computers

MDPI

Article

B-MFO: A Binary Moth-Flame Optimization for Feature Selection from Medical Datasets

Mohammad H. Nadimi-Shahraki [1,2,*], Mahdis Banaie-Dezfouli [1,2], Hoda Zamani [1,2], Shokooh Taghian [1,2] and Seyedali Mirjalili [3,4]

[1] Faculty of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad 8514143131, Iran; m.dezfouli@sco.iaun.ac.ir (M.B.-D.); hoda_zamani@sco.iaun.ac.ir (H.Z.); sh.taghian@sco.iaun.ac.ir (S.T.)
[2] Big Data Research Center, Najafabad Branch, Islamic Azad University, Najafabad 8514143131, Iran
[3] Centre for Artificial Intelligence Research and Optimisation, Torrens University Australia, Adelaide 5000, Australia; ali.mirjalili@torrens.edu.au
[4] Yonsei Frontier Lab, Yonsei University, Seoul 03722, Korea
* Correspondence: nadimi@iaun.ac.ir

Abstract: Advancements in medical technology have created numerous large datasets including many features. Usually, all captured features are not necessary, and there are redundant and irrelevant features, which reduce the performance of algorithms. To tackle this challenge, many metaheuristic algorithms are used to select effective features. However, most of them are not effective and scalable enough to select effective features from large medical datasets as well as small ones. Therefore, in this paper, a binary moth-flame optimization (B-MFO) is proposed to select effective features from small and large medical datasets. Three categories of B-MFO were developed using S-shaped, V-shaped, and U-shaped transfer functions to convert the canonical MFO from continuous to binary. These categories of B-MFO were evaluated on seven medical datasets and the results were compared with four well-known binary metaheuristic optimization algorithms: BPSO, bGWO, BDA, and BSSA. In addition, the convergence behavior of the B-MFO and comparative algorithms were assessed, and the results were statistically analyzed using the Friedman test. The experimental results demonstrate a superior performance of B-MFO in solving the feature selection problem for different medical datasets compared to other comparative algorithms.

Keywords: optimization; binary metaheuristic algorithms; swarm intelligence algorithms; feature selection; medical datasets; transfer function

Citation: Nadimi-Shahraki, M.H.; Banaie-Dezfouli, M.; Zamani, H.; Taghian, S.; Mirjalili, S. B-MFO: A Binary Moth-Flame Optimization for Feature Selection from Medical Datasets. *Computers* **2021**, *10*, 136. https://doi.org/10.3390/computers10110136

Academic Editors: Ivanoe De Falco, Antonino Galletta, Giovanna Sannino and Antonio Celesti

Received: 20 September 2021
Accepted: 19 October 2021
Published: 25 October 2021

1. Introduction

Nowadays, with advances in science and medical technology, numerous large medical datasets including many features have been created, which also contain redundant and irrelevant features. Data-driven decision making in high-risk diseases such as heart diseases [1] is a significant trend in which many data mining and machine learning methods are introduced [2]. Since medical data are obtained from multiple sources, all captured features are not necessary and some of them are irrelevant and redundant, which may reduce algorithms' performance in the data-driven decision-maker software. The irrelevant and redundant data can be removed since they are useless in improving the accuracy's classification, because the irrelevant data have a weak correlation with class and the redundant data have a strong correlation with one or more features. For instance, the FSBRR algorithm [3] removes the feature of radius in the Breast Cancer Wisconsin Dataset as a redundant feature because its correlation is very high with feature of smoothness. Feature selection can address this problem, through which a subset of the relevant and

effective features must be found. Feature selection is used in a variety of real-world applications such as disease diagnosis [4,5], email spam detection [6], text clustering [7,8], and human activity recognition [9].

Based on the strategy used for selecting features, the feature selection algorithms can be categorized into three methods [10]: filter-based, wrapper-based, and hybrid methods. The filter-based methods analyze features based on intrinsic properties of the data, without using the classification algorithm [11], whereas wrapper-based methods use classifiers to assess possible solutions [12]. Hybrid methods combine the benefits of both filter-based and wrapper-based approaches. Although the wrapper-based methods are computationally expensive and their performance depends on the utilized learning algorithm, they are usually more accurate than the other two categories [13]. The wrapper-based methods use different search approaches such as exhaustive, random, greedy, heuristic, and metaheuristic [14], which, except for the last search approach, are impractical to select effective features from medium and large datasets [15]. Thus, a wide range of metaheuristic optimization algorithms is proposed to solve the feature selection problems for applications with large datasets such as medicine [16].

Metaheuristic optimization algorithms are mostly inspired by nature, and can be classified into three categories: evolutionary, physics-based, and swarm intelligence. The simplicity in development and sufficient results of the swarm intelligence (SI) algorithms for a variety of problems have made some of them very attractive and popular, such as particle swarm optimization (PSO) [17], grey wolf optimizer (GWO) [18], whale optimization algorithm (WOA) [19], moth-flame optimization (MFO) [20], and aquila optimizer (AO) [21]. Since SI algorithms mimic the behavioral model of insects, aquatic animals, terrestrial animals, and birds, these algorithms can share information between their swarms that enhances their robustness [22]. Local optima trapping, premature convergence, and unbalanced search strategy may all be disadvantages of these algorithms, despite their benefits [23,24]. Hence, many improvements of these algorithms have been introduced thus far, including optimal control strategies [25], chaotic whale optimization algorithm (CWOA) [26], grasshopper optimization algorithm (GOA) based on the opposition-based learning (OBLGOA) [27], improved grey wolf optimizer (I-GWO) [28], disruption bare-bones particle swarm optimization (DBPSO) [29], improved krill herd (IKH) [30], particle swarm optimization with backtracking search optimization algorithm (PSOBSA) [31], and representative-based grey wolf optimizer (R-GWO) [32].

Since feature selection is an NP-complete problem [33], SI algorithms are widely used to solve this problem. Many researchers have adapted different SI algorithms for converting from the continuous form to binary, such as wrapper-based binary sine cosine algorithm (WBSCA) [34], binary grasshopper optimization algorithm (BGOA) [35], binary butterfly algorithm (BBA) [36], efficient binary symbiotic organisms search (EBSOS) [37], binary grey wolf optimizer with support vector machine (GWOSVM) [38], and island binary moth-flame optimization (IsBMFO) [39]. However, most binary SI algorithms are not effective or scalable enough to select effective features from large datasets as well as small ones.

Therefore, in this study, a binary moth-flame optimization (B-MFO) is proposed to solve the feature selection problem. The canonical MFO was introduced by Mirjalili [20], which was inspired by the transverse orientation mechanism of the moths in the night around artificial lights. There have been many variants of MFO developed, such as EMFO [40], IMFO [41], CLSGMFO [42], and improved MFO [43], given its simplicity, weaknesses, and applications. Several researchers have applied S-shaped and V-shaped transfer functions to convert the continuous MFO to binary. In this study, in addition to the S-shape and V-shaped transfer functions, another transfer function named U-shaped was adapted, which is a novel transfer function with multiple alterable parameters for solving feature selection problems. Each category contains four versions of transfer functions. Therefore, twelve versions of B-MFO were introduced in three categories of transfer functions. Then, they were evaluated by seven medical datasets: Pima, Lymphography, Breast-

WBDC, PenglungEw, Parkinson, Colon, and Leukemia. In addition, the winner versions of B-MFO were compared with the best results gained by four well-known binary metaheuristic optimization algorithms: BPSO [44], bGWO [45], BDA [46], and BSSA [47]. The convergence behavior of the winner versions of B-MFO and comparative algorithms was evaluated and visualized. Finally, the results were statistically analyzed by the Friedman test.

In the rest of this study, Section 2 discusses the related works. Section 3 describes the canonical MFO algorithm. Then, the proposed B-MFO is presented and evaluated in Sections 4 and 5. Finally, the conclusion and future works are explained in Section 6.

2. Related Work

There are many different discrete problems such as feature selection [48,49], tour planning [50], complex systems [51], and traveling salesman problems [52] that must be solved with discrete optimization algorithms [53]. To solve feature selection problems, wrapper-based methods widely apply discrete metaheuristic optimization algorithms as search strategies to find effective feature subsets [47,54–57]. Since the majority of metaheuristic optimization algorithms such as DA [58], SSA [59], HGSO [60], FFA [61], MTDE [62], QANA [63], and AO [21] have been proposed to solve continuous problems such as engineering [64–68], cloud computing [69], and rail-car fleet sizing [70], they should be converted into binary algorithms for using in wrapper-based methods and solving discrete problems. The continuous algorithm can be converted to a binary form in a variety of ways [71]. The JayaX [72] and BitABC [73] use the logical operators for changing to the binary form. Another way is using the transfer function (TF), which converts the continuous search space to the binary one in which the search agents can shift to nearer or farther corners of a hypercube by flipping various numbers of bits [44]. Thus, transfer functions apply a mapping function to gain the probability of changing a solution from 0 to 1 or vice versa.

