IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 23, 2021, accepted August 21, 2021, date of publication August 26, 2021, date of current version September 7, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3108097

An Improved Equilibrium Optimizer Algorithm for Features Selection: Methods and Analysis

DINA A. ELMANAKHLY[1], **MOHAMED MOSTAFA SALEH**[2], **AND ESSAM A. RASHED**[1,3], **(Senior Member, IEEE)**

[1]Department of Mathematics, Faculty of Science, Suez Canal University, Ismailia 41522, Egypt
[2]Faculty of Computers and Informatics, Cairo University, Giza 12613, Egypt
[3]Department of Electrical and Mechanical Engineering, Nagoya Institute of Technology, Nagoya 466-8555, Japan

Corresponding author: Dina A. Elmanakhly (dina_almnakhly@science.suez.edu.eg)

ABSTRACT In the last decade, data generated from different digital devices has posed a remarkable challenge for data representation and analysis. Because of the high-dimensional datasets and the rapid growth of data volume, a lot of challenges have been encountered in various fields such as data mining and data science. Conventional machine learning classifiers are of limited ability to handle the problems of high dimensionality that includes memory limitation, computational cost, and low accuracy performance. Consequently, there is a need to reduce the dimension of datasets by choosing the most significant features that would represent the data efficiently with minimum volume. This study proposes an improved binary version of the equilibrium optimizer algorithm (IBEO) to mitigate features selection problem. Two main enhancements are added to the original equilibrium optimizer (EO) to strengthen its performance. Opposition based learning is the first advancement added to the initialization stage of EO to enhance the diversity of the population in the search space. Local search algorithm is the second advancement added to enhance the exploitation of EO. Wrapper approaches can offer premium solutions. Thus, we used k-nearest neighbour classifier and support vector machine classifiers as the most popular wrapper methods. Moreover, dealing with the problem of over-fitting is an essential task that urges on applying k-fold cross-validation to split each dataset into training and testing data. Comparative tests with different well-known algorithms such as grey wolf optimization, grasshopper optimization, particle swarm optimization, whale optimization, dragonfly, and improved salp swarm algorithms are considered. The proposed algorithm is applied to the most commonly datasets used in the field to validate the performance. Statistical analysis studies demonstrate the effectiveness of the IBEO.

INDEX TERMS Equilibrium optimizer (EO), feature selection, optimization, machine learning (ML), opposition based learning (OBL), classification.

I. INTRODUCTION

The technological evolution in many fields such as finance, biomedical, bioinformatics, and telecommunication has produced an exponential volume of pervasive data. The rapid growth in the data volume has fabricated datasets with thousands of features (also known as attributes) comprising diverse data, resulting in a lot of challenges and fundamental tasks in data science applications. Typically, high dimensional datasets are accompanied by redundant, irrelevant, and noisy records, which hurts the accuracy of machine learning (ML) classification and raises computational costs [1]. Conventional ML classifiers are incapable of dealing with such a large number of features, and are frequently trapped in local optima [2]. Accordingly, a preprocessing procedure like features selection becomes a necessity to address the high dimensional data problem and filter out the unnecessary/redundant features. Features selection in ML and statistics is also referred to as attribute selection, variable selection, or variable subset selection. It is a part of data preprocessing which is supposed to be the most time-consuming process of any ML pipeline. Features selection process can help build robust models by removing redundant and irrelevant features and choosing the most informative ones.

The associate editor coordinating the review of this manuscript and approving it for publication was Hiu Yung Wong.

Features selection methods are mainly categorized into wrapper methods, and filter methods. The feature subset evaluation step in wrapper approaches is based on the performance of the classification algorithm. A wrapper employs the classification algorithm as a ''black box'' to evaluate the quality of the chosen subset via the classification performance [3]. A filter approach is not based on any learning model and the features are picked and ranked in principal based on statistical approaches [4]. Filter algorithms are frequently more general and less computationally expensive than wrapper methods [5]. Filters, on the other hand, ignore the classification algorithm's performance of the selected features, whereas wrappers are highly dependent on the used learning algorithm to evaluate the feature subsets. In terms of classification accuracy, wrappers beat filter methods [6]. Wrapper methods in many classification problems do not significantly rely on a large number of selected attributes to attain high classification accuracy. Wrappers are more accurate because they take into account the relationships between their features [7]. Moreover, this approach can use various ML methods and knowledge extraction such as k nearest neighbor (kNN) [8], discriminant analysis [9], artificial neural networks (ANN) [10], and support vector machines (SVMs) [11]. Some researchers divide features selection approaches into three categories: wrapper, filter, and embedded approaches [5]. Embedded techniques are methods that combine feature selection and classifier learning into a single process [12].

The interactions and relationships among the features, as well as the extensive search space, have made features selection one of the most difficult data mining and classification processes. Features interaction can be two-way, three-way, or even include several features. A feature may not have a confident impact on the target when it is used individually, however its effect can be amplified when combined with other features. In addition, a feature that is useful on its own can become redundant when combined with others. The extensive search space 2^n, n is the total number of features, is another challenging process. In other words, exhaustively searching for all possible solutions is not possible in most situations. As a result, a variety of search algorithms have been proposed in order to locate a sufficiently good subset.

According to A. Jovic [13], there are three types of search techniques: exponential, sequential, and random selection strategy. The number of evaluated features in the exponential methods increases exponentially with the size of features. Although this approach produces reliable results, it is impractical to use due to its high computational cost. Some exponential algorithms are exhaustive search, bound and branch methods [14]. Sequential methods add or delete features sequentially. The main problem that leads to local optima is that it is not possible to make modifications once a feature has been added or deleted from the selected subset. Sequential forward selection (SFS) is a sequential search strategy that works best when the optimal subset has a limited number of features. The key drawback of SFS is that it cannot be used to remove features that become outdated when new ones are added [15]. Sequential backward elimination (SBE) is another example of sequential algorithms but it works in the opposite direction of SFS. When the feature subset has a large number of features, SBE performs best. SBE's key flaw is its inability to reevaluate the utility of a feature after it has been removed [16]. With some backtracking capabilities, plus-L minus-R selection (LRS) tries to compensate for SFS and SBE's flaws. The lack of theory to estimate the optimum values of L and R is its primary weakness [17]. To overcome this limitation, sequential forward floating selection (SFFS) and sequential backward floating selection (SBFS) have been proposed [18]. These approaches have been shown to be superior to static sequential approaches. The common issues with most approaches are premature convergence, high computing cost, and enormous complexity. Random algorithms, also known as population-based techniques, use randomization in the search process to avoid trapping the algorithms in local optima. Metaheuristics are examples of random search strategy that draw researchers' attention to deal with these kinds of problems.

Wrapper approaches based on metaheuristic algorithms have shown their productivity and efficiency as an attempt to solve features selection problem. The idea behind solving features selection problems with metaheuristics is that it can deliver a solution that is closer to the optimal solution in a reasonable amount of time [19]. Metaheuristic algorithms have a stochastic behavior, as they begin their optimization process by producing random solutions to efficiently explore the search space. It is not necessary to compute the search space derivative. Because of their simple principle and easy implementation, metaheuristics are easily adaptable according to particular problem. The key feature of these algorithms is their extraordinary ability to avoid algorithms from converging prematurely. Metaheuristic algorithms maintain a balance between its two important aspects, exploration and exploitation [20].

Metaheuristic methods are mainly divided into four categories based on the source of inspiration: (i) swarm intelligence [21], (ii) human-based methods [22], (iii) physics-based methods [23], [24], and (iv) evolutionary algorithms [25]. Swarm intelligence methods are inspired by the way in which animals behave in swarms as individual information is shared throughout the optimization process. Particle swarm optimization (PSO), one of the most well-known contributions in this class, it has a number of advantages, including its simplicity and high convergence rate [26], [27]. To deal with the problem of picking the minimal subset of features for high-dimensional data, a modification strategy for the PSO method was presented in [28]. One of the promising approaches is the grey wolf optimizer (GWO), which is an efficient nature-inspired population-based metaheuristic algorithm [29]. Emary *et al.* [30] employed the sigmoid transfer function to get the first binary version of GWO (BGWO). The kNN classifier was used to calculate the classification accuracy and was applied to eighteen distinct UCI datasets.

Levy flight GWO was proposed by Pathak *et al.* [31], and it was utilized to pick significant features from the original datasets. The acquired findings demonstrated its excellent performance in terms of achieving high convergence. Al-Tashi *et al.* [32] utilized GWO algorithm to choose the best features with SVM as a classifier to diagnose cardiovascular disease. In addition, the author created a binary version (BMOGW-S) based on a sigmoidal function to solve multi-objective features selection problems with an artificial neural network for classification process [33]. On fifteen benchmark datasets, BMOGW-S outperformed MOGWO, which used the tanh transfer function, in both classification accuracy and feature selection. The grasshopper optimization algorithm (GOA), another new population-based optimizer, introduced by Saremi *et al.* to simulate the swarm behavior in natural populations of grasshopper insects [34]. A new transfer function was introduced by Hichem *et al.* [35]. The hamming distance was used to transform continuous variables into binary ones to suit the nature of the features selection problem. The latest version of GOA (NBGOA) was tested on 20 standard datasets and compared to previous GOA versions. The reported data demonstrated NBGOA's potential to deliver excellent results. Mafarja *et al.* used mutation operator sigmoid and V-shaped transfer functions to improve the exploration quality of BGOA [36]. To solve the binary optimization problems, Eid *et al.* [37] added transformation functions to the conventional whale optimization algorithm (WOA). The exploitation capability of WOA has been improved by Mafarja *et al.* [38], he integrated the simulated annealing into WOA in each iteration step to enhance the best solution. Sayed *et al.* extended the WOA algorithm with the chaotic search to avoid the slow convergence rate and fight local optima stagnation in feature selection problems [39]. Mirjalili *et al.* investigated dragonfly's behavior bringing out a new metaheuristic algorithm called dragonfly algorithm (DA) [40]. Medjahed *et al.* [41] suggested a comprehensive cancer diagnosis method based on the binary dragonfly (BDA) algorithm and SVM. SVM-RFE (SVM-recursive feature elimination) was used to select the gene from the datasets, with BDF being utilized to improve SVM-RFE performance. Six micro-array datasets were used to test the proposed technique, which yielded high accuracy results. Mafarja *et al.* developed a binary version of the dragonfly method that uses time-varying transfer functions to strike a good balance between exploration and exploitation [42].

Physics-based methods are derived from natural physics laws. The big bang-big crunch theory brought out an optimization method called the big bang-big crunch (BB-BC) algorithm [43]. In addition, a novel optimization method for developing charged system search (CSS) was inspired by the newtonian mechanical laws and the coulomb's law in electrostatics [24]. Another instances of this class: simulated annealing (SA) [44], multi-verse optimizer (MVO) [45], henry gas solubility optimization (HGSO) [46], lightning search algorithm (LSA) [47], chemical reaction optimization (CRO) [48], gravitational search algorithm (GSA) [49], and electromagnetic field optimization (EFO) [50]. Evolutionary methods are based on darwinian evolutionary theory and simulate natural evolution laws. Genetic algorithms (GA) are an instance of evolutionary methods [51]. They have a lot of potential for addressing complex optimization problems. Sayed *et al.* utilized a nested-GA algorithm for the features selection of cancer micro-array datasets. Nested-GA contains two GA algorithms: inner and outer GA algorithms and they work on two different kinds of datasets [52]. The results of nested-GA showed a small optimal subset of features with the highest classification performance. The authors in [53] used the GA feature selection algorithm with the chaos optimization for text categorization. Differential evolution algorithms (DE) are another instance of evolutionary methods [54].

Human behavior and human interaction in society inspire human-based methods. Agrawal [55] introduced the first novel binary version of gaining sharing knowledge based algorithm (GSK) for feature selection problem (FSNBGSK). The authors introduced binary junior and senior gaining and sharing stages. With the kNN classifier, the FS-NBGSK method was tested over 23 benchmark datasets from the UCI repository. In terms of classification accuracy and the smallest number of selected features, the proposed method outperformed the other compared algorithms. Representative algorithms of human based methods include: cultural evolution algorithm (CEA) [56], imperial competition algorithms (ICA) [57], teaching-learning-based optimization (TLBO) [22], and the volleyball premier league (VPL) [58]. Recently, interesting new methods have been developed, taking inspiration from different fields. For instance, paints have been used to propose a new optimization method as one of the most important fields of art [59]. Other new instances can be found in [60]–[63]. Finally, hybridization of multiple algorithms is a trendy method in features selection to take advantage of the strength of different algorithms [64]–[69].

