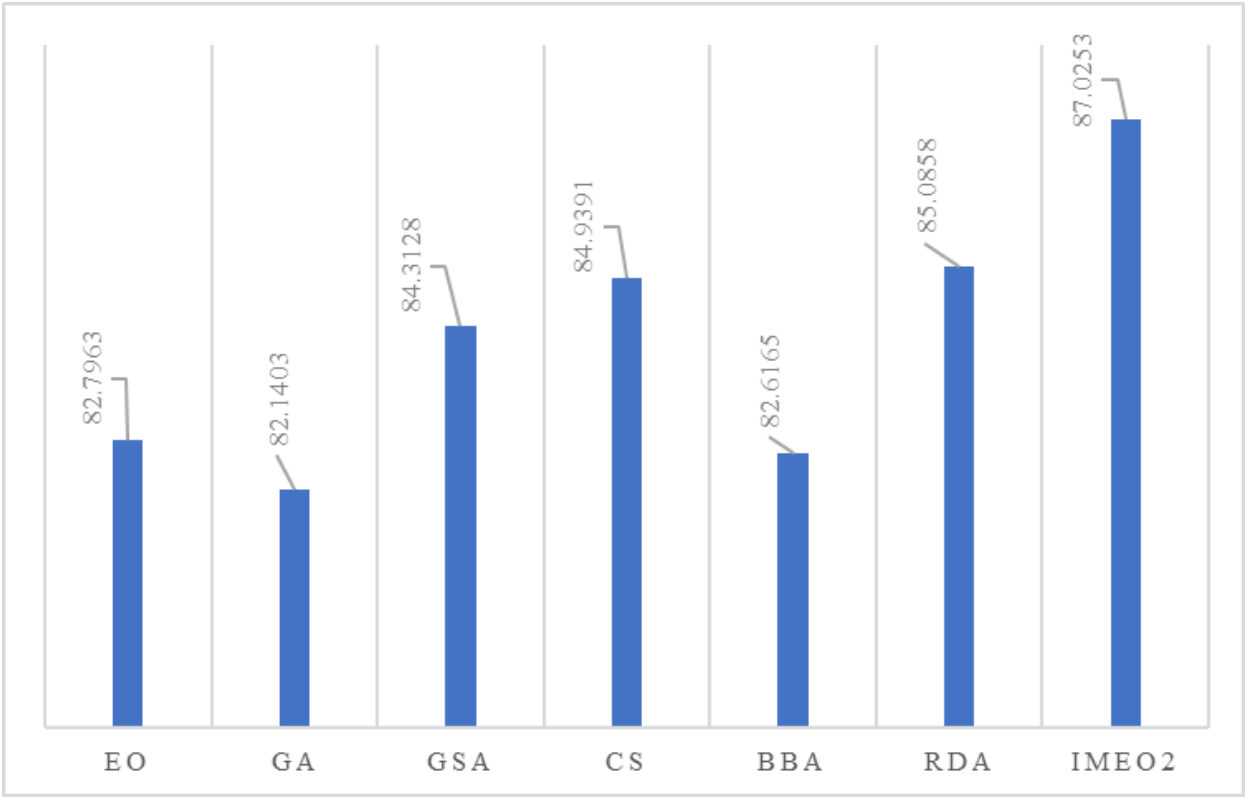

Fig. 10. Average classification accuracy of the IMEO2, GA, GSA, CS, BBA, RDA, and EO over 15 datasets.

7- Conclusion and Future works

In order to improve the problem-solving process, evolutionary algorithms were used because traditional algorithms were inefficient. Equilibrium Optimizer (EO) is a physics-based model that makes educated guesses about equilibrium states using dynamic source and sink models. This paper proposes an improved EO that enhances the capability to explore divergent areas based on density and entropy operators. The improved EO is used to solve the FS problem using wrapper methods for classification purposes. We perform two experimental series to investigate the performance of the improved EO method using ten benchmark functions and fifteen datasets with different properties. Results of experimental series 1 show that the improved EO algorithm provides superior results on most tests, especially when solving multimodal functions. Experimental series 2 also shows that the improved EO is more accurate than other comparative algorithms (i.e., EO, GA, GSA, CS, RDA, and BBA) at selecting the optimal subset of features. For future work, the EO method can be tested on different types of datasets. In addition, the efficiency of the proposed approach should be evaluated with other classifiers (e.g., artificial neural networks). Finally, we aim to apply other evolutionary methods to deal with feature selection and parameter tuning.