Many transfer functions were introduced such as S-shaped [44,74], V-shaped [74,75], and U-shaped [76] to convert the continuous metaheuristic optimization algorithms to binary ones. The binary particle swarm optimization (BPSO) [44] was introduced by Kennedy and Eberhart, which applied a sigmoid function to solve various discrete optimization problems [77–79]. Yuan et al. [80] proposed a new improved binary PSO (IBPSO) in which the BPSO is combined with the lambda-iteration method to solve the unit commitment problem. The BPSO has been applied for various problems such as text clustering [81,82], text feature selection [83], and disease diagnosis [84–86].

Binary grey wolf optimizer (bGWO) is another wrapper method for feature selection which was proposed by Emary et al. [45]. The binary version of GWO was performed using the sigmoid transfer function and was utilized to fix the feature selection problems and large-scale unit commitment [87,88], and text classification [89]. To enhance the solution quality of transfer functions, Hu et al. in [90] introduced new transfer functions and improved them based on analysis parameters of GWO. Al-tashi et al. [87] proposed a new hybrid optimization algorithm named (BPSOGWO) to find the best feature subset.

Zamani et al. [91] proposed a new metaheuristic algorithm named feature selection based on whale optimization algorithm (FSWOA) to reduce the dimensionality of medical datasets. Hussien et al. proposed two binary variants of WOA (bWOA) [92,93] based on V-shaped and S-shaped to use for dimensionality reduction and classification problems. The binary WOA (BWOA) [94] was suggested by Reddy et al. for solving the PBUC problem, which mapped the continuous WOA to the binary one through various transfer functions.

The binary dragonfly algorithm (BDA) [95] was proposed by Mafarja to solve discrete problems. The BDFA [96] was proposed by Sawhney et al. which incorporates a penalty function for optimal feature selection. Although BDA has good exploitation ability, it suffers from being trapped in local optima. Thus, a wrapper-based approach named hyper learning binary dragonfly algorithm (HLBDA) [97] was developed by Too et al. to solve

the feature selection problem. The HLBDA used the hyper learning strategy to learn from the personal and global best solutions during the search process.

Faris et al. employed the binary salp swarm algorithm (BSSA) [47] in the wrapper feature selection method. Ibrahim et al. proposed a hybrid optimization method for the feature selection problem which combines the slap swarm algorithm with the particle swarm optimization (SSAPSO) [98]. The chaotic binary salp swarm algorithm (CBSSA) [99] was introduced by Meraihi et al. to solve the graph coloring problem. The CBSSA applies a logistic map to replace the random variables used in the SSA, which causes it to avoid the stagnation to local optima and improves exploration and exploitation. A time-varying hierarchal BSSA (TVBSSA) was proposed in [15] by Faris et al. to design an improved wrapper feature selection method, combined with the RWN classifier.

3. The Canonical Moth-Flame Optimization

Moth-flame optimization (MFO) [20] is a nature-inspired algorithm that imitates the transverse orientation mechanism of moths in the night around artificial lights. This mechanism applies to navigation, and forces moths to fly in a straight line and maintain a constant angle with the light. MFO's mathematical model assumes that the moths' position in the search space corresponds to the candidate solutions, which are represented in a matrix, and the corresponding fitness of the moths are stored in an array. In addition, a flame matrix shows the best positions obtained by the moths so far, and an array is used to indicate the corresponding fitness of the best positions. To find the best result, moths search around their corresponding flame and update their positions; therefore, moths never lose their best position. Equation (1) shows the position updating of each moth relative to the corresponding flame.

$$M_i = S(M_i, F_j) \tag{1}$$

where S is the spiral function, and M_i and F_j represent the *i-th* moth and the *j-th* flame, respectively. The main update mechanism is a logarithmic spiral, which is defined by Equation (2):

$$S(M_i, F_j) = D_i . e^{bt} . \cos(2\pi t) + F_j \tag{2}$$

where D_i is the distance between the *i-th* moth and the *j-th* flame, which is computed by Equation (3), and b is a constant value for defining the shape of the logarithmic spiral. The parameter t is a random number in the range $[-r, 1]$, in which r is a convergence factor and linearly decreases from −1 to −2 during the course of iterations.

$$D_i = |M_i - F_j| \tag{3}$$

To avoid trapping in local optima, each moth updates its position using one flame. In each iteration, the list of flames is updated and sorted based on their fitness values. The first moth updates its position according to the best flame and the last moth updates its position according to the worst flame. In addition, to increase the exploitation of the best promising solutions, the number of flames is reduced in the course of iterations by an adaptive mechanism, which is shown in Equation (4):

$$flame_{No} = round\ (N - iter \cdot (N - 1)/MaxIter) \tag{4}$$

where N indicates the maximum number of moths, and *iter* and *MaxIter* are the current and maximum number of iterations, respectively.

4. Binary Moth-Flame Optimization (B-MFO)

In this study, three different categories of S-shaped, V-shaped, and U-shaped transfer functions are applied to convert the MFO algorithm from continuous to binary for solving the feature selection problem. First, in Section 4.1, these different categories of transfer functions and how to apply them to develop different variants of B-MFO are described in detail accompanied by their flowchart and pseudo-code. Then, in Section 4.2, solving feature selection problem using B-MFO is explained.

4.1. Developing Different Variants of B-MFO

4.1.1. B-MFO Using S-Shaped Transfer Function

The sigmoid (S-shaped) function shown in Equation (5) is a usual transfer function named S_2 [100], which was originally introduced for developing the binary PSO (BPSO) [44].

$$TF_s(v_i^d(t+1)) = 1/(1 + exp^{-v_i^d(t)}) \quad (5)$$

where $v_i^d(t)$ is the *i-th* search agent's velocity in dimension d at iteration t. The TF_s converts the velocity to a probability value and the next position $x_i^d(t+1)$ is obtained with the probability value of its velocity as given in Equation (6), where r is a random value between 0 and 1.

$$x_i^d(t+1) = \begin{cases} 0 & If\ r < TF_s(v_i^d(t+1)) \\ 1 & If\ r \geq TF_s(v_i^d(t+1)) \end{cases} \quad (6)$$

According to Equation (6), the position updating of each search agent is computed by the current velocity and the previous position. In some binary metaheuristic algorithms such as BPSO [44] and BGSA [101], the velocity is used in transfer functions to calculate the probability value of changing the position. In some other algorithms such as bGWO [45] and BMFO [102], transfer functions apply the updated position of each search agent to calculate the probability value. In addition to the S_2 function introduced in Equation (5), three variants of the S-shaped function named S_1, S_3, and S_4 [74] are developed by manipulating the coefficient of the velocity value in Equation (5). All variants of the S-shaped transfer function are shown in Table 1 and visualized in Figure 1, which shows that if the slope of the S-shaped transfer function increases, the probability value of changing the position value increases. Thus, among of S-shaped functions, the S_1 obtains the highest probability and the S_4 provides the lowest value for the same velocity, which can affect the position updating of search agents and finding the optimum solution. In addition to the advantages of S-shaped, this category of transfer functions has a shortcoming in those metaheuristic algorithms that search agents are updated considering by their velocity value. The zero value of velocity is converted to one or zero with a probability of 0.5, while the search agents should not be moved with the zero value of velocity [103]. Several researchers tried to resolve this shortcoming, but they could not avoid trapping into local optima.

Table 1. The variants of S-shaped, V-shaped, and U-shaped transfer functions.

S-Shaped Transfer Function		V-Shaped Transfer Function		U-Shaped Transfer Function	
No.	**Transfer Function**	**No.**	**Transfer Function**	**No.**	**Transfer Function**
S_1	$TF_s(x) = 1/(1 + exp^{-2x})$	V_1	$TF_v(x) = \left\|erf\left(\frac{\sqrt{\pi}}{2}x\right)\right\| = \left\|\left(\frac{\sqrt{\pi}}{2}\int_0^{\frac{\sqrt{\pi}}{2}x} e^{-t^2}dt\right)\right\|$	U_1	$TF_u(x) = \alpha\|x^{1.5}\|$
S_2	$TF_s(x) = 1/(1 + exp^{-x})$	V_2	$TF_v(x) = \|\tanh(x)\|$	U_2	$TF_u(x) = \alpha\|x^2\|$
S_3	$TF_s(x) = 1/(1 + exp^{(-x/2)})$	V_3	$TF_v(x) = \left\|(x)/\sqrt{1+x^2}\right\|$	U_3	$TF_u(x) = \alpha\|x^3\|$
S_4	$TF_s(x) = 1/(1 + exp^{(-x/3)})$	V_4	$TF_v(x) = \left\|\frac{2}{\pi}\arctan\left(\frac{\pi}{2}x\right)\right\|$	U_4	$TF_u(x) = \alpha\|x^4\|$

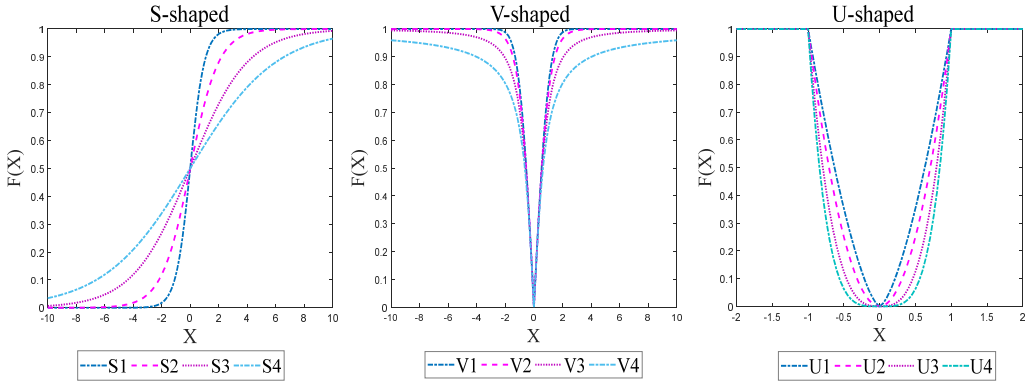

Figure 1. S-shaped, V-shaped, and U-shaped transfer functions.