Opposition-Based Learning (OBL) is a field of study that has been widely employed to enhance the search process in several algorithms [70], [71]. Tizhoosh *et al.* [72] was the first to use OBL to speed up the convergence rate and enhance the quality of solutions suggested by metaheuristics. Its primary principle is to think about a solution and its opposite at the same time. The logic behind this concept is that an opposing solution has a larger chance of being closer to the optimal solution than another random guess, given that the current original solution is not closer to the optimal solution [73]. Type-I opposition and type-II opposition are two types of search that use the OBL idea [74]. Type-I opposition is a function that maps each solution in the search space to its opposite, allowing the search to proceed with the better solution. Type-II opposition establishes a relationship between the original solution and its opposing based on evaluating their qualities. The most common OBL type in the literature is type-I opposition [74], which was employed in this study. Several opposition-based metaheuristics based on various OBL techniques have been proposed in the literature.

This is because the distance between the initial solutions and the unknown optimal solution is crucially related to the convergence speed of metaheuristics to the global optimum. If the optimal solution is far away from the initial solutions, the convergence speed to the global optimum could be slow. As a result, well-diversified initial solutions can speed up metaheuristic convergence [75]. Some metaheuristics based on various OBL techniques are: an advanced charged system search (ACSS) [76], opposition-based differential evolution (ODE) [77], [78], opposition-based harmony search (OHS) [79], opposition-based sine cosine algorithm (OBSCA) [80], and improved salp swarm algorithm (ISSA) [81].

A recent algorithm called equilibrium optimizer algorithm (EO) was developed by Faramarzi *et al.* to predict equilibrium states, where physical dynamics and sink models are the main inspiration for the algorithm [82]. EO belongs to the physics-based group of optimization algorithms where it is based on the laws of physical theory in nature. As addressed above, metaheuristic algorithms have shown supporting positive effects on features selection problems over recent decades. Despite all the research in this direction, most of the metaheuristic algorithms still face several challenges that need to be addressed. For instance, entrapment in local optima, lack of diversity, and imbalance between the explorative and exploitative abilities of the algorithm. It is still necessary to have more optimization techniques to get additional enhanced results. To the best of the author's information, there are a few studies in the literature for the binary version of EO [83], [84]. This has motivated us in this study to propose a new binary version of EO and test its benefit in features selection problems as a binary optimization algorithm. The main contributions of this paper are summarized as follows:

1) IBEO: a new modified version of EO is introduced. Two different transfer functions are used to deal with the binary problem.
2) OBL is applied to enhance the population diversity of EO at the initialization stage.
3) Local search algorithm (LSA) is applied to enhance the exploitation capability of the EO. At the end of each EO iteration, LSA is integrated to prevent it from getting stuck in local optima.
4) Test the proposed algorithm by comparing its result with six state-of-the-art features selection methods.

The remainder of this paper is organized as follows: Section 2 describes the original EO. The proposed algorithm is discussed in detail in section 3, while the experimental results are given in section 4. Finally, section 5 concludes the paper.

II. EQUILIBRIUM OPTIMIZER ALGORITHM: AN OVERVIEW

The equilibrium optimizer algorithm (EO) is a recent algorithm applied in continuous optimization problems [82]. EO belongs to physics-based methods. Particles and concentrations in EO can be portrayed as particles with their positions in particle swarm optimization algorithm (PSO). According to the best solutions or the equilibrium candidates, the concentrations of the particles (search agents) are randomly updated until the optimal result equilibrium state is obtained. One of the advantages of EO is its ability to randomly update the solution within a high balance between exploration and exploitation. Eq.(1) represents the update pattern for the concentration of a solution in the EO Algorithm:

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

The concentration of each particle is updated through three terms. The equilibrium concentration is the first term, and it is picked randomly from a pool, a vector that contains five equilibrium candidates or five best solutions, named the equilibrium pool. The second component serves as a direct exploration mechanism, and it indicates the difference between the concentration of a solution and the concentration of an equilibrium state. This term acts as explorers, which supports particles to explore the domain globally. Moreover, F's presence in the second term will help EO achieve a sensible balance between exploration and exploitation. The generation rate is the third term. A mathematical explanation of these terms is given in the next subsections.

1) CANDIDATES AND EQUILIBRIUM POOL

The convergence state is called the equilibrium state of the algorithm that is supposed to be the global optimum. There is no information about the concentration equilibrium state at the beginning of the optimization process. Thus, equilibrium candidates are useful in providing a search pattern for the particles. As mentioned before, the EO algorithm creates an equilibrium pool, a vector that contains five equilibrium candidates. Four candidates are the finest concentrations named in the entire process optimization plus their average. These four candidates support EO in exploration, while the average particle supports having better exploitation capability. It is worth considering that the number of candidates is optional and connected with the nature of the optimization problem. The equilibrium pool vector can be expressed as:

$$\vec{C}_{eq,pool} = \left\{\vec{C}_{eq(1)}, \vec{C}_{eq(2)}, \vec{C}_{eq(3)}, \vec{C}_{eq(4)}, \vec{C}_{eq(ave)}\right\} \tag{2}$$

The particle's concentration update is executed after each iteration by a random selection among candidates that have the same probability of being selected.

2) EXPONENTIAL TERM: F

As mentioned previously, the exponential term F in Eq.(3) can enormously assist balancing in EO exploitation and exploration. An explanation of the exponential term F can be expressed as:

$$\vec{F} = e^{-\vec{\lambda}(t-t_0)} \tag{3}$$

Here, $\vec{\lambda} \in [0], [1]$ is a random vector, t denotes a function of iteration that is decreasing with the number of iterations, and t_0 slows down the search speed to assure the convergence:

$$t = \left(1 - \frac{Iter}{Max\_iter}\right)^{a_2 \frac{Iter}{Max\_iter}} \tag{4}$$

$$t_0 = \frac{1}{\vec{\lambda}} \ln(-a_1 sign(\vec{r} - 0.5)[1 - e^{-\vec{\lambda} t}]) + t \tag{5}$$

Here, Iter and Max_iter represent the current and the maximum number of iterations, respectively. $\vec{r} \in [0], [1]$ is a random vector, $sign(r - 0.5)$ affects exploration and exploitation directions, a_1 and a_2 are constants. In the related work, $a_1 = 2$ and $a_2 = 1$. The higher a_1 value proposes better exploration performance and worse exploitation ability. The higher of a_2 value means better exploitation performance and worse exploration ability. By substituting Eq.(5) in Eq.(3), the last explanation of the exponential term is as following:

$$\vec{F} = a_1 sign(\vec{r} - 0.5) \left[e^{-\vec{\lambda} t} - 1\right] \tag{6}$$

3) GENERATION RATE (G)

The third term in Eq.(1) comes with the generation rate (G), it performs a vital role in EO to improve the exploitation process:

$$\vec{G} = \vec{G}_0 \vec{F} \tag{7}$$

$$\vec{G}_0 = G\vec{C}P(\vec{C}_{eq} - \vec{\lambda}\vec{C}) \tag{8}$$

$$G\vec{C}P = \begin{cases} 0.5 r_1 & r_2 \geq GP \\ 0 & r_2 \prec GP \end{cases} \tag{9}$$

where r_1 and r_2 are random numbers in the interval of [0], [1], GCP is an abbreviation for generation rate control parameter; it implies the probability of generating term addition towards the updating rule. GP is an abbreviation for Generation Probability, $GP = 0.5$ and it guarantees a fair balance between exploitation and exploration. GP denotes the probability that tells how many particles update their concentration using the generation term.

III. THE PROPOSED ALGORITHM (IBEO)

In this section, a wrapper-based method is proposed for tackling the problem of features selection. Initialization, transformation function, local search algorithm (LSA), and evaluation are the essential steps of IBEO. A detailed description of each step will be presented in the next subsections.

A. INITIALIZATION

Several studies have raised the quality of their initiated population solutions by using an optimization technique called opposition based learning (OBL). The OBL strategy works by diversifying the solutions to give a better probability of discovering promising regions. It searches the two directions in search space. The first direction is the primary solution, and the other direction is the opposite solution. It is reasonable to assume that if current solutions are far from the unknown optimal solution, computing the opposite solutions will lead to the opposite direction toward the unknown optimal solution. The purpose of the OBL approach is to take the fittest solutions from all solutions. Let c be a real number that belongs to the interval [lb, ub]. The opposite number of c is indicated by:

$$\tilde{c} = lb + ub - c \tag{10}$$

Eq.(10) can be generalized to use it in features selection problems. The following equations can express the concentration of each particle and its other direction (opposite solution) in EO:

$$c = [c_1, c_2, c_3, \ldots, c_d] \tag{11}$$

$$\tilde{c} = \left[\tilde{c}_1, \tilde{c}_2, \tilde{c}_3, \ldots, \tilde{c}_d\right] \tag{12}$$

All elements values in $\tilde{c}$ can be defined by:

$$\tilde{c}_j = lb + ub - c_j \quad where \quad j = 1, 2, 3, \ldots, d \tag{13}$$

where lb and ub are lower and upper bounds of each value in the current solution, respectively. In this strategy, lb and ub are utilized to calculate the opposite of the current solution. Then, two populations, the set of the opposite solutions OC and its initial form C, are combined and the fittest solutions are selected from $C \cup OC$ as the new initial population.

In the IBEO initialization stage, a number of particles are randomly generated depending on the size of the population. Each particle serves as an available solution with a dimension d, where d is the number of the all features that exist in the original dataset. OBL applied only in the initialization step to get the opposite solution of every generated solution. The new initial population of IBEO is created by choosing the best solutions from the set of the opposite solutions and its initial form. IBEO determines the fitness values based on kNN or SVM classifiers.

B. TRANSFORMATION FUNCTION

Several optimization problems have been modeled as binary problems such as feature selection problems. As mentioned before, the particle concentration produced from the original EO has continuous values. Thus, the conversion process from the continuous space of the original EO to a binary search space requires a transformation function. The concentrations of the particles in feature subset selection problems can only be 0 or 1. In IBEO, the binary solution space is represented by a population size $n \times$ number of features N matrix. The **1/0** values indicate that the related feature is selected/unselected, respectively. Fig. 1 represents the binary search space in IBEO. This study uses two different transformation functions to convert the continuous space into a binary search space [85]. Later in the results and analysis section, the performance of these functions will be examined. The sigmoidal function is an instance of the S-shaped family transfer functions as given in:

$$c_{s_i}^k(t) = \frac{1}{1 + e^{-c_i^k(t)}} \tag{14}$$

where the i^{th} particle of the k^{th} dimension at the t^{th} iteration is a continuous value $c_i^k(t)$ calculated by Eq.(1). Eq.(14) still shows the output for the S-shaped function in a continuous manner. To get the binary value, the i^{th} particle concentration is updated as follows:

$$c_i^k(t+1) = \begin{cases} 0 & rand < c_{s_i}^k(t) \\ 1 & rand \geq c_{s_i}^k(t) \end{cases} \tag{15}$$

where rand is a random number in the interval of [0, 1]. The second function is categorized as an instance of the V-Shaped function:

$$c_{v_i}^k(t) = \left| \frac{c_i^k(t)}{\sqrt{1 + (c_i^k(t))^2}} \right| \tag{16}$$

The V-shaped function's output in Eq.(16) still appears in a continuous manner. To get the binary value, the i^{th} particle concentration is updated as follows:

$$c_i^k(t+1) = \begin{cases} 0 & rand < c_{v_i}^k(t) \\ 1 & rand \geq c_{v_i}^k(t) \end{cases} \tag{17}$$

S-shaped and V-shaped functions transform a continuous space to a binary search space through the following procedure:

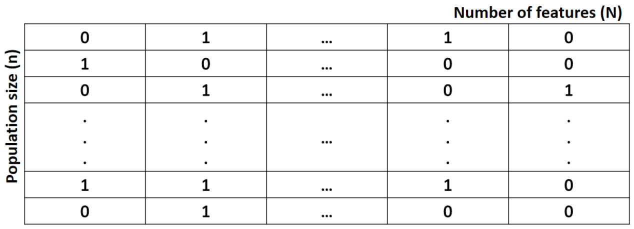
1) Passing the continuous value $c_i^k(t)$ value to the transfer function.
2) Considering $c_i^k(t)$, $c_{s_i}^k(t)$ or $c_{v_i}^k(t)$ are calculated.
3) Generating a random value between 0 and 1 for every $c_i^k(t)$.
4) Assigning the value 0 or 1 to the $c_i^k(t+1)$ using Eq.(15) or Eq.(17).

C. LOCAL SEARCH ALGORITHM (LSA)

The existing best solution B is passed to a local search algorithm in each iteration with the aim of discovering a better solution. LSA selects randomly three features in each iteration from the existing best solution. The algorithm switches the value of these features from 1 to 0 and vice-versa. Afterward, the fitness value for the new solution is calculated. LSA will update the value B only if the new solution gives a better fitness value than the current solution (Algorithm 1).