References

[1] A. Faramarzi, M. Heidarinejad, B. Stephens, S. Mirjalili, Equilibrium optimizer: a novel optimization algorithm, Knowledge-Based Systems, vol. 191, 2020, doi: 10.1016/j.knosys.2019.105190

[2] J. Chen, S. Yuan, D. Lv, Y. Xiang, A novel self-learning feature selection approach based on feature attributions, Expert Systems with Applications, vol. 183, 2021, doi: 10.1016/j.eswa.2021.115219

[3] F.A. Hashim, E.H. Houssein, K. Hussain, M.S. Mabrouk, W. Al-Atabany, Honey badger algorithm: new metaheuristic algorithm for solving optimization problems, Mathematics and Computers in Simulation, vol. 192, pp. 84-110, 2022, doi: 10.1016/j.matcom.2021.08.013

[4] F.A. Hashim, K. Hussain, E.H. Houssein, M.S. Mabrouk, W. Al-Atabany, Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems, Applied Intelligence, vol. 51, pp. 1531–1551, 2021, doi: 10.1007/s10489-020-01893-z

[5] B. Abdollahzadeh, F. Soleimanian Gharehchopogh, S. Mirjalili, African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems, Computers & Industrial Engineering, vol. 158, 2021, doi: 10.1016/j.cie.2021.107408

[6] L.P. Jhansi, P.T. Maruthi, Identifying optimal features using whale optimization technique, 2nd International Conference for Emerging Technology, 2021, doi: 10.1109/INCET51464.2021.9456188

[7] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Computers & Electrical Engineering, vol. 40, pp. 16-28, 2014, doi: 10.1016/j.compeleceng.2013.11.024

[8] S. Nagpal, S. Arora, S. Dey, Shreya, Feature selection using gravitational search algorithm for biomedical data, Procedia Computer Science, vol. 115, pp. 258-265, 2017, doi: 10.1016/j.procs.2017.09.133

[9] E. Emary, W. Yamany, A. E. Hassanien, New approach for feature selection based on rough set and bat algorithm, 9th International Conference on Computer Engineering & Systems (ICCES), pp. 346-353, 2014, doi: 10.1109/ICCES.2014.7030984

[10] Z.M. Elgamal, N.M. Yasin, A.Q.M. Sabri, R. Sihwail, M. Tubishat, H. Jarrah, Improved equilibrium optimization algorithm using elite opposition-based learning and new local search strategy for feature selection in medical datasets, Computation, vol. 9, 2021, doi: 10.3390/computation9060068

[11] N. Maleki, Y. Zeinali, S.T.A. Niaki, A k-NN method for lung cancer prognosis with the use of a genetic algorithm for feature selection, Expert Systems with Applications, Volume 164, 2021, https://doi.org/10.1016/j.eswa.2020.113981.

[12] C.L. Huang, C. J. Wang, A GA-based feature selection and parameters optimization for support vector machines, Expert Systems with Applications, Volume 31, Issue 2, 2006, pp. 231-240, https://doi.org/10.1016/j.eswa.2005.09.024.

[13] S.S. Alotaibi, Ensemble technique with optimal feature selection for saudi stock market prediction: a novel hybrid red deer-grey algorithm, IEEE Access, vol. 9, pp. 64929-64944, 2021, doi: 10.1109/ACCESS.2021.3073507

[14] M. Tarade, M. Mafarja and A.A. Heidari, H. Faris, I. Aljarah, S. Mirjalili, H. Fujita, An evolutionary gravitational search-based feature selection, Information Sciences, vol. 497, pp. 219-239, 2019, doi: 10.1016/j.ins.2019.05.038

[15] G. Tikhe, T. Joshi, A. Lahorkar, A. Sane, J. Valadi, Feature selection using equilibrium optimizer, Data Engineering and Intelligent Computing, Springer, pp. 307-315, 2021.

[16] Zahra Asghari Varzaneh, Soodeh Hossein, Sepehr Ebrahimi Mood, Mohammad Masoud Javidi, A new hybrid feature selection based on Improved Equilibrium Optimization, Chemometrics and Intelligent Laboratory Systems, Volume 228, 2022, https://doi.org/10.1016/j.chemolab.2022.104618.

[17] Y. Gao, Y. Zhou, Q. Luo, An efficient binary equilibrium optimizer algorithm for feature selection, IEEE Access, vol. 8, pp. 140936-140963, 2020, doi: 10.1109/ACCESS.2020.3013617

[18] R. Guha, B. Chatterjee, K.H. Sk, S. Ahmed, T. Bhattacharya, R. Sarkar, Py_FS: a python package for feature selection using meta-heuristic optimization algorithms. In: Springer AISC series of 3rd International Conference on Computational Intelligence in Pattern

Recognition (CIPR-2021), Kolkata, India, 2021.

[19] J. Too S. Mirjalili, General learning equilibrium optimizer: a new feature selection method for biological data classification, Applied Artificial Intelligence, 2020, doi: 10.1080/08839514.2020.1861407

[20] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Transactions on Evolutionary Computation, vol. 3, pp. 82-102, 1999

[21] M. Molga, C. Smutnicki, Test functions for optimization needs, Test Functions for Optimization Needs, 2005

[22] A.E. Ezugwu, O.J. Adeleke, A.A. Akinyelu, S.Viriri, A conceptual comparison of several metaheuristic algorithms on continuous optimisation problems, Neural Computing and Applications, vol. 32, pp. 6207–6251, 2020, doi: 10.1007/s00521-019-04132-w

HOW TO CITE THIS ARTICLE

M. Amiri Ebrahimabadi, B. Mohammad Hasani Zade, N. Mansouri, Improved Equilibrium Optimizer using Density-based Population and Entropy Operator for Feature Selection, AUT J. Model. Simul., 55(1) (2023) 17-38.

DOI: 10.22060/miscj.2023.21474.5288