4.1.2. B-MFO Using V-Shaped Transfer Function

The hyperbolic (V-shaped) function [104] named V_2 was presented for developing BGSA [101] which has new position updating as shown in Equation (7).

$$TF_v(v_i^d(t+1)) = \left|tanh\left(v_i^d(t)\right)\right| \tag{7}$$

where $v_i^d(t)$ shows the *i-th* search agent's velocity in dimension d at iteration t. Since the V-shaped function is different from the S-shaped function, this function is updated with new rules that are shown in Equation (8).

$$x_i^d(t+1) = \begin{cases} \neg(x_i^d(t)) & If\ r < TF_v(v_i^d(t+1)) \\ x_i^d(t) & If\ r \geq TF_v(v_i^d(t+1)) \end{cases} \tag{8}$$

where $x_i^d(t)$ indicates the *i-th* search agent's position in dimension d at iteration t and $\neg(x_i^d(t))$ represents the complement of $(x_i^d(t))$. In addition, r is a random value between 0 and 1. If the velocity is low, the TF_v encourages the search agents to stay in their current positions; otherwise, if the velocity is high, the search agents switch to complement. In addition to the function introduced in Equation (7), three variants of V-shaped function named V_1, V_3, and V_4 are introduced [74], which are shown in Table 1 and Figure 1. It can be seen that V_1 provides a higher probability than V_2, V_3, and V_4 for the same velocity, which can affect the position updating of search agents and finding the optimum solution. The V-shaped transfer function was proposed to tackle some shortcomings of the S-shaped. Although this transfer function can solve the problem of metaheuristic algorithms with the velocity by zero value, they still suffer from falling into local optima. If in an

iteration, the velocity of a search agent is low, in the next iteration, the search agent remains the same with high probability [103]. In this study, in addition to the transfer functions mentioned so far, we utilized a novel U-shaped transfer function to convert the continuous MFO to binary form.

4.1.3. B-MFO Using U-Shaped Transfer Function

The U-shaped transfer function [76] was proposed with two control parameters α and β that define the slope and width of the U-shaped function's basin, respectively. Equations (9) and (10) indicate the U-shaped function.

$$TF_u\left(v_i^d(t+1)\right) = \alpha\left|(v_i^d(t))^\beta\right| \quad \alpha=1,\ \beta=1.5,\ 2,\ 3,\ 4 \tag{9}$$

$$x_i^d(t+1) = \begin{cases} \neg(x_i^d(t)) & \text{If } r < TF_u(v_i^d(t+1)) \\ x_i^d(t) & \text{If } r \geq TF_u(v_i^d(t+1)) \end{cases} \tag{10}$$

where $v_i^d(t)$ shows the *i-th* search agent's velocity in dimension d at iteration t, and the r is a uniform random number between 0 and 1. We used the U-shaped transfer function accompanied by two main conditions shown in Equations (11) and (12) in which the lower and upper bounds are limited by 1.

$$\lim_{v_i \to \infty} U\left(v_i^d(t)\right) = 1 \tag{11}$$

$$\lim_{v_i \to -\infty} U\left(v_i^d(t)\right) = 1 \tag{12}$$

The variants of the U-shaped transfer function named U_1, U_2, U_3, and U_4 were introduced using a different control parameter β [76], shown in Table 1 and Figure 1. In the initial iterations, the exploration is an important step to seek the whole search space and after switching from exploration to exploitation, in the final iterations, the exploitation step is essential to find better solutions. The U-shaped transfer functions with different shapes have different exploratory and exploitative behaviors. As illustrated in Figure 1, the U-shaped is comparable to V-shaped; however, variants of U-shaped have a higher exploration in contrast to variants of V-shaped. The U1 function and variants of V-shaped intersect around +0.7 and −0.7; before this point, the exploration of V-shaped is higher, and after it, variants of U-shaped display a higher exploration. Therefore, U-shaped transfer functions have sufficient potential to outperform the other transfer functions.

To map the continuous MFO to the binary one, each search agents' dimension obtained by Equation (2) is converted to a probability value in the range [0,1] using all variants of TF_s, TF_v, and TF_u. Therefore, we adapted twelve different transfer functions from the three categories, S-shaped, V-shaped, and U-shaped. By using these transfer functions, each search agents' position is mapped to the probability value using Equations (5), (7), and (9). Finally, this probability value is updated using Equations (6), (8), and (10), and a new search agent's position is created. Thus, we compare twelve versions of proposed B-MFO to find the suitable version. It is important to apply a proper transfer function, since converting a continuous search space to a binary one is significant in the results of classifier of feature selection problems. Algorithm 1 and Figure 2 show the pseudo-code and the flowchart of B-MFO, respectively. The time complexity of B-MFO is *O(NDT)* where *N*, *D*, and *T* signify the population size, dimension, and maximum number of iterations, respectively.

4.2. B-MFO for Solving Feature Selection Problem

The feature selection problem is to select an optimum subset of the relevant and effective features to construct a more accurate data model. To formulate the feature selection problem, a vector of one or zero as a subset of features is defined by using a transfer function, which obtains probability values to change elements in the vector that can be 0 (not selected) or 1 (selected). The length of the vector is equal to the dimensions of the dataset. In addition, a fitness function is determined to evaluate the subset of features. The problem of feature selection is referred to as a multi-objective optimization problem [105,106] since it usually aims to minimize the number of selected features and maximize the data model accuracy. As shown in Equation (13), the objectives are represented in a fitness function, where *CE* shows the classification error. N_{sf} and N_{tf} are the number of selected features and total features of the dataset, respectively. η and λ *(1- η)* demonstrate the significance of classification quality and feature reduction, respectively [46].

$$Fitness = \eta . CE + \lambda \frac{N_{sf}}{N_{tf}} \quad (13)$$

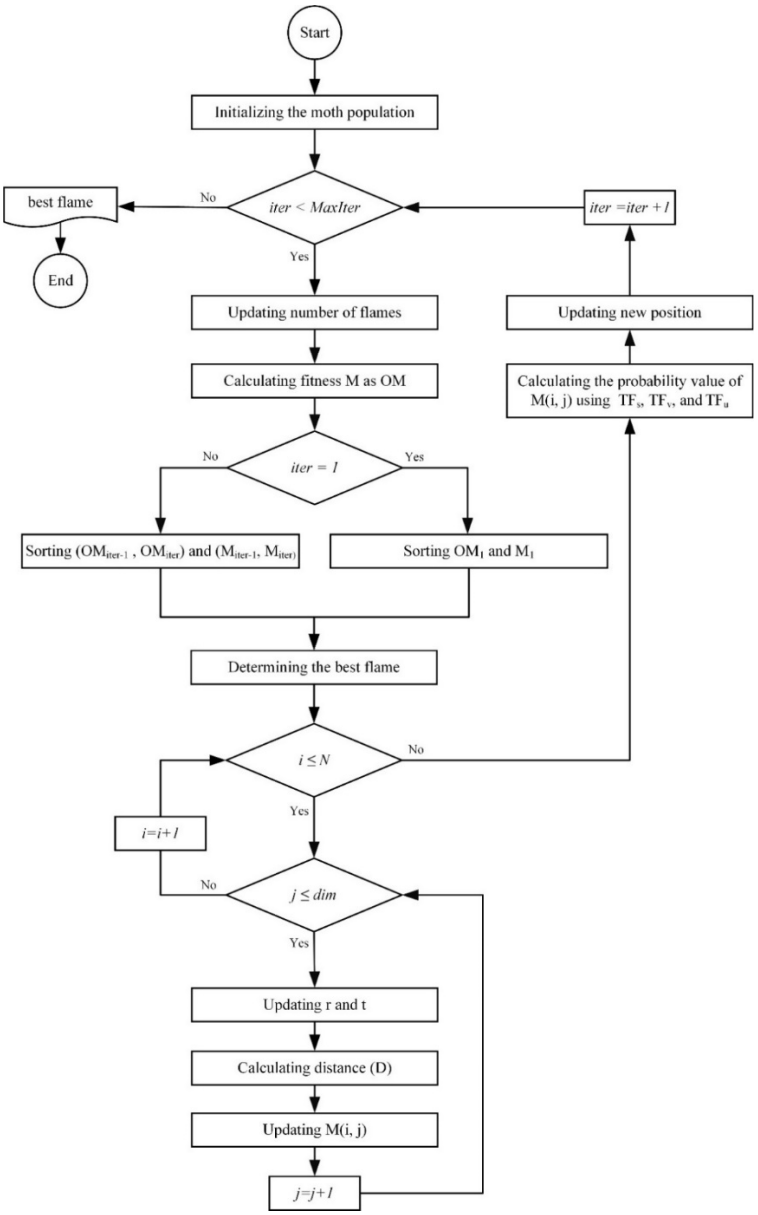

Figure 2. The flowchart of B-MFO.