D. EVALUATION

Scalability of ML algorithms becomes a serious issue when dealing with datasets that have a large number of features. One of the concerns with adding more features to the data is that the redundant and irrelevant features have a negative influence on the performance of the classifier in many ways [86]. Moreover, there will be a need to add more instances, which will cause the classifier to take a longer time to learn. In such cases, the dimensionality of the data must be reduced. The feature selection enhances the learning time and the accuracy performance of a given classifier by eliminating unnecessary and irrelevant features. In this regard, we need to choose only the most relevant features and simultaneously raise the classification performance.



Population size (n) \ Number of features (N)				
0	1	...	1	0
1	0	...	0	0
0	1	...	0	1
. . .	. . .	...	. . .	. . .
1	1	...	1	0
0	1	...	0	0

FIGURE 1. Binary search space for IBEO.

Algorithm 1 LSA Algorithm

```
$temp\_value = B$ where $B$ is the current best solution
while $t < max\_iterations$ do
    Randomly select three features from temp_value
    if $selected\_feature == 1$ where 1 means the feature is selected and 0 means not selected) then
        $selected\_feature = 0$
    else
        $selected\_feature = 1$
    end if
    Evaluate temp_value using kNN or SVM classifiers
    Calculate the fitness value of temp_value
    if $f(temp\_value) < f(B)$ then
        $B = temp\_value$
    end if
    $t = t + 1$
end while
return $B$
```

The features selection problem is known as a multi-objective optimization problem (MOP) since it needs to obtain the following:

- Reduce the number of selected features.
- Improve the classifier performance by maximizing the accuracy value.

In order to balance between the two objectives, the fitness function is used in IBEO to evaluate the solutions:

$$Fitness = \alpha Error(D) + \beta \frac{|M|}{|N|} \tag{18}$$

where error (D) is the rate of classification error that is calculated using the kNN or SVM classifier. α and β are the weight parameters where $\alpha \in [0, 1]$ and $\beta = 1 - \alpha$. The two parameters α and β reflect the importance of the classification accuracy and the length of the selected feature subset. $|N|$ is the number of the original features. $|M|$ is the size of the feature selected. kNN or SVM is employed in IBEO as a classifier [11], [87]. We prefer to use the SVM classifier if a dataset has two classes. Otherwise, the kNN classifier is used. Fig. 2 represents features selection processes for IBEO. The steps of the IBEO are shown in Algorithm 2.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

We have tested all algorithms on Matlab Software (ver. R2016a) that is installed on Microsoft Windows 10, 64-bit Edition, Intel Core i7-3630QM processor, 2.40 GHz and 8 GB RAM machine.

A. DATASETS

There are 25 datasets taken from https://www.openml.org were used for verifying and evaluating the performance of IBEO compared to other algorithms. Table 1 demonstrates a brief overview of the used datasets, which contain different numbers of classes (from 2 to 10), different numbers of instances (from 47 to 6435), and different numbers of attributes (from 10 to 7129).

B. PARAMETER CONFIGURATION

The IBEO performance is compared to different state-of-the-art features selection methods. Each algorithm has 20 independent runs. For all experiments, the population size is equal to 5 and the maximum number of iterations is set to 30. The classification process is responsible for classifying new incoming instances where the class label is unknown. kNN and SVM are the preferred classifiers in the present study. To produce the optimal subset, the 5-NN classifier is preferred for datasets that contain more than two classes. Several trials and runs are performed on various datasets to figure out which K value in kNN is the best. For both kNN and SVM, K-fold cross-validation is equal to 10 to reduce the over-fitting problem. The idea of k-fold cross-validation depends on splitting the dataset into k-folds (subsets) that have almost equal size. Of the k subsets, $k-1$ subsets will be used to train the classifier and then the remaining single subset is treated as testing data for predicting the class label of each instance. Following that, the classification percentage error rate is determined as a percentage of the inaccurate class label predictions. Based on domain knowledge, parameters such as α and β in the fitness function are defined. The rest of the parameters are defined by trial and error. The parameters of IBEO are shown in Table 2.

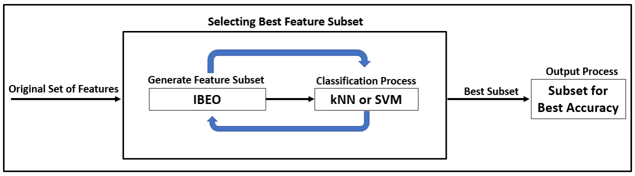


FIGURE 2. Features selection processes for IBEO.

Algorithm 2 Proposed Algorithm (IBEO)

```
Initialize population $C$ as $c_i \quad i = 1, \ldots, n$.
Initialize the parameters $a_1 = 2, a_2 = 1, GP = 0.5$.
Calculate the particle's opposite population OC using Eq.(13).
Select the $n$ fittest particles from $\{C \cup OC\}$ which represent the initial IBEO population.
Define $t = 0$
while $t < max\_iterations$ do
    Construct the equilibrium pool using using Eq.(2).
    for each particle do
        Update concentrations $\vec{C}$ using Eq.(1).
    end for
    Convert the concentration of particles into binary ones.
    Evaluate each particle in the population using kNN or SVM classifiers.
    Calculate the fitness of all the particles in the population using Eq.(18).
    $B = bestsolution$
    Apply LSA on $B$ to find if there is a better solution.
    $t = t + 1$
end while
return $B$
```

C. RESULTS AND ANALYSIS

The first experiment studies the effect of two different transformation functions on the original EO. The second experiment introduces a comparison of the proposed IBEO with the original EO and another binary version of EO called BEO that was published in 2020 [83]. In the third experiment, a comparison between IBEO and different algorithms such as PSO, GOA, GWO, WOA, DA, and ISSA is performed. The last-mentioned comparison is made by using three measures:

- Fitness value
- Classification accuracy
- Number of selected features

1) COMPARISON BETWEEN PERFORMANCES OF TWO TRANSFORMATION FUNCTIONS ON EO

A binary version of EO is executed using an instance of the S-shaped family transfer functions (EO-S) and an instance of the V-shaped family transfer functions (EO-V). The comparison based on the fitness function value is made between EO-S and EO-V by using three performance measures:

- The minimum fitness value (i.e. best fitness value) that is reached after running the two algorithms 20 times.

$$Bestfitness = \min_{i=1}^{20} f_i \tag{19}$$

- The mean fitness value that is reached after running the two algorithms 20 times.

$$Meanfitness = \frac{1}{20}\sum_{i=1}^{20} f_i \tag{20}$$

- The maximum fitness value (i.e. worst fitness value) that is reached after running the two algorithms 20 times.

$$Worstfitness = \max_{i=1}^{20} f_i \tag{21}$$

As seen in Table 3, EO-S works for most of the datasets better than EO-V. The best results are indicated by bold font. Moreover, Fig. 3 shows the performance of the two algorithms by using the total improvement percentage (IP).

TABLE 1. Selected datasets to be used in the experimental studies.

No.	Dataset	No. of Features	No. of Samples	No. of Classes
1	Fri_c0_500_10	10	500	2
2	Fri_c0_1000_10	10	1000	2
3	Fri_c1_1000_10	10	1000	2
4	HeartEW	13	270	5
5	Australian	14	690	2
6	Zoo	17	101	7
7	Lymphography	18	148	4
8	Vehicle	18	846	4
9	Segment	19	2310	7
10	Climate	20	540	2
11	Credit	20	1000	2
12	Kc1	21	2109	2
13	WaveForm	21	5000	4
14	Parkinsons	22	195	2
15	SpectEW	22	267	2
16	Dermatology	34	366	6
17	IonosphereEW	34	351	2
18	Satellite	36	6435	6
19	Spambase	57	4601	2
20	SonarEW	60	208	2
21	Robot-failures-lp1	90	88	4
22	Robot-failures-lp2	90	47	5
23	Semeion	256	1593	10
24	CNAE	856	1080	9
25	Leukemia	7129	72	2

TABLE 2. Parameter configuration for IBEO.

Parameter	Value
Number of search agents	5
Number of independent runs	20
Dimension	Number of features
Number of iterations	30
α	0.01
β	0.99
K-fold cross-validation	10
K-neighbors	5
a_1	2
a_2	1
GP	0.5

IP is the ratio of positive change in the two algorithms. IP can be calculated using the following formula:

$$IP = \sum_{i=1}^{m} \frac{full - f_{alg}}{full} \times 100 \tag{22}$$

where full is supposed to be the fitness value of choosing all dataset's original features, f_{alg} is the fitness value of the worst, mean, and best case in EO-S and EO-V. The value m is the number of datasets ($m = 25$). The above figure shows that the IP of the two algorithms has been increased and it is worth noting that the performance has considerably enhanced in comparison with choosing all features in every dataset. Besides, EO-S has better IP than EO-V for all datasets.

2) COMPARISON OF IBEO WITH EO AND BEO

The former experiment tested the performance of EO-S and EO-V to decide which one is better. The results showed that EO-S performs better than EO-V. Now is the time to investigate IBEO and determine the impact of incorporating the OBL strategy and the LSA algorithm into the EO-S. Thus, the proposed IBEO is compared with the original EO and BEO. K. Ghosh proposed BEO as the first binary version of EO, which improved EO's exploitative abilities by using simulated annealing algorithm [83]. The average number of the selected features, classification accuracy, and fitness values are the most crucial comparative performance measurements in our experiment.

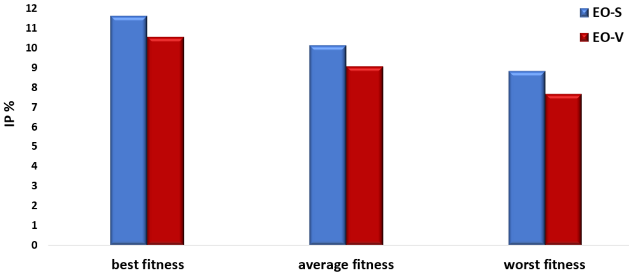


FIGURE 3. Comparison between EO-S and EO-V according to the IP of the fitness values.

As presented in Table 4, the classification accuracy of the IBEO algorithm in 24 cases out of 25 datasets exceeded the original EO (96%). While IBEO outperformed BEO in classification accuracy across 22 datasets. In the case of exactly, IBEO has shown that the remaining three datasets, HeartEW, Satellite, and Leukemia, have classification accuracy values that are too close to BEO values. Even if we keep aside classification accuracy and concentrate on the number of features used to obtain these results, IBEO performed best in 17 of the 25 datasets. Whereas, EO scored 2 datasets while BEO scored 6 datasets. IBEO has only used about half of the features for the classification process. For all 25 data sets, IBEO outperforms the original algorithm in terms of fitness values. While IBEO's fitness values across 20 datasets exceeded BEO. Fig. 4 shows a comparison among EO, BEO, and IBEO based on the total average of the fitness value, classification accuracy, and number of selected features for all the datasets. The IBEO exceeds both algorithms with a total average fitness value of 0.1563 and a total average

TABLE 3. Comparison between performances of EO-S and EO-V according to the fitness value.

No.	Dataset	Best fitness		Mean fitness		Worst fitness		Full
		EO-S	EO-V	EO-S	EO-V	EO-S	EO-V	
1	Fri_c0_500	**0.1616**	0.1714	**0.1883**	0.1912	0.2146	**0.211**	0.289
2	Fri_c0_1000	**0.1634**	**0.1634**	**0.1874**	0.1884	0.2104	**0.2074**	0.2899
3	Fri_c1_1000	**0.12**	0.1209	**0.1382**	0.1441	**0.1731**	0.1795	0.2935
4	HeartEW	**0.3092**	0.3246	**0.3233**	0.3410	**0.3368**	0.3568	0.4765
5	Australian	**0.1375**	0.1446	**0.1551**	0.2098	**0.1824**	0.26	0.3739
6	Zoo	**0.0828**	0.0798	**0.0994**	0.1048	**0.1122**	0.1242	0.6168
7	Lymphography	**0.1716**	**0.1716**	**0.1839**	0.1918	**0.2041**	0.2075	0.2824
8	Vehicle	**0.2865**	0.2944	**0.3042**	0.3113	**0.3193**	0.3271	0.4021
9	Segment	**0.0617**	0.0740	**0.0725**	0.0811	**0.0812**	0.0884	0.1506
10	Climate	**0.08**	0.0933	**0.0915**	0.1060	**0.0983**	0.1183	0.1916
11	Credit	**0.2703**	0.2811	**0.2824**	0.2978	**0.2911**	0.311	0.4078
12	Kc1	**0.1795**	0.1903	**0.1901**	0.1996	**0.1977**	0.2067	0.2698
13	WaveForm	0.2243	**0.2177**	0.2279	**0.2243**	0.2321	**0.2300**	0.2589
14	Parkinsons	**0.1012**	0.1195	**0.1213**	0.1324	**0.1378**	0.1468	0.2292
15	SpectEW	**0.2611**	0.2689	**0.2750**	0.2967	**0.2913**	0.3152	0.3966
16	IonosphereEW	**0.1187**	0.1441	**0.1300**	0.1602	**0.1400**	0.1697	0.2410
17	Dermatology	0.0654	**0.0650**	**0.0749**	0.0783	**0.0831**	0.0885	0.2005
18	Satellite	**0.1169**	0.1221	**0.1219**	0.1277	**0.1251**	0.1362	0.1823
19	Spambase	**0.1238**	0.1320	**0.1336**	0.1381	**0.1465**	0.1525	0.2772
20	SonarEW	**0.1415**	0.1711	**0.1670**	0.186	**0.1817**	0.2014	0.2557
21	Robot-failures-lp1	**0.1300**	0.1533	**0.1513**	0.1715	**0.1731**	0.1862	0.2636
22	Robot-failures-lp2	**0.3352**	0.3427	**0.3529**	0.3754	**0.3655**	0.3969	0.5021
23	Semeion	**0.1153**	0.1267	**0.1276**	0.1329	**0.1318**	0.1363	0.1791
24	CNAE	**0.1841**	0.1899	0.2117	**0.2034**	**0.2286**	0.2294	0.2375
25	Leukemia	**0.0402**	0.0727	**0.0662**	0.0865	**0.0800**	0.0951	0.1625

TABLE 4. Comparison between the performances of EO, BEO, and IBEO according to the average fitness value, average accuracy, and average number of selected features.

No.	Dataset	Fitness			Accuracy			Number of selected features		
		EO	BEO	IBEO	EO	BEO	IBEO	EO	BEO	IBEO
1	Fri_c0_500	0.1883	0.1702	**0.1634**	0.8349	0.8558	**0.866**	5.25	4.75	**4.65**
2	Fri_c0_1000	0.1874	0.1709	**0.1534**	0.8258	0.8596	**0.8836**	4.7	4.7	**4.65**
3	Fri_c1_1000	0.1382	0.1221	**0.1123**	0.8805	0.8974	**0.9036**	3.8	3.1	**3.05**
4	HeartEW	0.3233	**0.3111**	0.3197	0.6503	**0.6636**	0.6634	**2.7**	2.8	2.8
5	Australian	0.1551	0.1506	**0.1469**	0.8426	0.8523	**0.8592**	3.55	3.25	**3.1**
6	Zoo	0.0994	0.0938	**0.0891**	0.9356	0.943	**0.9480**	7.55	7.7	**7.2**
7	Lymphography	0.1839	**0.1731**	0.1737	0.8266	0.8479	**0.8486**	**7.05**	7.8	8.35
8	Vehicle	0.3042	0.2993	**0.2957**	0.6930	0.6998	**0.7060**	6.45	6.6	**5.85**
9	Segment	0.0725	0.0645	**0.0644**	0.9543	0.9593	**0.9614**	6.25	**5.4**	5.85
10	Climate	0.0915	**0.088**	0.0884	0.9109	0.9116	**0.9142**	3.65	2.95	**2.35**
11	Credit	0.2824	0.2682	**0.2626**	0.7145	0.7235	**0.7417**	5.8	**4.75**	6.8
12	Kc1	0.1901	0.1826	**0.1499**	0.8165	0.8207	**0.8469**	6	5.15	**2.95**
13	WaveForm	0.2279	0.2222	**0.2219**	0.7981	0.8029	**0.8066**	10.25	**9.5**	10.3
14	Parkinsons	0.1213	0.1138	**0.0762**	0.8874	0.8889	**0.9259**	6.95	5.95	**4.95**
15	SpectEW	0.2750	0.2629	**0.2421**	0.7183	0.7258	**0.7419**	7.85	6.6	**4.15**
16	IonosphereEW	0.1300	0.114	**0.0693**	0.8812	0.8954	**0.9430**	9.1	9.15	**8.35**
17	Dermatology	0.0749	0.0668	**0.0649**	0.9532	0.966	**0.9702**	13.85	**13.7**	14.1
18	Satellite	0.1219	**0.119**	0.1201	0.8965	**0.8978**	0.8956	10.9	10.05	**9.1**
19	Spambase	0.1336	0.1257	**0.1097**	0.8943	0.9019	**0.9057**	23.45	21.6	**14.95**
20	SonarEW	0.1670	0.1539	**0.1251**	0.8391	0.8526	**0.8807**	23.2	20.05	**19.65**
21	Robot-failures-lp1	0.1513	0.1398	**0.1392**	0.8602	0.8642	**0.8693**	33.75	31.55	**30.95**
22	Robot-failures-lp2	0.3529	0.3352	**0.3316**	0.6457	0.6531	**0.6542**	34.95	30.25	**28.75**
23	Semeion	0.1276	0.124	**0.1233**	0.8985	0.9005	**0.9026**	103.6	**98.35**	102.45
24	CNAE	0.2117	0.2191	**0.2047**	0.7697	0.7980	**0.8313**	349.35	**342.75**	352.8
25	Leukemia	0.0662	0.0617	**0.0628**	0.9465	**0.9590**	0.9534	2877.05	2844.2	**2742.65**

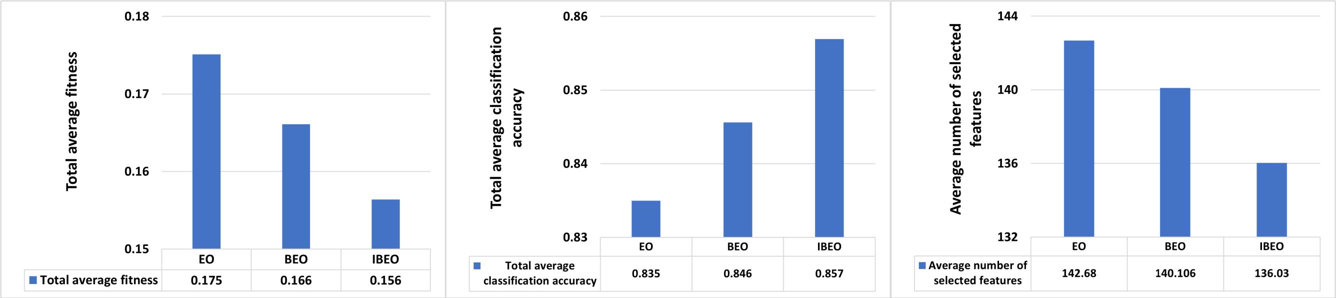


FIGURE 4. Total average fitness value, average accuracy, and average number of selected features over all datasets.

classification accuracy of 0.8569. Furthermore, the average number of the selected features over all the datasets was calculated and it has been reduced from 142.68 to 136.03. The use of OBL strategy and LSA algorithm clearly aids the original algorithm in exploring different parts of the search space and achieving better results. In Fig. 5, the average time for running overall the datasets 20 times is shown.

3) COMPARISON BETWEEN IBEO AND OTHER METAHEURISTIC ALGORITHMS

The previous experiment compared IBEO with the original EO. IBEO showed a higher performance over the standard EO algorithm because of the ability to balance between exploration and exploitation, the ability to escape from local optima, and the ability to improve population diversity. For results confirmation, the third experiment was conducted to compare our proposed algorithm with other six well-known algorithms (PSO, GOA, GWO, WOA, DA, ISSA). The following measures have been employed in the comparative performance evaluation: number of selected features, mean fitness, classification accuracy, and standard deviation. Table 5 shows the results of seven algorithms in terms of the average fitness values. IBEO shows minimum fitness values in most of the datasets compared to other metaheuristic algorithms. It can be seen from Table 5 that IBEO works best in 23 cases out of 25 datasets (92%). IBEO has been shown as the second-best in the case of Satellite and Leukemia. Furthermore, Fig. 6 presents the total mean fitness values over all the datasets. The average fitness value of IBEO is equal to 0.1564, which is the lowest value compared to the rest of the values.

In Table 6, a comparison is performed in terms of the classification accuracy that obtained by each algorithm. In most datasets, we found that IBEO outperformed other algorithms in terms of classification accuracy. Considering Table 6,

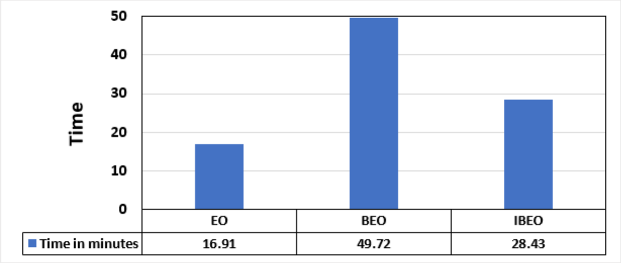


FIGURE 5. Average time in minutes for all datasets.

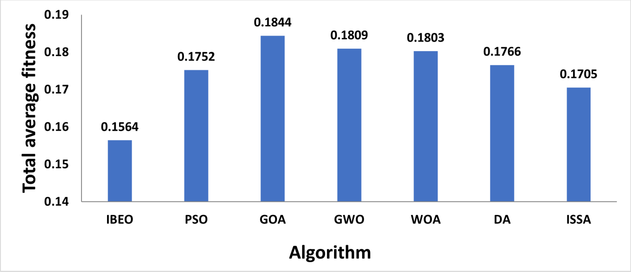


FIGURE 6. Total average fitness values of IBEO over all datasets compared with other algorithms.

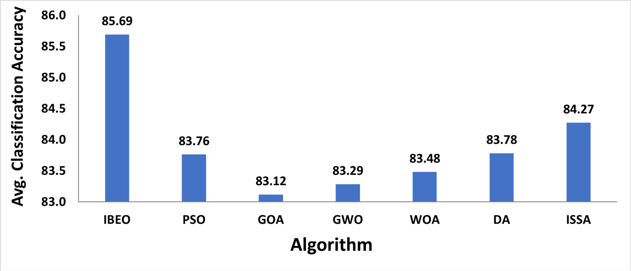


FIGURE 7. Average classification accuracy of IBEO over all datasets compared with other algorithms.

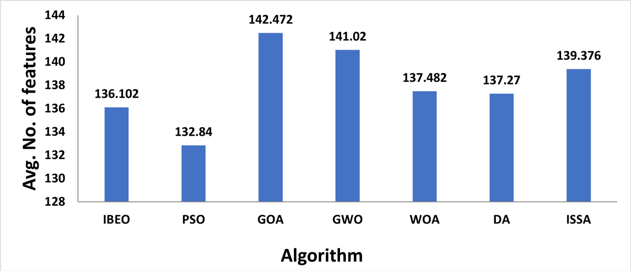


FIGURE 8. Average number of selected features of IBEO over all datasets compared with other algorithms.

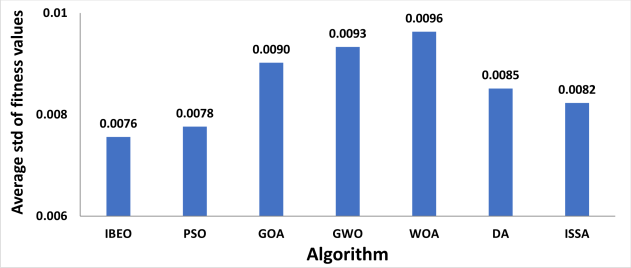


FIGURE 9. Total average standard deviation for the average fitness results of IBEO compared with other algorithms.

IBEO works best in 22 cases out of 25 datasets (88%). IBEO has been shown as the second-best in the case of Robot-failures-lp1. For Leukemia, it has achieved fifth-best coming after the results of PSO, ISSA, DA, and GOA, in this order. In Fig. 7, the average classification accuracy obtained by each algorithm over all datasets is reported. IBEO has reached 85.6% classification accuracy, and this is the best result compared to other algorithms. Moreover, Fig. 8 gives the average number of selected features for all datasets. IBEO has a value of 136.102 features, putting the proposed algorithm in second rank. Despite the fact that the PSO algorithm achieved the smallest average number of selected features, IBEO seems to be more helpful as the classification accuracy should take more consideration than the number of features. Additionally, the comparisons show that there is no huge difference between PSO and IBEO and the fitness values are influenced by the classification accuracy value more than the number of selected features. The advantage of the IBEO comes from its ability to balance the search process over iterations between exploration and exploitation. In Fig. 9, IBEO appears in the first rank according to the obtained total average standard deviation for the mean fitness values, indicating that the difference between the datasets' fitness values is smaller than the other algorithms. As aforementioned, IBEO has the smallest mean fitness value. Hence, it is concluded that the best and the worst fitness values are near the mean value.

Based on IBEO configuration, the maximum number of iterations in all the experiments is set to 30 and the population size is 5. LSA will update the value of the current best solution only if the new solution gives a better fitness value than the current solution. In this context, the evaluation function for the current best solution is called twice in each iteration. Thus, to ensure the efficiency of the proposed algorithm, algorithms that do not use iterated local search (PSO, GOA, GWO, WOA, DA) are tested again and we maximize the number of iterations to 60. Tables 7 and 8 give comparisons in terms of the average fitness value and the classification accuracy, respectively. The results show that the proposed algorithm continues to outperform the other algorithms across most of the datasets in both tables.