Algorithm 1. The pseudo-code of B-MFO.

Algorithm of binary moth-flame optimization (B-MFO)

Input: N (Population size), MaxIter (Maximum number of iterations), dim (the number of dimensions).
Output: The global optimum (best flame).

Procedure B-MFO
Initializing the moth population.
While iter < MaxIter
    Updating number of flames ($flame_{No}$) using Equation (4).
    Calculating the fitness function M as OM.
    If iter==1
        OF = sort (OM_1).
        F = sort (M_1).
    Else
        OF = sort (OM_{iter-1}, OM_{iter}).
        F = sort (M_{iter-1}, M_{iter}).
    End if
    Determining the best flame.
    For i = 1: N
      For j = 1: dim
            Updating r and t.
            Calculating D using Equation (3).
            Updating M(i, j) using Equation (1) and Equation (2).
      End for
    End for
    Calculating the probability value of M(i, j) using TF_s in Equation (5), TF_v in Equation (7), and TF_u in Equation (9).
    Updating new position using Equation (6), Equation (8), and Equation (10).
    iter = iter +1.
End while
Return the global optimum (best flame).
End Procedure

5. Experimental Assessment

To execute the proposed B-MFO and other comparative algorithms in a fair condition, all algorithms were implemented using the MATLAB 2018a platform. They were conducted on Windows 10 with a processor Intel Core i7-6500U CPU (2.50 GHz) with 8 GB on main memory. In addition, the population size (*N*) and the maximum number of iterations (*MaxIter*) were considered as 20 and 300, respectively, and each algorithm was run 30 times. The proposed B-MFO is experimentally evaluated on all transfer functions of the three categories of S-shaped [44], V-shaped [101], and U-shaped [76] to solve the feature selection problem. The experimental results were compared with the best result gained by four well-known binary metaheuristic algorithms: BPSO [44], bGWO [45], BDA [46], and BSSA [47]. The parameters of BPSO and bGWO were set to be the same as original studies such as that for BPSO w= *[0.9 to 0.4]* and C_1=C_2=*2*, and for bGWO a= *[2 to 0]*. In addition, the other algorithms did not need any parameter setting.

5.1. Data Description

In this study, seven medical datasets [107,108] were applied to evaluate the B-MFO and comparative algorithms in the feature selection problem. Table 2 shows the details of datasets in terms of the number of features, number of samples, and size that is considered large if the number of features is more than 100. In our evaluation, a k-nearest neighbor (k-NN) classifier with a Euclidean distance metric and $k_{neighbor} = 5$ [56] was applied as a fitness function to assess the quality of selected features subsets. To reduce the overfitting problem, the k-fold cross-validation with $k_{fold} = 10$ was utilized, which divides datasets into k folds, and the classifier used the k-1 folds for training data and the 1 fold for test data. This process was repeated for each of the k folds, and all folds were selected once as test data.

Table 2. The datasets' descriptions.

No.	Medical Datasets	No. Features	No. Samples	Size
1.	Pima	8	768	Small
2.	Lymphography	18	148	Small
3.	Breast-WDBC	30	569	Small
4.	PenglungEW	325	73	Large
5.	Parkinson	754	756	Large
6.	Colon	2000	62	Large
7.	Leukemia	7129	72	Large

5.2. Evaluation Criteria

The proposed B-MFO was compared with comparative algorithms using various metrics consisting of average accuracy, the standard deviation of accuracy, average fitness, the standard deviation of fitness, and the average number of selected features. Moreover, the performance of the k-NN classifier was measured using sensitivity and specificity derived from the confusion matrix, which includes the information about actual and predicted classifications given by the classifier. The sensitivity is a metric that evaluates the ability of the model to predict true positives, and the specificity is the metric that measures the ability of the model to predict true negatives. The average accuracy gained by B-MFO and comparative algorithms was statistically analyzed by the nonparametric Friedman test [109]. In addition, the convergence behavior of B-MFO and comparative algorithms were visualized.

5.3. Discussion of the Results

In this section, the best results achieved by B-MFO using three categories S-shaped, V-shaped, and U-shaped for each dataset are compared to comparative algorithms in terms of various metrics. Table 3 reports the average accuracy, the standard deviation of accuracy, and the average number of selected features. The average fitness and the standard deviation of fitness are indicated in Table 4, where the bold letters characterize the best results. In addition, Table 5 demonstrates achieved specificity and sensitivity by the k-NN classifier on large datasets, which proves that B-MFO has presented better results than the comparative algorithms. Our hypothesis is that the sensitivity and specificity of B-MFO are more reliable than other comparative algorithms when the size of datasets is increased. According to Figure 3 and the average accuracy obtained, the B-MFO outperforms the comparative algorithms, especially on large datasets. In addition, in most datasets, the minimum number of features selected by B-MFO shows that B-MFO could avoid the local optima trapping and obtain the optimum solution. Figure 4 presents the average number of selected features in large datasets: PenglungEW, Parkinson, Colon,

and Leukemia. These results indicate the significant effect of transfer functions on algorithms' behavior in the position updating of search agents and finding the optimum solution in the feature selection problem. Among the three categories of transfer functions used by B-MFO, the U-shaped transfer functions outperform the V-shaped and S-shaped in terms of maximizing the classification accuracy and minimizing the number of selected features, especially for large datasets.

Table 3. The accuracy and selected features' number gained by winner versions of B-MFO and comparative algorithms.

Datasets (Winner)	Metrics	BPSO	bGWO	BDA	BSSA	B-MFO
Pima (B-MFO-S1)	Avg accuracy	**0.7922**	0.7726	0.7849	0.7798	0.7902
	Std accuracy	0.0033	0.0063	0.0119	0.0079	0.0046
	Avg no. features	4.7333	7.6000	3.2667	4.7667	5.2667
Lymphography (B-MFO-V3)	Avg accuracy	**0.9163**	0.8694	0.9041	0.8882	0.9095
	Std accuracy	0.0099	0.0108	0.0182	0.8882	0.0089
	Avg no. features	8.9333	16.9667	5.5333	9.1000	5.3667
Breast-WDBC (B-MFO-U3)	Avg accuracy	0.9710	0.9626	0.9666	0.9655	**0.9719**
	Std accuracy	0.0021	0.0028	0.0078	0.0030	0.0020
	Avg no. features	12.8333	27.6000	2.4000	13.8000	3.2333
PenglungEW (B-MFO-U2)	Avg accuracy	0.9626	0.9541	0.9507	0.9567	**0.9692**
	Std accuracy	0.0040	0.0044	0.0126	0.0058	0.0063
	Avg no. features	161.0667	322.6667	83.5667	199.5000	81.5333
Parkinson (B-MFO-V2)	Avg accuracy	0.7952	0.7736	0.7643	0.7793	**0.8603**
	Std accuracy	0.0243	0.0036	0.0056	0.0126	0.0094
	Avg no. features	376.4333	741.2333	192.7333	332.7667	79.1000
Colon (B-MFO-U2)	Avg accuracy	0.9625	0.9526	0.9296	0.9535	**0.9694**
	Std accuracy	0.0056	0.0048	0.0207	0.0051	0.0059
	Avg no. features	999.9333	1948.8667	618.4333	1152.2000	350.7667
Leukemia (B-MFO-U2)	Avg accuracy	0.9988	0.9901	0.9703	0.9954	**0.9998**
	Std accuracy	0.0013	0.0021	0.0167	0.0023	0.0005
	Avg no. features	3542.0670	6746.9670	2283.7330	3435.2330	669.2333

Table 4. The comparison results between winner versions of B-MFO and comparative algorithms on fitness.

Datasets (Winner)	Metrics	BPSO	bGWO	BDA	BSSA	B-MFO
Pima	Avg fitness	**0.2117**	0.2347	0.2456	0.2240	0.2143
(B-MFO-S1)	Std fitness	0.0034	0.0068	0.0052	0.0076	0.0046
Lymphography	Avg fitness	**0.0878**	0.1387	0.1503	0.1157	0.0925
(B-MFO-V3)	Std fitness	0.0095	0.0110	0.0189	0.0106	0.0084
Breast-WDBC	Avg fitness	0.0330	0.0462	0.0571	0.0387	**0.0289**
(B-MFO-U3)	Std fitness	0.0019	0.0027	0.0111	0.0033	0.0021
PenglungEW	Avg fitness	0.0420	0.0554	0.8845	0.0490	**0.0330**
(B-MFO-U2)	Std fitness	0.0040	0.0043	0.1006	0.0059	0.0061
Parkinson	Avg fitness	0.2078	0.2340	2.1607	0.2229	**0.1393**
(B-MFO-V2)	Std fitness	0.0241	0.0035	0.2104	0.0135	0.0095
Colon	Avg fitness	0.0421	0.0567	6.2540	0.0518	**0.0321**
(B-MFO-U2)	Std fitness	0.0055	0.0048	0.5740	0.0051	0.0056
Leukemia	Avg fitness	0.0062	0.0192	22.8667	0.0094	**0.0011**
(B-MFO-U2)	Std fitness	0.0013	0.0022	2.6745	0.0023	0.0006

Table 5. The comparison results between winner versions of B-MFO and comparative algorithms on specificity and sensitivity.