Based on the results shown in Fig. 10, IBEO exceeded all other algorithms in convergence over most of the datasets. Hence, the comparison test for convergence between IBEO and other algorithms showed the higher performance for IBEO. Besides, we can observe from Fig. 10 the convergence curves as it shows that the IBEO solved the premature convergence in most of the datasets by improving population diversity and balancing between exploitation and exploration. Fig. 11 presents the boxplots for all datasets

TABLE 5. Average fitness results of the IBEO compared with other algorithms (30 iterations).

No.	Dataset	IBEO	PSO	GOA	GWO	WOA	DA	ISSA
1	Fri_c0_500	**0.1634**	0.1878	0.2072	0.1962	0.1949	0.1891	0.1855
2	Fri_c0_1000	**0.1534**	0.1951	0.2133	0.1998	0.2029	0.1928	0.1833
3	Fri_c1_1000	**0.1123**	0.1356	0.1556	0.1483	0.1463	0.1417	0.1331
4	HeartEW	**0.3197**	0.3228	0.3275	0.3270	0.3286	0.3223	0.3203
5	Australian	**0.1469**	0.1522	0.1867	0.1755	0.1606	0.166	0.1575
6	Zoo	**0.0891**	0.0947	0.1047	0.1017	0.1014	0.0951	0.0919
7	Lymphography	**0.1737**	0.1845	0.1915	0.1902	0.1897	0.1816	0.1782
8	Vehicle	**0.2957**	0.3031	0.3082	0.3104	0.3101	0.302	0.2994
9	Segment	**0.0644**	0.0692	0.0756	0.0739	0.0752	0.0707	0.0659
10	Climate	**0.0884**	0.0912	0.0976	0.0955	0.0943	0.0931	0.0906
11	Credit	**0.2626**	0.2775	0.2869	0.2814	0.2828	0.2769	0.2734
12	Kc1	**0.1499**	0.1888	0.1936	0.1913	0.1911	0.1897	0.1857
13	WaveForm	**0.2219**	0.2284	0.2320	0.2302	0.2289	0.2269	0.2220
14	Parkinsons	**0.0762**	0.1216	0.1280	0.1261	0.1255	0.1220	0.1175
15	SpectEW	**0.2421**	0.2722	0.2888	0.2841	0.2862	0.2767	0.2707
16	IonosphereEW	**0.0693**	0.1227	0.1317	0.1335	0.1355	0.1241	0.1212
17	Dermatology	**0.0649**	0.0757	0.0816	0.0759	0.0780	0.0737	0.0671
18	Satellite	0.1201	0.1211	0.1229	0.1227	0.1222	0.1220	**0.1199**
19	Spambase	**0.1097**	0.1360	0.1382	0.1366	0.1367	0.1363	0.1239
20	SonarEW	**0.1251**	0.1682	0.1726	0.1714	0.1703	0.1675	0.1555
21	Robot-failures-lp1	**0.1392**	0.1495	0.1582	0.1508	0.1552	0.1520	0.1423
22	Robot-failures-lp2	**0.3316**	0.3425	0.3509	0.3499	0.3497	0.3467	0.3389
23	Semeion	**0.1233**	0.1272	0.1293	0.1294	0.1282	0.1384	0.1248
24	CNAE	**0.2047**	0.2494	0.2555	0.2522	0.2459	0.2375	0.2323
25	Leukemia	0.0628	0.0630	0.0715	0.0693	0.0665	0.0690	**0.0624**

TABLE 6. Classification accuracy results of the IBEO compared with other algorithms (30 iterations).

No.	Dataset	IBEO	PSO	GOA	GWO	WOA	DA	ISSA
1	Fri_c0_500	**0.866**	0.8405	0.8203	0.8312	0.8351	0.8396	0.8433
2	Fri_c0_1000	**0.8836**	0.8359	0.8191	0.8317	0.8319	0.8392	0.8543
3	Fri_c1_1000	**0.9036**	0.8899	0.8710	0.8713	0.8770	0.8799	0.8912
4	HeartEW	**0.6634**	0.6534	0.6542	0.6488	0.6545	0.6598	0.6612
5	Australian	**0.8592**	0.8505	0.82	0.8371	0.8476	0.8422	0.8505
6	Zoo	**0.9480**	0.9386	0.9302	0.9311	0.9306	0.9356	0.9401
7	Lymphography	**0.8486**	0.8307	0.8236	0.8256	0.8283	0.8391	0.8354
8	Vehicle	**0.7060**	0.6932	0.6949	0.688	0.6947	0.7014	0.7039
9	Segment	**0.9614**	0.9564	0.9528	0.9524	0.9536	0.9577	0.9596
10	Climate	**0.9142**	0.9114	0.9113	0.9095	0.9113	0.9104	0.9137
11	Credit	**0.7417**	0.7180	0.7102	0.7141	0.7137	0.7187	0.7227
12	Kc1	**0.8469**	0.8180	0.8156	0.8182	0.8183	0.8172	0.8198
13	WaveForm	**0.8066**	0.7973	0.7922	0.7958	0.7970	0.7955	0.8056
14	Parkinsons	**0.9259**	0.8892	0.8882	0.8871	0.8871	0.8923	0.8838
15	SpectEW	**0.7419**	0.7138	0.7095	0.7046	0.7078	0.7142	0.7198
16	IonosphereEW	**0.9430**	0.8856	0.8793	0.8789	0.8733	0.8846	0.8873
17	Dermatology	**0.9702**	0.9553	0.9530	0.9571	0.9568	0.9614	0.9656
18	Satellite	0.8956	**0.8982**	0.8965	0.8967	0.8973	0.8975	0.8978
19	Spambase	**0.9057**	0.8914	0.8905	0.8924	0.8948	0.8934	0.9043
20	SonarEW	**0.8807**	0.8432	0.8363	0.8377	0.8338	0.8389	0.8572
21	Robot-failures-lp1	0.8693	0.8613	0.8568	0.8568	0.8613	0.8653	**0.8715**
22	Robot-failures-lp2	**0.6542**	0.6425	0.6414	0.6414	0.6425	0.6383	0.6414
23	Semeion	**0.9026**	0.8983	0.8979	0.8974	0.899	0.8830	0.90009
24	CNAE	**0.8313**	0.7670	0.7594	0.7659	0.7725	0.7829	0.7799
25	Leukemia	0.9534	**0.9611**	0.9555	0.9506	0.9506	0.9576	0.9583

to compare the average performance of algorithms visually. Boxplots represent the classification accuracy values and they are plotted after executing each algorithm 20 times. Each boxplot can give information about five components: maximum, median, minimum, first quartile, and third quartile of the data. The whisker extending down represents the minimum value, the bottom of the rectangle represents quartile one, the top of the rectangle represents quartile three, the line that separates the rectangle is the median value, and the top whisker is the maximum value. Additionally, outliers may be plotted as individual points. It can be noticed that IBEO has higher boxplots characterized by higher median values compared to other algorithms in most of the datasets.

Furthermore, in order to determine whether there is a statistical difference between IBEO's results and those of the other comparative methods, the nonparametric Wilcoxon-based fitness function rank-sum test is conducted for 10 randomly selected datasets (Fri_c1_1000, Australian, Lymphography, Credit, Kc1, Parkinsons, Dermatology,

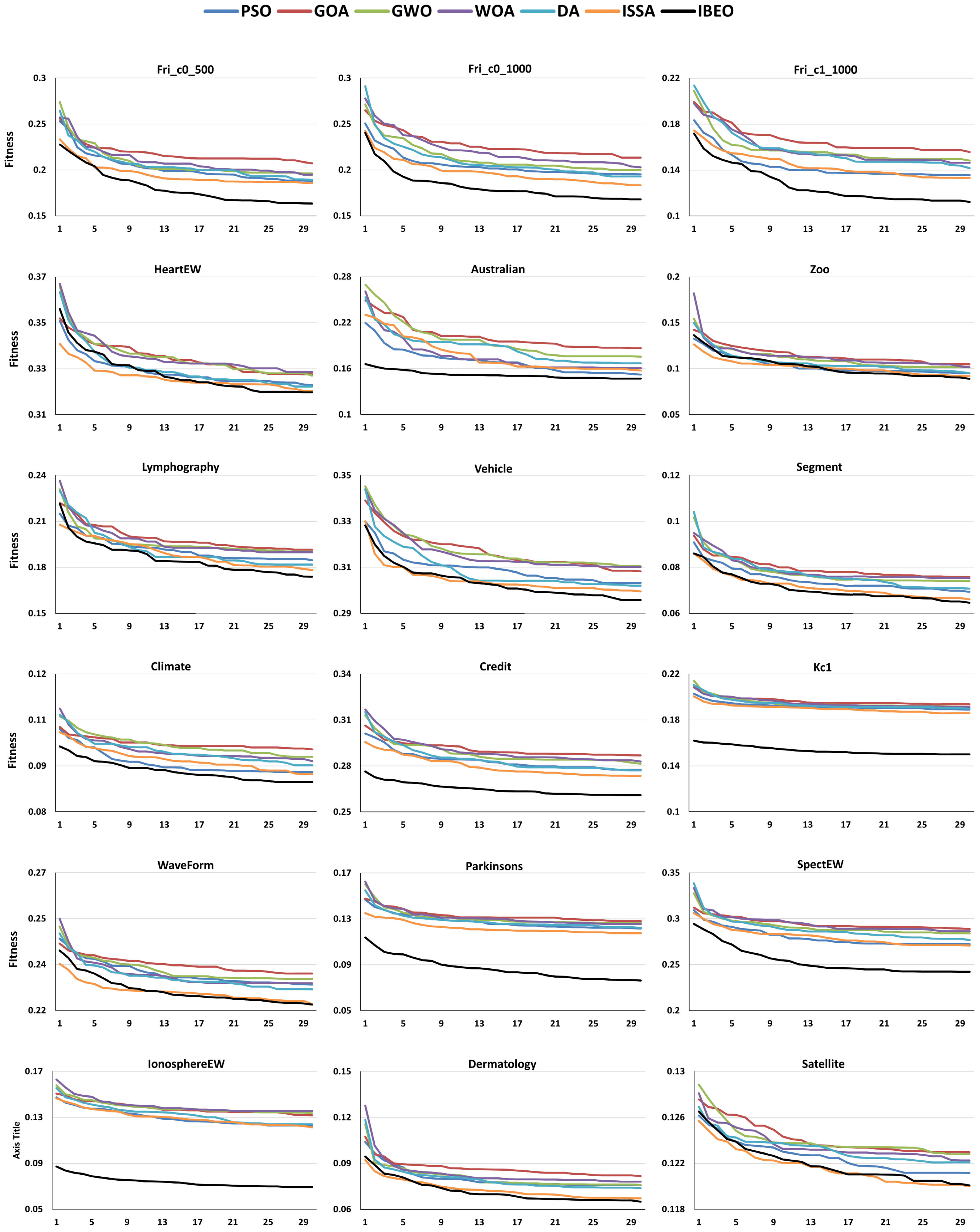


FIGURE 10. Convergence of IBEO compared with other metaheuristics.