Datasets (Winner)	Metrics	BPSO	bGWO	BDA	BSSA	B-MFO
PenglungEW	Avg specificity	0.9975	**1.0000**	0.9940	0.9980	0.9945
(B-MFO-U2)	Avg sensitivity	0.9722	0.9444	0.9333	0.9500	**0.9722**
Parkinson	Avg specificity	0.9004	0.8898	0.8876	0.8915	**0.9321**
(B-MFO-V2)	Avg sensitivity	0.3882	0.2913	0.3002	0.3686	**0.5510**
Colon	Avg specificity	0.9467	**0.9500**	0.9392	0.9467	0.9475
(B-MFO-U2)	Avg sensitivity	0.7712	0.6970	0.6606	0.7227	**0.8227**
Leukemia	Avg specificity	1.0000	1.0000	0.9972	0.9957	**1.0000**
(B-MFO-U2)	Avg sensitivity	0.9667	0.8413	0.7800	0.9267	**0.9947**

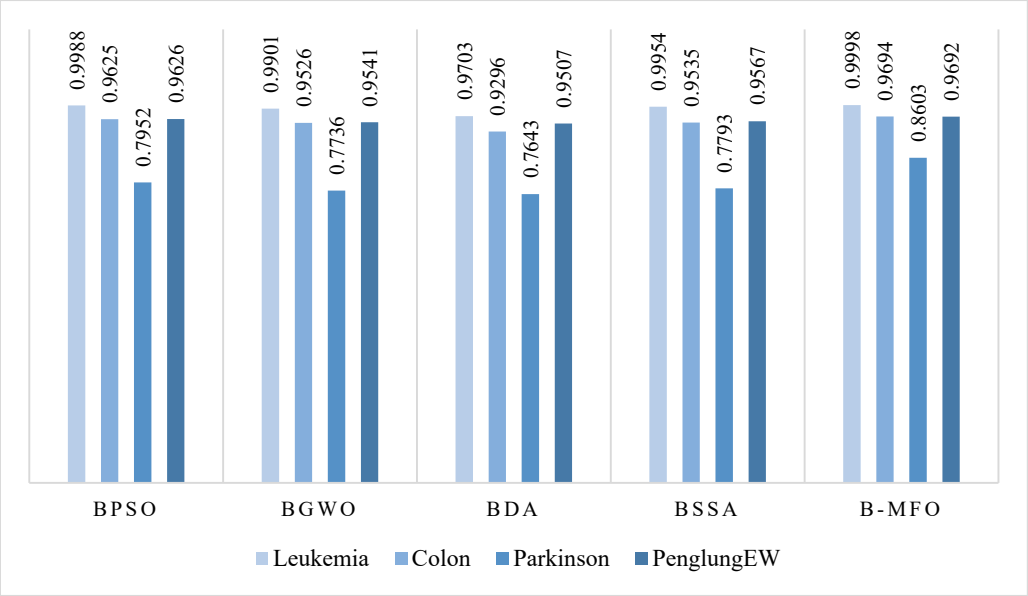

Figure 3. Average accuracy obtained by B-MFO and comparative algorithms on large datasets.

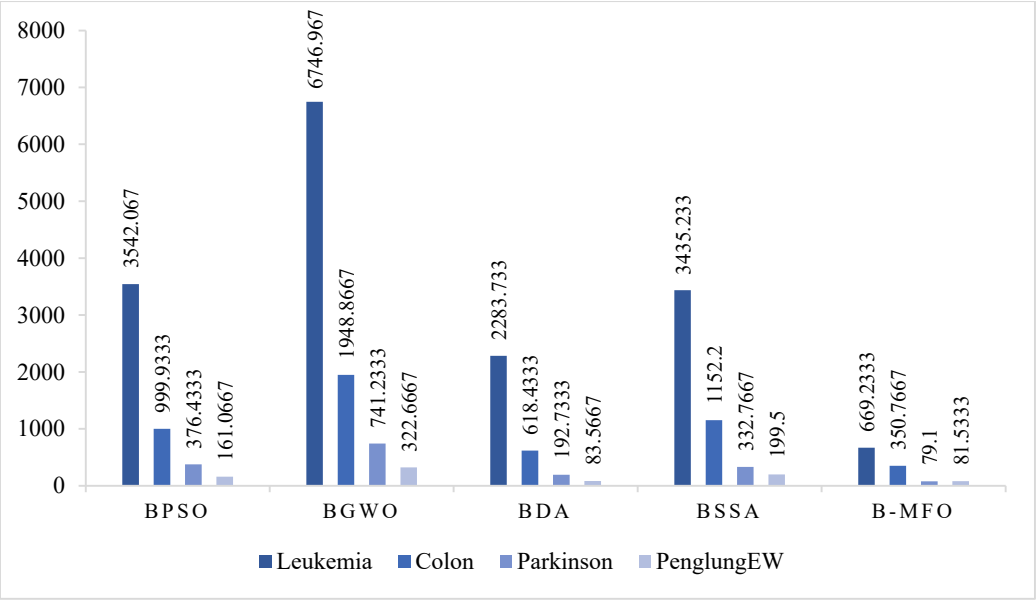

Figure 4. Average features' number selected by B-MFO and comparative algorithms on large datasets.

Figure 5 shows convergence curves of average fitness achieved by the winner version of B-MFO and comparative algorithms. The curves show that B-MFO can find better solutions and provide a balance between exploration and exploitation. To statistically analyze the results, the Friedman test was applied to rank B-MFO and comparative algorithms. Table 6 presents the results of the Friedman test on average accuracy achieved by B-MFO and comparative algorithms, which shows B-MFO has the first rank in comparison with other algorithms.

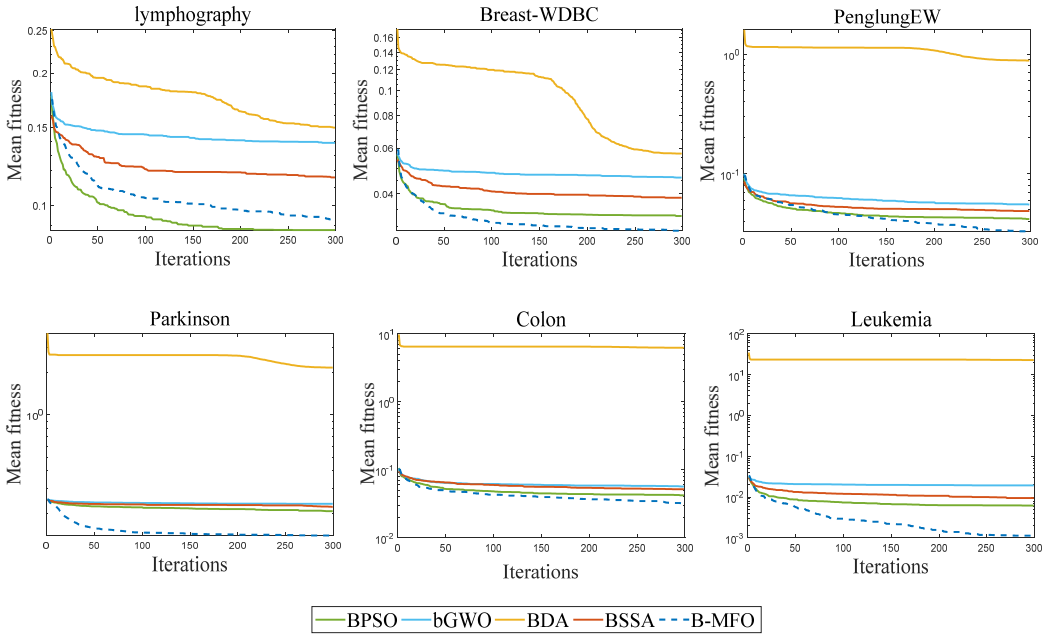

Figure 5. The convergence curves of winner versions of B-MFO and comparative algorithms.

Table 6. Friedman test for the accuracy obtained by winner versions of B-MFO and comparative algorithms.