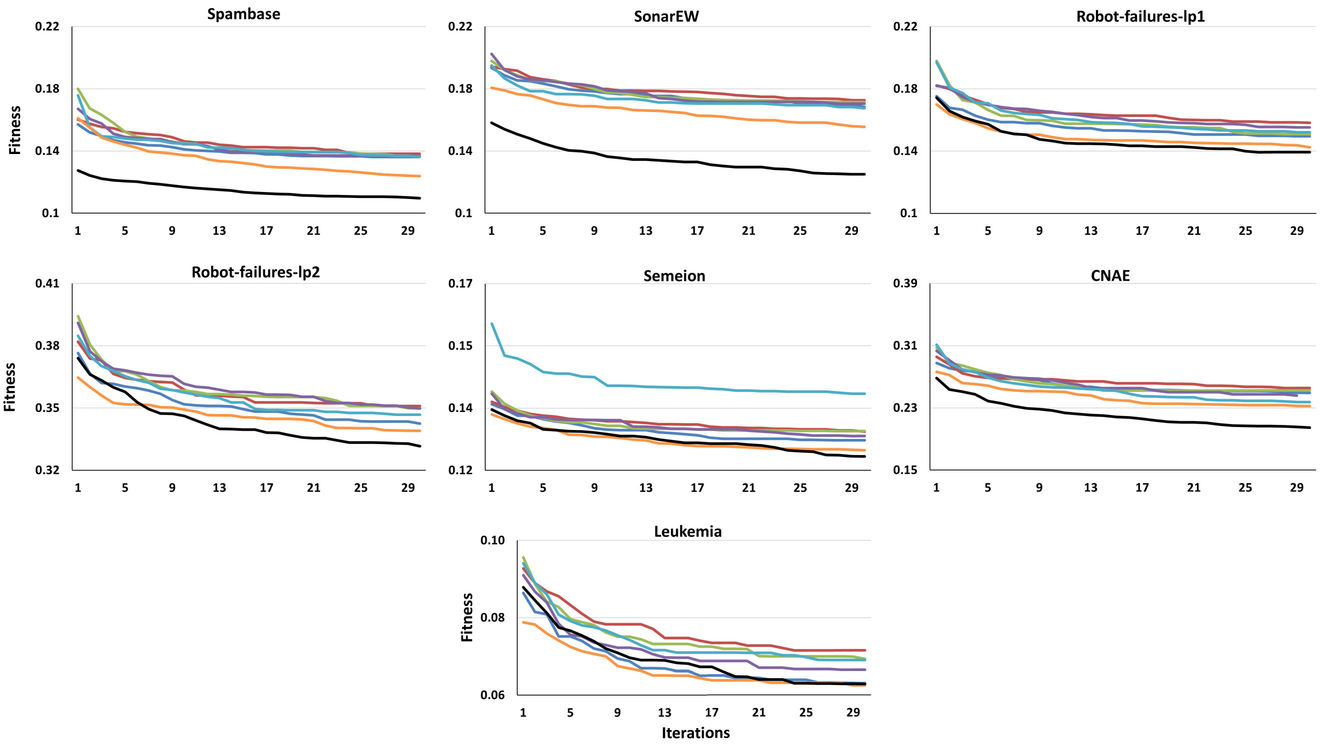


FIGURE 10. ***(Continued.)*** **Convergence of IBEO compared with other metaheuristics.**

TABLE 7. **Average fitness results of the IBEO compared with other algorithms (60 iterations).**

No.	Dataset	IBEO	PSO	GOA	GWO	WOA	DA
1	Fri_c0_500	**0.1634**	0.1763	0.1976	0.1939	0.1938	0.1827
2	Fri_c0_1000	**0.1534**	0.1850	0.1978	0.1929	0.1909	0.1813
3	Fri_c1_1000	**0.1123**	0.1328	0.1427	0.1398	0.1338	0.1279
4	HeartEW	0.3197	0.3191	0.3224	0.3223	0.3227	**0.3147**
5	Australian	**0.1469**	0.1506	0.1620	0.1648	0.1544	0.1552
6	Zoo	**0.0891**	0.0930	0.0986	0.0967	0.09334	0.0929
7	Lymphography	**0.1737**	0.1837	0.1856	0.1857	0.1864	0.1813
8	Vehicle	**0.2957**	0.3001	0.3062	0.3032	0.3040	0.3008
9	Segment	**0.0644**	0.0670	0.0749	0.0716	0.0715	0.0687
10	Climate	**0.0884**	0.0901	0.0948	0.0912	0.093	0.0916
11	Credit	**0.2626**	0.2733	0.2814	0.2787	0.2767	0.2760
12	Kc1	**0.1499**	0.1863	0.1930	0.1892	0.1872	0.1886
13	WaveForm	**0.2219**	0.2248	0.2299	0.229	0.2273	0.2260
14	Parkinsons	**0.0762**	0.1177	0.1220	0.1222	0.11888	0.1179
15	SpectEW	**0.2421**	0.2680	0.2798	0.2768	0.2751	0.2715
16	IonosphereEW	**0.0693**	0.1190	0.1304	0.1312	0.1280	0.1287
17	Dermatology	**0.0649**	0.0700	0.0753	0.0743	0.0728	0.0721
18	Satellite	**0.1201**	0.1203	0.1224	0.1215	0.1215	0.1204
19	Spambase	**0.1097**	0.1285	0.1362	0.1339	0.1358	0.1290
20	SonarEW	**0.1251**	0.1614	0.1679	0.1658	0.1664	0.1659
21	Robot-failures-lp1	**0.1392**	0.1425	0.1519	0.1507	0.1472	0.1444
22	Robot-failures-lp2	**0.3316**	0.3384	0.3490	0.3461	0.3428	0.3403
23	Semeion	**0.1233**	0.1254	0.1268	0.1273	0.1268	0.1254
24	CNAE	**0.2047**	0.2350	0.2465	0.2425	0.2275	0.2323
25	Leukemia	**0.0628**	0.0636	0.0660	0.0640	0.0631	0.0660

SonarEW, Robot-failures-lp2, and CNAE) [88]. The Wilcoxon test's significance level is set at 0.05, and the results are presented in Table 9. The number of positive ranks in which IBEO outranks the comparative algorithms is referred to as No. R+. The number of negative ranks in which the IBEO fails to outrank the comparative algorithms is represented by No. R−. The number of ties is the number of equal ranks for the IBEO with the other comparative algorithm.

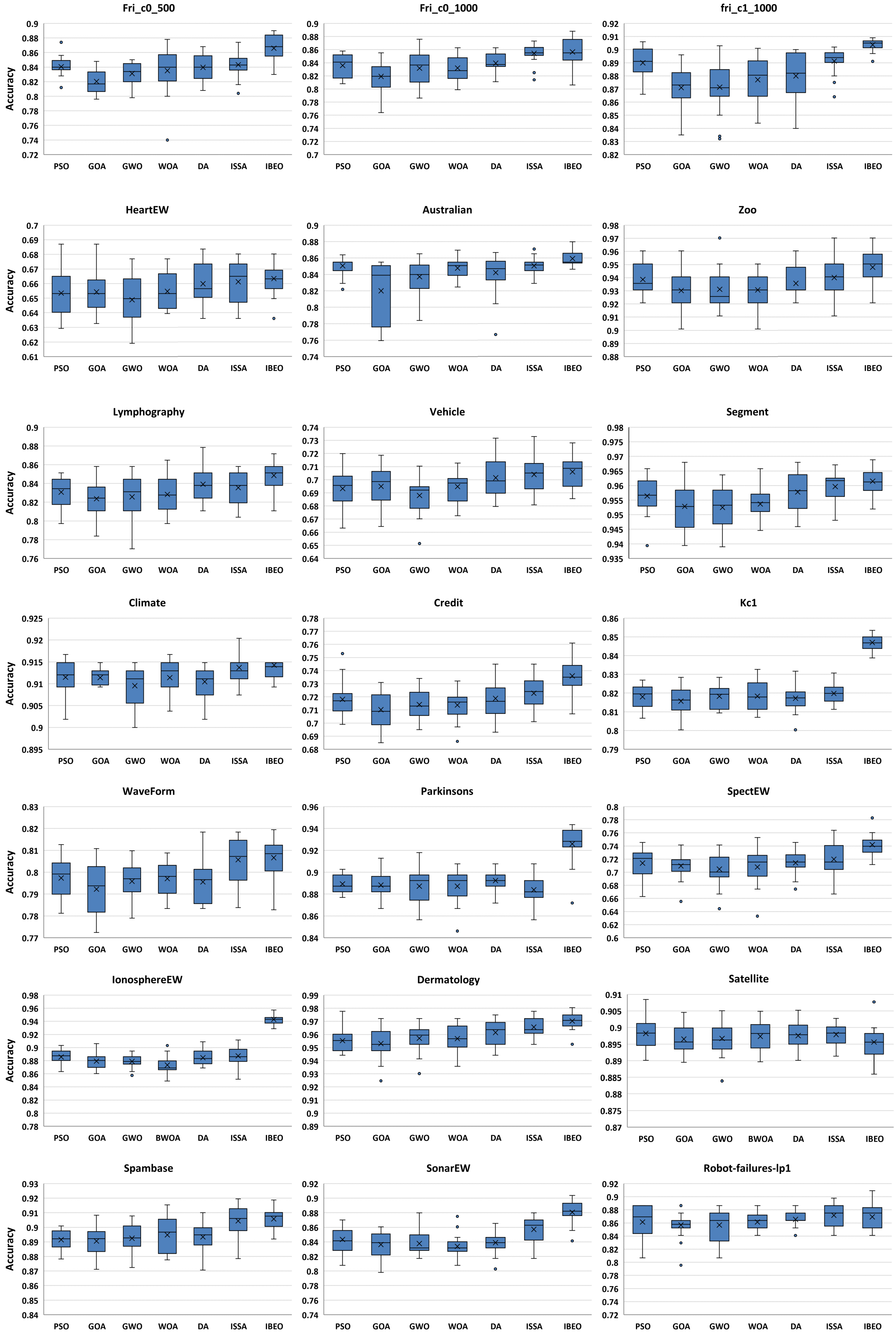


FIGURE 11. Boxplots of IBEO compared with other metaheuristics over all datasets.

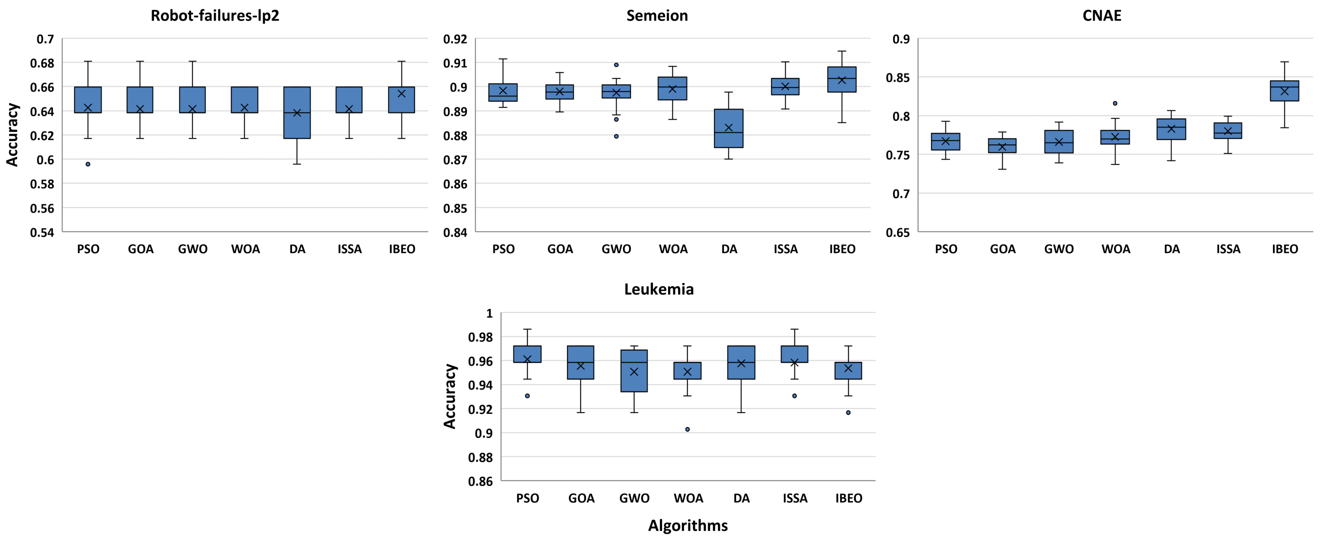


FIGURE 11. ***(Continued.)*** **Boxplots of IBEO compared with other metaheuristics over all datasets.**

TABLE 8. **Classification accuracy results of the IBEO compared with other algorithms (60 iterations).**

No.	Dataset	IBEO	PSO	GOA	GWO	WOA	DA
1	Fri_c0_500	**0.866**	0.854	0.8294	0.8376	0.837	0.8417
2	Fri_c0_1000	**0.8836**	0.8494	0.833	0.834	0.8346	0.844
3	Fri_c1_1000	**0.9036**	0.8908	0.8788	0.8826	0.8878	0.8949
4	HeartEW	**0.6634**	0.6573	0.6591	0.6603	0.6576	0.6568
5	Australian	**0.8592**	0.8542	0.8449	0.8434	0.8472	0.8479
6	Zoo	**0.9480**	0.9366	0.9287	0.9391	0.9346	0.9410
7	Lymphography	**0.8486**	0.8283	0.8371	0.8348	0.8361	0.8236
8	Vehicle	**0.7060**	0.7005	0.6883	0.6976	0.6961	0.6981
9	Segment	**0.9614**	0.9555	0.9541	0.9539	0.9553	0.9548
10	Climate	**0.9142**	0.9120	0.9114	0.9129	0.9112	0.9123
11	Credit	**0.7417**	0.7197	0.7176	0.7154	0.7155	0.7200
12	Kc1	**0.8469**	0.8162	0.8183	0.8173	0.8176	0.8180
13	WaveForm	**0.8066**	0.7969	0.7965	0.7995	0.8016	0.7989
14	Parkinsons	**0.9259**	0.8843	0.8848	0.89	0.8841	0.8887
15	SpectEW	**0.7419**	0.7194	0.7153	0.7168	0.72116	0.7292
16	IonosphereEW	**0.9430**	0.8907	0.8823	0.8794	0.8779	0.8837
17	Dermatology	**0.9702**	0.9628	0.9603	0.9596	0.9649	0.9643
18	Satellite	0.8956	0.8965	0.8966	0.8962	**0.8987**	0.8967
19	Spambase	**0.9057**	0.8987	0.8939	0.8980	0.8984	0.9005
20	SonarEW	**0.8807**	0.8468	0.8324	0.8447	0.8384	0.8413
21	Robot-failures-lp1	0.8693	0.8676	0.8608	0.8619	0.8608	0.8642
22	Robot-failures-lp2	**0.6542**	0.6436	0.6383	0.6393	0.6436	0.6489
23	Semeion	**0.9026**	0.8990	0.8974	0.8964	0.8990	0.9004
24	CNAE	**0.8313**	0.7797	0.7694	0.7737	0.7927	0.7852
25	Leukemia	0.9534	**0.9597**	0.9513	0.9555	**0.9597**	0.9569

The total of positive and negative ranks is represented by Sum_ R+ and Sum_ R−, respectively.