Dataset	BPSO	bGWO	BDA	BSSA	B-MFO
Pima	4.27	1.47	3.17	2.33	3.77
Lymphography	3.17	7.17	2.33	8.73	3.77
Breast-WDBC	7.17	2.33	14.30	8.73	3.77
PenglungEW	2.33	17.40	18.90	8.73	3.77
Parkinson	10.10	22.40	23.90	13.70	7.10
Colon	10.10	26.50	28.60	13.90	7.10
Leukemia	10.10	27.00	29.10	13.90	7.10
Average rank	6.76	14.90	17.20	10.00	**5.20**
Overall rank	2	4	5	3	**1**

6. Conclusions and Future Work

Numerous large datasets that include redundant and irrelevant features have been created in the field of medical technology. To select effective features from different medical datasets, this study proposed three categories of binary moth-flame optimization (B-MFO). Consequently, the canonical MFO was converted from continuous to binary using variants of S-shaped, V-shaped, and U-shaped transfer functions. Each category contains four versions of transfer functions; accordingly, twelve versions of B-MFO were experimentally evaluated on seven medical datasets. Finally, the winner versions of B-MFO were compared with the best results achieved by four well-known binary metaheuristic optimization algorithms: BPSO, bGWO, BDA, and BSSA. The results show that the B-MFO

algorithm outperforms other comparative algorithms in terms of classification accuracy and minimizing the number of selected features, especially for large medical datasets. In addition, among variants of transfer functions used by B-MFO, the U-shaped functions outperform the V-shaped and S-shaped in terms of classification accuracy and minimized the number of selected features. For future works, B-MFO particularly using U-shaped transfer functions can be applied to select effective features in large-scale optimization problems, since it showed sufficient results for large datasets. In addition, B-MFO can be applied to other applications such as various engineering applications.

Author Contributions: Conceptualization, M.H.N.-S., M.B.-D.; Methodology, M.H.N.-S., M.B.-D.; Software, M.H.N.-S., M.B.-D.; Validation, M.H.N.-S., M.B.-D., H.Z., S.T.; Formal analysis, M.H.N.-S., M.B.-D., H.Z., S.T.; Investigation, M.H.N.-S., M.B.-D., H.Z., S.T.; Resources, M.H.N.-S., M.B.-D., H.Z., S.T., S.M.; Data curation, M.H.N.-S., M.B.-D.; Writing, M.H.N.-S., M.B.-D.; Original draft preparation, M.H.N.-S., M.B.-D.; Writing—review and editing, M.H.N.-S., M.B.-D., H.Z., S.T., S.M.; Visualization, M.H.N.-S., M.B.-D., H.Z., S.T., S.M.; Supervision, M.H.N.-S.; Project administration, M.H.N.-S., S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sorkhabi, L.B.; Gharehchopogh, F.S.; Shahamfar, J. A systematic approach for pre-processing electronic health records for mining: Case study of heart disease. *Int. J. Data Min. Bioinform.* **2020**, *24*, 97.
2. Esfandiari, N.; Babavalian, M.R.; Moghadam, A.-M.E.; Tabar, V.K. Knowledge discovery in medicine: Current issue and future trend. *Expert Syst. Appl.* **2014**, *41*, 4434–4463. https://doi.org/10.1016/j.eswa.2014.01.011.
3. Zhang, B.; Cao, P. Classification of high dimensional biomedical data based on feature selection using redundant removal. *PLoS ONE* **2019**, *14*, e0214406. https://doi.org/10.1371/journal.pone.0214406.
4. Arjenaki, H.G.; Nadimi-Shahraki, M.H.; Nourafza, N. A low cost model for diagnosing coronary artery disease based on effective features. *IJECCE* **2015**, *6*, 93–97.
5. Dezfuly, M.; Sajedi, H. Predict Survival of Patients with Lung Cancer Using an Ensemble Feature Selection Algotithm and Classification Methods in Data Mining. *J. Inf.* **2015**, *1*, 1–11.
6. Mohammadzadeh, H.; Gharehchopogh, F.S. Novel Hybrid Whale Optimization Algorithm with Flower Pollination Algorithm for Feature Selection: Case Study Email Spam Detection. *Comput. Intell.* **2021**, *37*, 176–209.
7. Abualigah, L.M.; Khader, A.T.; Al-Betar, M.A. Unsupervised feature selection technique based on genetic algorithm for improving the Text Clustering. **2016**, 1–6. https://doi.org/10.1109/csit.2016.7549453.
8. Abualigah, L.M.Q. *Feature Selection and Enhanced Krill Herd Algorithm for Text Document Clustering*; Springer: Berlin, Germany, 2019. https://doi.org/10.1007/978-3-030-10674-4.
9. Helmi, A.M.; Al-Qaness, M.A.; Dahou, A.; Damaševičius, R.; Krilavičius, T.; Elaziz, M.A. A novel hybrid gradient-based optimizer and grey wolf optimizer feature selection method for human ac-tivity recognition using smartphone sensors. *Entropy* **2021**, *23*, 1065.
10. Huan, L.; Lei, Y. Toward integrating feature selection algorithms for classification and clustering. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 491–502.
11. Xue, B.; Zhang, M.; Browne, W.N. Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. *Appl. Soft Comput.* **2014**, *18*, 261–276. https://doi.org/10.1016/j.asoc.2013.09.018.
12. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res*. **2003**, *3*, 1157–1182.
13. Pashaei, E.; Aydin, N. Binary black hole algorithm for feature selection and classification on biological data. *Appl. Soft Comput.* **2017**, *56*, 94–106. https://doi.org/10.1016/j.asoc.2017.03.002.
14. de Souza RC, T.; de Macedo, C.A.; dos Santos Coelho, L.; Pierezan, J.; Mariani, V.C. Binary coyote optimization algorithm for feature selection. *Pattern Recognit.* **2020**, *107*, 107470.
15. Faris, H.; Heidari, A.A.; Al-Zoubi, A.M.; Mafarja, M.; Aljarah, I.; Eshtay, M.; Mirjalili, S. Time-varying hierarchical chains of salps with random weight networks for feature selection. *Expert Syst. Appl.* **2019**, *140*, 112898. https://doi.org/10.1016/j.eswa.2019.112898.
16. Taghian, S.; Nadimi-Shahraki, M.H. Binary Sine Cosine Algorithms for Feature Selection from Medical Data. *Adv. Comput. Int. J.* **2019**, *10*, 1–10. https://doi.org/10.5121/acij.2019.10501.