According to Table 9, the No. R+ in which IBEO outperforms PSO, GOA, GWO, WOA, DA, and ISSA are 10 cases out of the 10 randomly selected datasets. For instance, in the Fri_c1_1000 dataset, the number of runs in which IBEO outperforms PSO is 19 out of 20 runs. In the Robot-failureslp2 dataset, the number of runs in which IBEO outperforms GWO is 16 out of 20 runs and it fails to outrank GWO in 3 runs and shows a similar performance in one run. Interestingly, IBEO has obtained the top performance in all 20 runs for the datasets: Kc1, Parkinsons, and SonarEW. For the datasets used in this test, we can observe that Sum_ R+ is greater than Sum_R−. The p-values in Table 9 indicate if there is a significant difference between the proposed algorithm and the compared algorithm. The evidence becomes stronger as the p-value becomes smaller. Statistical significance is defined as a p-value of less than 0.05. It means there's strong evidence against the null hypothesis. From Table 9, we observe that IBEO outperformed GOA, GWO, WAO, and DA in 10 datasets, where in 10 out of these 10 datasets there is a significant difference (p-value less than 0.05). Comparing IBEO to PSO and ISSA, we can notice that IBEO has significantly outperformed them in 9 and 7 cases out

TABLE 9. Wilcoxon signed ranks test results (p-values ≥ 0.05 are underlined).

Dataset	Algorithm	No. R+	No. R-	No. ties	Sum_R+	Sum_R-	p-value
Fri_c1_1000	PSO	19	1	0	209	1	0.000
	GOA	20	0	0	210	0	0.000
	GWO	20	0	0	210	0	0.000
	WAO	20	0	0	210	0	0.000
	DA	20	0	0	210	0	0.000
	ISSA	19	1	0	209	1	0.000
Australian	PSO	14	5	1	143	47	0.053
	GOA	18	2	0	202	8	0.000
	GWO	19	1	0	208	2	0.000
	WAO	15	5	0	180	30	0.005
	DA	18	2	0	200	10	0.000
	ISSA	14	5	1	165	25	0.005
Lymphography	PSO	17	3	0	184	26	0.003
	GOA	18	2	0	199.5	10.5	0.000
	GWO	16	4	0	196.5	13.5	0.001
	WAO	16	4	0	196	14	0.001
	DA	16	4	0	169	41	0.017
	ISSA	13	7	0	137.5	72.5	0.225
Credit	PSO	20	0	0	210	0	0.000
	GOA	20	0	0	210	0	0.000
	GWO	20	0	0	210	0	0.000
	WAO	20	0	0	210	0	0.000
	DA	17	3	0	202	8	0.000
	ISSA	19	1	0	207	3	0.000
Kc1	PSO	20	0	0	210	0	0.000
	GOA	20	0	0	210	0	0.000
	GWO	20	0	0	210	0	0.000
	WAO	20	0	0	210	0	0.000
	DA	20	0	0	210	0	0.000
	ISSA	20	0	0	210	0	0.000
Parkinsons	PSO	20	0	0	210	0	0.000
	GOA	20	0	0	210	0	0.000
	GWO	20	0	0	210	0	0.000
	WAO	20	0	0	210	0	0.000
	DA	20	0	0	210	0	0.000
	ISSA	20	0	0	210	0	0.000
Dermatology	PSO	18	2	0	205	5	0.000
	GOA	20	0	0	210	0	0.000
	GWO	16	4	0	200	10	0.000
	WAO	20	0	0	210	0	0.000
	DA	18	1	1	189	1	0.000
	ISSA	14	6	0	147	63	0.117
SonarEW	PSO	20	0	0	210	0	0.000
	GOA	20	0	0	210	0	0.000
	GWO	20	0	0	210	0	0.000
	WAO	20	0	0	210	0	0.000
	DA	20	0	0	210	0	0.000
	ISSA	20	0	0	210	0	0.000
Robot-failures-lp2	PSO	15	5	0	171.5	38.5	0.013
	GOA	17	3	0	194	16	0.001
	GWO	16	3	1	176	14	0.001
	WAO	20	0	0	210	0	0.000
	DA	15	4	1	174	16	0.001
	ISSA	15	5	0	154	56	0.067
CNAE	PSO	20	0	0	210	0	0.000
	GOA	20	0	0	210	0	0.000
	GWO	20	0	0	210	0	0.000
	WAO	20	0	0	210	0	0.000
	DA	20	0	0	210	0	0.000
	ISSA	19	1	0	208	2	0.000

of 10 datasets, respectively. Finally, the p-values confirm that the results of the proposed approach are significantly different from the results of the state-of-the-art algorithms on the majority of the used datasets.

V. CONCLUSION

The comparative experiments and the aforementioned results indicate the advantage of IBEO in comparison with other optimization algorithms. Based on some measures, we evaluated the IBEO Algorithm across 25 datasets and compared it to other well-known optimization algorithms (PSO, GOA, GWO, WOA, DA, ISSA). These measures are fitness, classification accuracy, and number of selected features. This advantage in IBEO performance resulted from using the sigmoid function to deal with the binary problem. In addition, two enhancements to the original EO algorithm were added, including the OBL strategy and the LSA algorithm. OBL is applied to improve the population diversity of EO at the initialization phase, and LSA is used at the end of each EO iteration to prevent it from getting stuck in a local optimum. The kNN or SVM classifier gives high-quality solutions with the IBEO algorithm, and it can effectively learn from the training data. k-fold cross validation is an excellent decision for bypassing the over-fitting problem. In the future, the ability of the proposed algorithm to select fewer features could be increased through using new selection strategies. Hybridization can also be a good move toward improving the exploration of the algorithm. Furthermore, different classifiers such as neural networks can be used to examine the performance of the IBEO.

REFERENCES

[1] B. Ghaddar and J. Naoum-Sawaya, "High dimensional data classification and feature selection using support vector machines," *Eur. J. Oper. Res.*, vol. 265, no. 3, pp. 993–1004, Mar. 2018.

[2] C. Zhang, J. Guo, and J. Lu, "Research on classification method of high-dimensional class-imbalanced datasets based on SVM," in *Proc. IEEE 2nd Int. Conf. Data Sci. Cyberspace (DSC)*, Jun. 2017, pp. 60–67.

[3] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary ant lion approaches for feature selection," *Neurocomputing*, vol. 213, pp. 54–65, Nov. 2016.

[4] R. Jensen and Q. Shen, *Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches*, vol. 8. Hoboken, NJ, USA: Wiley, 2008.

[5] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, Aug. 2016.

[6] S. Maldonado and R. Weber, "A wrapper method for feature selection using support vector machines," *Inf. Sci.*, vol. 179, no. 13, pp. 2208–2217, 2009.

[7] H. Faris, M. M. Mafarja, A. A. Heidari, I. Aljarah, and H. Fujita, "An efficient binary Salp Swarm Algorithm with crossover scheme for feature selection problems," *Knowl.-Based Syst.*, vol. 154, pp. 43–67, Aug. 2018.

[8] B. Dasarathy, *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. Washington, DC, USA: IEEE Comput. Soc. Press, 1991.

[9] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, USA: Springer-Verlag, 2009.

[10] A. Verikas and M. Bacauskiene, "Feature selection with neural networks," *Pattern Recognit. Lett.*, vol. 23, no. 11, pp. 1323–1335, 2002.

[11] V. Vapnik and V. Vapnik, *Statistical Learning Theory*, vol. 1. New York, NY, USA: Wiley, 1998, p. 624.

[12] P. Moradi and M. Gholampour, "A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy," *Appl. Soft. Comput.*, vol. 43, pp. 117–130, Jun. 2016.

[13] A. Jovic, K. Brkic, and N. Bogunovic, "A review of feature selection methods with applications," in *Proc. 38th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, 2015, pp. 1200–1205.

[14] K. Mnich and W. R. Rudnicki, "All-relevant feature selection using multidimensional filters with exhaustive search," *Inf. Sci.*, vol. 524, pp. 277–297, Jul. 2020.

[15] K. Mani and P. Kalpana, "An efficient feature selection based on Bayes theorem, self information and sequential forward selection," *Int. J. Inf. Eng. Electron. Bus.*, vol. 8, no. 6, pp. 46–54, Nov. 2016.

[16] H. B. Nguyen, B. Xue, I. Liu, and M. Zhang, "Filter based backward elimination in wrapper based PSO for feature selection in classification," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 3111–3118.

[17] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," in *Proc. Sci. Inf. Conf.*, Aug. 2014, pp. 372–378.

[18] Y. S. Jeong, K. S. Shin, and M. K. Jeong, "An evolutionary algorithm with the partial sequential forward floating search mutation for large-scale feature selection problems," *J. Oper. Res. Soc.*, vol. 66, no. 4, pp. 529–538, Dec. 2015.

[19] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003.

[20] L. Lin and M. Gen, "Auto-tuning strategy for evolutionary algorithms: Balancing between exploration and exploitation," *Soft Comput.*, vol. 13, no. 2, pp. 157–168, Jan. 2009.

[21] A. Kaveh and N. Farhoudi, "A new optimization method: Dolphin echolocation," *Adv. Eng. Softw.*, vol. 59, no. 5, pp. 53–70, May 2013.

[22] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems," *Comput.-Aided Des.*, vol. 43, no. 3, pp. 303–315, Mar. 2011.

[23] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *J. Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, 2009.

[24] A. Kaveh and S. Talatahari, "A novel heuristic optimization method: Charged system search," *Acta Mechanica*, vol. 213, nos. 3–4, pp. 267–289, 2010.

[25] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evol. Comput.*, vol. 11, no. 1, pp. 1–18, Mar. 2003.

[26] Y. Zhang, D. Gong, Y. Hu, and W. Zhang, "Feature selection algorithm based on bare bones particle swarm optimization," *Neurocomputing*, vol. 148, pp. 150–157, Jan. 2015.

[27] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. ICNN*, vol. 4, Nov./Dec. 1995, pp. 1942–1948.

[28] S. Gu, R. Cheng, and Y. Jin, "Feature selection for high-dimensional classification using a competitive swarm optimizer," *Soft Comput.*, vol. 22, no. 3, pp. 811–822, 2018.

[29] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[30] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, Jan. 2016.

[31] Y. Pathak, K. V. Arya, and S. Tiwari, "Feature selection for image steganalysis using levy flight-based grey wolf optimization," *Multimedia Tools Appl.*, vol. 78, no. 2, pp. 1473–1494, Jan. 2019.

[32] Q. Al-Tashi, H. Rais, and S. Jadid, "Feature selection method based on grey wolf optimization for coronary artery disease classification," in *Recent Trends Data Science Soft Computing*, F. Saeed, N. Gazem, F. Mohammed, and A. Busalim, Eds. Cham, Switzerland: Springer, 2019, pp. 257–266.

[33] Q. Al-Tashi, S. J. Abdulkadir, H. M. Rais, S. Mirjalili, H. Alhussian, M. G. Ragab, and A. Alqushaibi, "Binary multi-objective grey wolf optimizer for feature selection in classification," *IEEE Access*, vol. 8, pp. 106247–106263, 2020.

[34] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: Theory and application," *Adv. Eng. Softw.*, vol. 105, pp. 30–47, Mar. 2017.

[35] H. Hichem, M. Elkamel, M. Rafik, M. T. Mesaaoud, and C. Ouahiba, "A new binary grasshopper optimization algorithm for feature selection problem," *J. King Saud Univ.-Comput. Inf. Sci.*, Nov. 2019, doi: 10.1016/j.jksuci.2019.11.007.

[36] M. Mafarja, I. Aljarah, H. Faris, A. I. Hammouri, A. M. Al-Zoubi, and S. Mirjalili, "Binary grasshopper optimisation algorithm approaches for feature selection problems," *Expert Syst. Appl.*, vol. 117, pp. 267–286, Mar. 2019.

[37] H. F. Eid, "Binary whale optimisation: An effective swarm algorithm for feature selection," *Int. J. Metaheuristics*, vol. 7, no. 1, pp. 67–79, 2018.

[38] M. M. Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neurocomputing*, vol. 260, pp. 302–312, Oct. 2017.

[39] G. I. Sayed, A. Darwish, and A. E. Hassanien, "A new chaotic whale optimization algorithm for features selection," *J. Classification*, vol. 35, no. 2, pp. 300–344, 2018.

[40] S. Mirjalili, "Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Comput. Appl.*, vol. 27, no. 4, pp. 1053–1073, 2016.

[41] S. A. Medjahed, T. A. Saadi, A. Benyettou, and M. Ouali, "Kernel-based learning and feature selection analysis for cancer diagnosis," *Appl. Soft Comput.*, vol. 51, pp. 39–48, Feb. 2017.

[42] M. M. Mafarja, D. Eleyan, I. Jaber, A. Hammouri, and S. Mirjalili, "Binary dragonfly algorithm for feature selection," in *Proc. Int. Conf. New Trends Comput. Sci. (ICTCS)*, Oct. 2017, pp. 12–17.

[43] O. K. Erol and I. Eksin, "A new optimization method: Big bang-big crunch," *Adv. Eng. Softw.*, vol. 37, no. 2, pp. 106–111, 2006.

[44] F. S. Hosseini, B. Choubin, A. Mosavi, N. Nabipour, S. Shamshirband, H. Darabi, and A. T. Haghighi, "Flash-flood hazard assessment using ensembles and Bayesian-based machine learning models: Application of the simulated annealing feature selection method," *Sci. Total Environ.*, vol. 711, Apr. 2020, Art. no. 135161.

[45] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 495–513, 2016.

[46] F. A. Hashim, E. H. Houssein, M. S. Mabrouk, W. Al-Atabany, and S. Mirjalili, "Henry gas solubility optimization: A novel physics-based algorithm," *Future Gener. Comput. Syst.*, vol. 101, pp. 646–667, Dec. 2019.

[47] H. Shareef, A. A. Ibrahim, and A. H. Mutlag, "Lightning search algorithm," *Appl. Soft Comput. J.*, vol. 36, pp. 315–333, Nov. 2015.

[48] A. Y. S. Lam and V. O. K. Li, "Chemical-reaction-inspired metaheuristic for optimization," *IEEE Trans. Evolut. Comput.*, vol. 14, no. 3, pp. 381–399, Jun. 2010.

[49] M. Taradeh, M. Mafarja, A. A. Heidari, H. Faris, I. Aljarah, S. Mirjalili, and H. Fujita, "An evolutionary gravitational search-based feature selection," *Inf. Sci.*, vol. 497, pp. 219–239, Sep. 2019.

[50] H. Abedinpourshotorban, S. M. Shamsuddin, Z. Beheshti, and D. N. A. Jawawi, "Electromagnetic field optimization: A physics-inspired metaheuristic optimization algorithm," *Swarm Evol. Comput.*, vol. 26, pp. 8–22, Feb. 2016.

[51] M. Kumar, M. Husain, N. Upreti, and D. Gupta, "Genetic algorithm: Review and application," *SSRN Electron. J.*, vol. 2, no. 2, pp. 451–454, Dec. 2010.

[52] S. Sayed, M. Nassef, A. Badr, and I. Farag, "A nested genetic algorithm for feature selection in high-dimensional cancer microarray datasets," *Expert Syst. Appl.*, vol. 121, pp. 233–243, May 2019.

[53] H. Chen, W. Jiang, C. Li, and R. Li, "A heuristic feature selection approach for text categorization by using chaos optimization and genetic algorithm," *Math. Problems Eng.*, vol. 2013, Dec. 2013, Art. no. 524017.

[54] Y. Zhang, D.-W. Gong, X.-Z. Gao, T. Tian, and X.-Y. Sun, "Binary differential evolution with self-learning for multi-objective feature selection," *Inf. Sci.*, vol. 507, pp. 67–85, Jan. 2020.

[55] P. Agrawal, T. Ganesh, and A. W. Mohamed, "A novel binary gaining–sharing knowledge-based optimization algorithm for feature selection," *Neural Comput. Appl.*, vol. 33, pp. 5989–6008, Oct. 2020.

[56] H. C. Kuo and C. H. Lin, "Cultural evolution algorithm for global optimizations and its applications," *J. Appl. Res. Technol.*, vol. 11, no. 4, pp. 510–522, 2013.

[57] S. Hosseini and A. A. Khaled, "A survey on the imperialist competitive algorithm metaheuristic: Implementation in engineering domain and directions for future research," *Appl. Soft Comput.*, vol. 24, pp. 1078–1094, Nov. 2014.

[58] R. Moghdani and K. Salimifard, "Volleyball premier league algorithm," *Appl. Soft. Comput.*, vol. 64, pp. 161–185, Mar. 2018.

[59] A. Kaveh, S. Talatahari, and N. Khodadadi, "Stochastic paint optimizer: Theory and application in civil engineering," *Eng. Comput.*, vol. 13, pp. 1–32, Oct. 2020.

[60] S. Talatahari and M. Azizi, "Optimal design of real-size building structures using quantum-behaved developed swarm optimizer," *Struct. Des. Tall Buildings*, vol. 29, no. 11, p. e1747, Aug. 2020.

[61] R. Sheikholeslami and S. Talatahari, "Developed swarm optimizer: A new method for sizing optimization of water distribution systems," *J. Comput. Civil Eng.*, vol. 30, no. 5, Sep. 2016, Art. no. 04016005.

[62] S. Talatahari and M. Azizi, "Chaos game optimization: A novel metaheuristic algorithm," *Artif. Intell. Rev.*, vol. 54, pp. 917–1004, Feb. 2021.

[63] S. Talatahari and M. Azizi, "Optimization of constrained mathematical and engineering design problems using chaos game optimization," *Comput. Ind. Eng.*, vol. 145, Jul. 2020, Art. no. 106560.

[64] S. Arora, H. Singh, M. Sharma, S. Sharma, and P. Anand, "A new hybrid algorithm based on grey wolf optimization and crow search algorithm for unconstrained function optimization and feature selection," *IEEE Access*, vol. 7, pp. 26343–26361, 2019.

[65] Q. Al-Tashi, S. J. A. Kadir, H. M. Rais, S. Mirjalili, and H. Alhussian, "Binary optimization using hybrid grey wolf optimization for feature selection," *IEEE Access*, vol. 7, pp. 39496–39508, 2019.

[66] H. Jia, J. Li, W. Song, X. Peng, C. Lang, and Y. Li, "Spotted hyena optimization algorithm with simulated annealing for feature selection," *IEEE Access*, vol. 7, pp. 71943–71962, 2019.

[67] S. Ahmed, K. K. Ghosh, P. K. Singh, Z. W. Geem, and R. Sarkar, "Hybrid of harmony search algorithm and ring theory-based evolutionary algorithm for feature selection," *IEEE Access*, vol. 8, pp. 102629–102645, 2020.

[68] K. H. Sheikh, S. Ahmed, K. Mukhopadhyay, P. K. Singh, J. H. Yoon, Z. W. Geem, and R. Sarkar, "EHHM: Electrical harmony based hybrid meta-heuristic for feature selection," *IEEE Access*, vol. 8, pp. 158125–158141, 2020.

[69] Z. M. Elgamal, N. B. M. Yasin, M. Tubishat, M. Alswaitti, and S. Mirjalili, "An improved Harris Hawks optimization algorithm with simulated annealing for feature selection in the medical field," *IEEE Access*, vol. 8, pp. 186638–186652, 2020.

[70] S. Sapre and S. Mini, "Opposition-based moth flame optimization with Cauchy mutation and evolutionary boundary constraint handling for global optimization," *Soft Comput.*, vol. 23, no. 15, pp. 6023–6041, Aug. 2019.

[71] Y. Zhou, J.-K. Hao, and B. Duval, "Opposition-based memetic search for the maximum diversity problem," *IEEE Trans. Evol. Comput.*, vol. 21, no. 5, pp. 731–745, Oct. 2017.

[72] H. R. Tizhoosh, "Opposition-based reinforcement learning," *J. Adv. Comput. Intell. Intell. Inform.*, vol. 10, no. 3, pp. 1–9, 2006.

[73] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition versus randomness in soft computing techniques," *Appl. Soft. Comput.*, vol. 8, no. 2, pp. 906–918, 2008.

[74] N. Rojas-Morales, M.-C. R. Rojas, and E. M. Ureta, "A survey and classification of opposition-based metaheuristics," *Comput. Ind. Eng.*, vol. 110, pp. 424–435, Aug. 2017.

[75] M. Aladeemy, L. Adwan, A. Booth, M. T. Khasawneh, and S. Poranki, "New feature selection methods based on opposition-based learning and self-adaptive cohort intelligence for predicting patient no-shows," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105866.

[76] A. Kaveh, N. Khodadadi, B. F. Azar, and S. Talatahari, "Optimal design of large-scale frames with an advanced charged system search algorithm using box-shaped sections," *Eng. Comput.*, vol. 2020, pp. 1–21, Feb. 2020.

[77] M. Basu, "Quasi-oppositional differential evolution for optimal reactive power dispatch," *Int. J. Electr. Power Energy Syst.*, vol. 78, pp. 29–40, Jun. 2016.

[78] M. A. Ahandani, "Opposition-based learning in the shuffled bidirectional differential evolution algorithm," *Swarm Evol. Comput.*, vol. 26, pp. 64–85, Feb. 2016.

[79] A. Banerjee, V. Mukherjee, and S. P. Ghoshal, "An opposition-based harmony search algorithm for engineering optimization problems," *Ain Shams Eng. J.*, vol. 5, no. 1, pp. 85–101, 2014.

[80] M. A. Elaziz, D. Oliva, and S. Xiong, "An improved opposition-based sine cosine algorithm for global optimization," *Expert Sys. Appl.*, vol. 90, pp. 484–500, Dec. 2017.

[81] D. Bairathi and D. Gopalani, "Opposition based Salp Swarm Algorithm for numerical optimization," in *Intelligent Systems Design and Applications*, A. Abraham, A. K. Cherukuri, P. Melin, and N. Gandhi, Eds. Cham, Switzerland: Springer, 2020, pp. 821–831.

[82] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowl.-Based Syst.*, vol. 191, Mar. 2020, Art. no. 105190.

[83] K. K. Ghosh, R. Guha, S. K. Bera, R. Sarkar, and S. Mirjalili, "BEO: Binary equilibrium optimizer combined with simulated annealing for feature selection," Res. Square, Durham, NC, USA, Tech. Rep., 2020, doi: 10.21203/rs.3.rs-28683/v1.

[84] Y. Gao, Y. Zhou, and Q. Luo, "An efficient binary equilibrium optimizer algorithm for feature selection," *IEEE Access*, vol. 8, pp. 140936–140963, 2020.

[85] M. Beşkirli, !‘. Koç, H. Haklı, and H. Kodaz, "A new optimization algorithm for solving wind turbine placement problem: Binary artificial algae algorithm," *Renew. Energy*, vol. 121, pp. 301–308, Jun. 2018.

[86] U. Kamath and K. Choppella, *Mastering Java Machine Learning*. Birmingham, U.K.: Packt, 2017.

[87] F. Pernkopf, "Bayesian network classifiers versus selective k-NN classifier," *Pattern Recognit.*, vol. 38, no. 1, pp. 1–10, Jan. 2005.

[88] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, Mar. 2011.

DINA A. ELMANAKHLY received the M.Sc. degree in computer science from Suez Canal University, Egypt. Her current research interests include human–computer interactions and multi-objective optimization.

MOHAMED MOSTAFA SALEH received the Ph.D. degree in system dynamics from the University of Bergen, Norway. He is currently the Head of the Operations Research and Decision Support Department, Cairo University. His research interests include data mining, machine learning, operations management, optimization, scheduling, and modeling.

ESSAM A. RASHED (Senior Member, IEEE) received the B.Sc. degree in scientific computing and the M.Sc. degree in computer science from Suez Canal University, Ismailia, Egypt, in 1998 and 2002, respectively, and the Ph.D. degree in computer science from the University of Tsukuba, Tsukuba, Japan, in 2010. From 2010 to 2012, he was a Research Fellow with Japan Society for the Promotion of Science (JSPS), University of Tsukuba. He has served as an Assistant Professor for the Department of Mathematics, Faculty of Science, Suez Canal University, from 2012 to 2016. Since then, he has been promoted to an Associate Professor with Suez Canal University and The British University in Egypt (on Secondment). He is currently a Research Associate Professor with Nagoya Institute of Technology, Japan. He has also served as a Principal Investigator (PI) for several projects funded from the Science and Technology Development Fund (STDF), Egypt. His research interests include medical image processing, data analysis and visualization, deep learning, and pattern recognition. He was a recipient of the Egyptian National Doctoral Scholarship, in 2006, the JSPS Postdoctoral Fellowship, in 2010, the JAMIT Best Presentation Award, in 2008 and 2012, and the Chairman Award, Department of Computer Science, University of Tsukuba, in 2010. He is also an Associate Editor of IEEE Access.