17. Eberhart, R. and J. Kennedy. A new optimizer using particle swarm theory. In Proceedings of the MHS'Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October, 1995.
18. Mirjalili, S.; Mirjalili, S.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61.
19. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67.
20. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl. Based Syst.* **2015**, *89*, 228–249. https://doi.org/10.1016/j.knosys.2015.07.006.
21. Abualigah, L.; Yousri, D.; Elaziz, M.A.; Ewees, A.A.; Al-Qaness, M.A.; Gandomi, A.H. Aquila Optimizer: A novel meta-heuristic optimization algorithm. *Comput. Ind. Eng.* **2021**, *157*, 107250. https://doi.org/10.1016/j.cie.2021.107250.
22. Zamani, H.; Nadimi-Shahraki, M.H.; Gandomi, A. CCSA: Conscious Neighborhood-based Crow Search Algorithm for Solving Global Optimization Problems. *Appl. Soft Comput.* **2019**, *85*. https://doi.org/10.1016/j.asoc.2019.105583.
23. Askarzadeh, A. A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Comput. Struct.* **2016**, *169*, 1–12.
24. Shi, Y.; Pun, C.-M.; Hu, H.; Gao, H. An improved artificial bee colony and its application. *Knowl. Based Syst.* **2016**, *107*, 14–31. https://doi.org/10.1016/j.knosys.2016.05.052.
25. Shang, Y. Optimal Control Strategies for Virus Spreading in Inhomogeneous Epidemic Dynamics. *Can. Math. Bull.* **2013**, *56*, 621–629. https://doi.org/10.4153/cmb-2012-007-2.
26. Oliva, D.; Abd El Aziz, M.; Hassanien, A.E. Parameter estimation of photovoltaic cells using an improved chaotic whale optimization algorithm. *Appl. Energy* **2017**, *200*, 141–154.
27. Ewees, A.A.; Abd Elaziz, M.; Houssein, E.H. Improved grasshopper optimization algorithm using opposition-based learning. *Expert Syst. Appl.* **2018**, *112*, 156–172.
28. Nadimi-Shahraki, M.H.; Taghian, S.; Mirjalili, S. An improved grey wolf optimizer for solving engineering problems. *Expert Syst. Appl.* **2020**, *166*, 113917. https://doi.org/10.1016/j.eswa.2020.113917.
29. Liu, H.; Ding, G.; Wang, B. Bare-bones particle swarm optimization with disruption operator. *Appl. Math. Comput.* **2014**, *238*, 106–122. https://doi.org/10.1016/j.amc.2014.03.152.
30. Guo, L.; Wang, G.-G.; Gandomi, A.; Alavi, A.H.; Duan, H. A new improved krill herd algorithm for global numerical optimization. *Neurocomputing* **2014**, *138*, 392–402. https://doi.org/10.1016/j.neucom.2014.01.023.
31. Zaman, H.R.R.; Gharehchopogh, F.S. An improved particle swarm optimization with backtracking search optimization algorithm for solving continuous optimization problems. *Eng. Comput.* **2021**, 1–35. https://doi.org/10.1007/s00366-021-01431-6.
32. Banaie-Dezfouli, M.; Nadimi-Shahraki, M.H.; Beheshti, Z. R-GWO: Representative-based grey wolf optimizer for solving engineering problems. *Appl. Soft Comput.* **2021**, *106*,107328.
33. Fayyad, U.; Piatetsky-Shapiro, G.; Smyth, P. From data mining to knowledge discovery in databases. *AI Mag.* **1996**, *17*, 37–37.
34. Taghian, S.; Nadimi-Shahraki, M.H. A binary metaheuristic algorithm for wrapper feature selection. *Int. J. Comput. Sci. Eng.* **2019**, *8*, 168–172.
35. Mafarja, M.; Aljarah, I.; Faris, H.; Hammouri, A.I.; Al-Zoubi, A.M.; Mirjalili, S. Binary grasshopper optimisation algorithm approaches for feature selection problems. *Expert Syst. Appl.* **2018**, *117*, 267–286. https://doi.org/10.1016/j.eswa.2018.09.015.
36. Arora, S.; Anand, P. Binary butterfly optimization approaches for feature selection. *Expert Syst. Appl.* **2018**, *116*, 147–160. https://doi.org/10.1016/j.eswa.2018.08.051.
37. Mohammadzadeh, H.; Gharehchopogh, F.S. Feature Selection with Binary Symbiotic Organisms Search Algorithm for Email Spam Detection. *Int. J. Inf. Technol. Decis. Mak.* **2021**, *20*, 469–515.
38. Safaldin, M.; Otair, M.; Abualigah, L. Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *12*, 1559–1576. https://doi.org/10.1007/s12652-020-02228-z.
39. Khurma, R.; Alsawalqah, H.; Aljarah, I.; Elaziz, M.; Damaševičius, R. An Enhanced Evolutionary Software Defect Prediction Method Using Island Moth Flame Optimization. *Mathematics* **2021**, *9*, 1722. https://doi.org/10.3390/math9151722.
40. Elsakaan, A.A.; El-Sehiemy, R.A.; Kaddah, S.S.; Elsaid, M.I. An enhanced moth-flame optimizer for solving non-smooth economic dispatch problems with emissions. *Energy* **2018**, *157*, 1063–1078. https://doi.org/10.1016/j.energy.2018.06.088.
41. Pelusi, D.; Mascella, R.; Tallini, L.; Nayak, J.; Naik, B.; Deng, Y. An Improved Moth-Flame Optimization algorithm with hybrid search phase. *Knowl. Based Syst.* **2019**, *191*, 105277. https://doi.org/10.1016/j.knosys.2019.105277.
42. Xu, Y.; Chen, H.; Heidari, A.A.; Luo, J.; Zhang, Q.; Zhao, X.; Li, C. An efficient chaotic mutative moth-flame-inspired optimizer for global optimization tasks. *Expert Syst. Appl.* **2019**, *129*, 135–155. https://doi.org/10.1016/j.eswa.2019.03.043.
43. Neshat, M.; Sergiienko, N.; Mirjalili, S.; Nezhad, M.M.; Piras, G.; Garcia, D.A. Multi-Mode Wave Energy Converter Design Optimisation Using an Improved Moth Flame Optimisation Algorithm. *Energies* **2021**, *14*, 3737. https://doi.org/10.3390/en14133737.
44. Kennedy, J.; Eberhart, R. A discrete binary version of the particle swarm algorithm. In Proceedigs of the IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Orlando, FL, USA, 12–15 October 1997; pp. 4104–4108. https://doi.org/10.1109/icsmc.1997.637339.
45. Emary, E.; Zawbaa, H.M.; Hassanien, A.E. Binary grey wolf optimization approaches for feature selection. *Neurocomputing* **2016**, *172*, 371–381. https://doi.org/10.1016/j.neucom.2015.06.083.
46. Mafarja, M.; Aljarah, I.; Heidari, A.A.; Faris, H.; Fournier-Viger, P.; Li, X.; Mirjalili, S. Binary dragonfly optimization for feature selection using time-varying transfer functions. *Knowledge-Based Syst.* **2018**, *161*, 185–204. https://doi.org/10.1016/j.knosys.2018.08.003.

47. Faris, H.; Mafarja, M.M.; Heidari, A.A.; Aljarah, I.; Al-Zoubi, A.; Mirjalili, S.; Fujita, H. An efficient binary Salp Swarm Algorithm with crossover scheme for feature selection problems. *Knowledge-Based Syst.* **2018**, *154*, 43–67. https://doi.org/10.1016/j.knosys.2018.05.009.
48. El Aziz, M.A.; Hassanien, A.E. Modified cuckoo search algorithm with rough sets for feature selection. *Neural Comput. Appl.* **2016**, *29*, 925–934. https://doi.org/10.1007/s00521-016-2473-7.
49. Ibrahim, R.A.; Elaziz, M.A.; Oliva, D.; Cuevas, E.; Lu, S. An opposition-based social spider optimization for feature selection. *Soft Comput.* **2019**, *23*, 13547–13567. https://doi.org/10.1007/s00500-019-03891-x.
50. Dezfouli, M.B.; Nadimi-Shahraki, M.H.; Zamani, H. A novel tour planning model using big data. In Proceedings of the 2018 International Conference on Artificial Intelligence and Data Processing, New York, NY, USA, 28–30 September 2018; pp. 1–6. https://doi.org/10.1109/idap.2018.8620933.
51. Shang, Y.; Bouffanais, R. Consensus reaching in swarms ruled by a hybrid metric-topological distance. *Eur. Phys. J. B* **2014**, *87*, 1–7. https://doi.org/10.1140/epjb/e2014-50094-4.
52. Benyamin, A.; Farhad, S.G.; Saeid, B. Discrete farmland fertility optimization algorithm with metropolis acceptance criterion for traveling salesman problems. *Int. J. Intell. Syst.* **2020**, *36*, 1270–1303. https://doi.org/10.1002/int.22342.
53. Talbi, E.-G. *Metaheuristics: From Design to Implementation*; Wiley Publishing: Hoboken, NJ, USA, 2009.
54. Hafez, A.I.; Emary, E.; Zawbaa, H.M.; Hassanien, A.E. Sine cosine optimization algorithm for feature selection. In Proceedings of the 2016 International Symposium on INnovations in Intelligent SysTems and Applications, Sinaia, Romania, 2–5 August 2016.
55. Mafarja, M.; Mirjalili, S. Whale optimization approaches for wrapper feature selection. *Appl. Soft Comput.* **2018**, *62*, 441–453. https://doi.org/10.1016/j.asoc.2017.11.006.
56. Emary, E.; Zawbaa, H.M.; Hassanien, A.E. Binary ant lion approaches for feature selection. *Neurocomputing* **2016**, *213*, 54–65. https://doi.org/10.1016/j.neucom.2016.03.101.
57. Ghimatgar, H.; Kazemi, K.; Helfroush, M.S.; Aarabi, A. An improved feature selection algorithm based on graph clustering and ant colony optimization. *Knowledge-Based Syst.* **2018**, *159*, 270–285. https://doi.org/10.1016/j.knosys.2018.06.025.
58. Mirjalili, S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **2015**, *27*, 1053–1073. https://doi.org/10.1007/s00521-015-1920-1.
59. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191.
60. Hashim, F.A.; Houssein, E.H.; Mabrouk, M.S.; Al-Atabany, W.; Mirjalili, S. Henry gas solubility optimization: A novel physics-based algorithm. *Futur. Gener. Comput. Syst.* **2019**, *101*, 646–667. https://doi.org/10.1016/j.future.2019.07.015.
61. Shayanfar, H.; Gharehchopogh, F.S. Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems. *Appl. Soft Comput.* **2018**, *71*, 728–746. https://doi.org/10.1016/j.asoc.2018.07.033.
62. Nadimi-Shahraki, M.H.; Taghian, S.; Mirjalili, S.; Faris, H. MTDE: An effective multi-trial vector-based differential evolution algorithm and its applications for engineering design problems. *Appl. Soft Comput.* **2020**, *97*, 106761.
63. Zamani, H.; Nadimi-Shahraki, M.H.; Gandomi, A.H. QANA: Quantum-based avian navigation optimizer algorithm. *Eng. Appl. Artif. Intell.* **2021**, *104*, 104314. https://doi.org/10.1016/j.engappai.2021.104314.
64. Zamani, H.; Nadimi-Shahraki, M.H.; Taghian, S.; Banaie-Dezfouli, M. Enhancement of bernstain-search differential evolution algorithm to solve constrained engineering problems. *Int. J. Comput. Sci. Eng.* **2020**, 386-396.
65. Ghasemi, M.R.; Varaee, H. Enhanced IGMM optimization algorithm based on vibration for numerical and engineering problems. *Eng. Comput.* **2017**, *34*, 91–116. https://doi.org/10.1007/s00366-017-0523-0.
66. Varaee, H.; Hamzehkolaei, N.S.; Safari, M. A Hybrid Generalized Reduced Gradient-Based Particle Swarm Optimizer for Constrained Engineering Optimization Problems. *J. Soft Comput. Civil Eng.* **2021**, *5*, 86–119. https://doi.org/10.22115/SCCE.2021.282360.1304.
67. Sayarshad, H.R. Using bees algorithm for material handling equipment planning in manufacturing systems. *Int. J. Adv. Manuf. Technol.* **2009**, *48*, 1009–1018. https://doi.org/10.1007/s00170-009-2363-6.
68. Fard, E.S.; Monfaredi, K.; Nadimi-Shahraki, M.H. An Area-Optimized Chip of Ant Colony Algorithm Design in Hardware Platform Using the Address-Based Method. *Int. J. Electr. Comput. Eng.* **2014**, *4*, 989. https://doi.org/10.11591/ijece.v4i6.6923.
69. Tadi, A.A.; Aghajanloo, Z. Load Balancing in Cloud Computing using Cuckoo Optimization Algorithm. *J. Innov. Res. Eng. Sci.* **2018**, *4*.
70. Zahrani, H.K.; Nadimi-Shahraki, M.H.; Sayarshad, H.R. An intelligent social-based method for rail-car fleet sizing problem. *J. Rail Transp. Plan. Manag.* **2020**, *17*, 100231. https://doi.org/10.1016/j.jrtpm.2020.100231.
71. Taghian, S.; Nadimi-Shahraki, M.-H.; Zamani, H. Comparative Analysis of Transfer Function-based Binary Metaheuristic Algorithms for Feature Selection. In Proceedings of the International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turkey, 28–30 Septembrie 2018; pp. 1–6. https://doi.org/10.1109/idap.2018.8620828.
72. Aslan, M.; Gunduz, M.; Kiran, M.S. JayaX: Jaya algorithm with xor operator for binary optimization. *Appl. Soft Comput.* **2019**, *82*, 105576. https://doi.org/10.1016/j.asoc.2019.105576.
73. Jia, D.; Duan, X.; Khan, M.K. Binary Artificial Bee Colony optimization using bitwise operation. *Comput. Ind. Eng.* **2014**, *76*, 360–365. https://doi.org/10.1016/j.cie.2014.08.016.
74. Mirjalili, S.; Lewis, A. S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization. *Swarm Evol. Comput.* **2013**, *9*, 1–14. https://doi.org/10.1016/j.swevo.2012.09.002.

75. Nezamabadi-pour, H.; Rostami-Shahrbabaki, M.; Maghfoori-Farsangi, M. Binary particle swarm optimization: Challenges and new solutions. *CSI J Comput Sci Eng* **2008**, *6*, 21–32.
76. Mirjalili, S.; Zhang, H.; Mirjalili, S.; Chalup, S.; Noman, N. *A Novel U-Shaped Transfer Function for Binary Particle Swarm Optimisation. in Soft Computing for Problem Solving 2019*; Springer: Singapore, 2019.
77. Pedrasa, M.A.A.; Spooner, T.D.; MacGill, I.F. Scheduling of Demand Side Resources Using Binary Particle Swarm Optimization. *IEEE Trans. Power Syst.* **2009**, *24*, 1173–1181. https://doi.org/10.1109/tpwrs.2009.2021219.
78. Liao, C.-J.; Tseng, C.-T.; Luarn, P. A discrete version of particle swarm optimization for flowshop scheduling problems. *Comput. Oper. Res.* **2007**, *34*, 3099–3111. https://doi.org/10.1016/j.cor.2005.11.017.
79. Lin, J.C.-W.; Yang, L.; Fournier-Viger, P.; Hong, T.-P.; Voznak, M. A binary PSO approach to mine high-utility itemsets. *Soft Comput.* **2016**, *21*, 5103–5121. https://doi.org/10.1007/s00500-016-2106-1.
80. Yuan, X.; Nie, H.; Su, A.; Wang, L.; Yuan, Y. An improved binary particle swarm optimization for unit commitment problem. *Expert Syst. Appl.* **2009**, *36*, 8049–8055. https://doi.org/10.1016/j.eswa.2008.10.047.
81. Bharti, K.K.; Singh, P.K. Opposition chaotic fitness mutation based adaptive inertia weight BPSO for feature selection in text clustering. *Appl. Soft Comput.* **2016**, *43*, 20–34. https://doi.org/10.1016/j.asoc.2016.01.019.
82. Abualigah, L.; Khader, A.T.; Hanandeh, E.S. A new feature selection method to improve the document clustering using particle swarm optimization algorithm. *J. Comput. Sci.* **2018**, *25*, 456–466. https://doi.org/10.1016/j.jocs.2017.07.018.
83. Lu, Y.; Liang, M.; Ye, Z.; Cao, L. Improved particle swarm optimization algorithm and its application in text feature selection. *Appl. Soft Comput.* **2015**, *35*, 629–636. https://doi.org/10.1016/j.asoc.2015.07.005.
84. Sheikhpour, R.; Sarram, M.A.; Sheikhpour, R. Particle swarm optimization for bandwidth determination and feature selection of kernel density estimation based classifiers in diagnosis of breast cancer. *Appl. Soft Comput.* **2016**, *40*, 113–131.
85. Gunasundari, S.; Janakiraman, S.; Meenambal, S. Velocity Bounded Boolean Particle Swarm Optimization for improved feature selection in liver and kidney disease diagnosis. *Expert Syst. Appl.* **2016**, *56*, 28–47. https://doi.org/10.1016/j.eswa.2016.02.042.
86. Zamani, H.; Nadimi-Shahraki, M.H. Swarm Intelligence Approach for Breast Cancer Diagnosis. *Int. J. Comput. Appl.* **2016**, *151*, 40–44. https://doi.org/10.5120/ijca2016911667.
87. Al-Tashi, Q.; Kadir, S.J.A.; Rais, H.M.; Mirjalili, S.; Alhussian, H. Binary Optimization Using Hybrid Grey Wolf Optimization for Feature Selection. *IEEE Access* **2019**, *7*, 39496–39508. https://doi.org/10.1109/access.2019.2906757.
88. Panwar, L.K.; Reddy, S.; Verma, A.; Panigrahi, B.K.; Kumar, R. Binary Grey Wolf Optimizer for large scale unit commitment problem. *Swarm Evol. Comput.* **2018**, *38*, 251–266. https://doi.org/10.1016/j.swevo.2017.08.002.
89. Chantar, H.; Mafarja, M.; Alsawalqah, H.; Heidari, A.A.; Aljarah, I.; Faris, H. Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification. *Neural Comput. Appl.* **2019**, *32*, 12201–12220. https://doi.org/10.1007/s00521-019-04368-6.
90. Hu, P.; Pan, J.-S.; Chu, S.-C. Improved Binary Grey Wolf Optimizer and Its application for feature selection. *Knowledge-Based Syst.* **2020**, *195*, 105746. https://doi.org/10.1016/j.knosys.2020.105746.
91. Zamani, H.; Nadimi-Shahraki, M.H. Feature selection based on whale optimization algorithm for diseases diagnosis. *Int. J. Comput. Sci. Inf. Secur.* **2016**, *14*, 1243.
92. Hussien, A.G.; Houssein, E.H.; Hassanien, A.E. A binary whale optimization algorithm with hyperbolic tangent fitness function for feature selection. In Proceedings of the Eighth International Conference on Intelligent Computing and Information Systems, Cairo, Egypt, 5–7 December 2017; pp. 166–172. https://doi.org/10.1109/intelcis.2017.8260031.
93. Hussien, A.G.; Hassanien, A.E.; Houssein, E.H.; Bhattacharyya, S.; Amin, M. S-shaped binary whale optimization algorithm for feature selection. In *Recent Trends in Signal and Image Processing*; Springer: Singapore, 2019, pp. 79–87. https://doi.org/10.1007/978-981-10-8863-6_9.
94. Reddy, K.S.; Panwar, L.; Panigrahi, B.K.; Kumar, R. Binary whale optimization algorithm: A new metaheuristic approach for profit-based unit commitment problems in competitive electricity markets. *Eng. Optim.* **2019**, *51*, 369–389.
95. Mafarja, M.M.; Eleyan, D.; Jaber, I.; Hammouri, A.; Mirjalili, S. Binary dragonfly algorithm for feature selection. In Proceedings of the 2017 International Conference on New Trends in Computing Sciences (ICTCS), Amman, Jordan, 11–13 October 2017.
96. Sawhney, R.; Jain, R. Modified Binary Dragonfly Algorithm for Feature Selection in Human Papillomavirus-Mediated Disease Treatment. In Proceedings of the 2018 International Conference on Communication, Computing and Internet of Things (IC3IoT), Tamil Nadu, India, 15–17 February 2018; pp. 91–95. https://doi.org/10.1109/ic3iot.2018.8668174.
97. Too, J.; Mirjalili, S. A Hyper Learning Binary Dragonfly Algorithm for Feature Selection: A COVID-19 Case Study. *Knowledge-Based Syst.* **2020**, *212*, 106553. https://doi.org/10.1016/j.knosys.2020.106553.
98. Ibrahim, R.A.; Ewees, A.; Oliva, D.; Elaziz, M.A.; Lu, S. Improved salp swarm algorithm based on particle swarm optimization for feature selection. *J. Ambient. Intell. Humaniz. Comput.* **2018**, *10*, 3155–3169. https://doi.org/10.1007/s12652-018-1031-9.
99. Meraihi, Y.; Ramdane-Cherif, A.; Mahseur, M.; Achelia, D. A Chaotic binary salp swarm algorithm for solving the graph coloring problem. In *International Symposium on Modelling and Implementation of Complex Systems*; Springer: Cham, Switzerland, 2018, pp. 106–118. https://doi.org/10.1007/978-3-030-05481-6_8.
100. Geem, Z.W.; Kim, J.H.; Loganathan, G. A New Heuristic Optimization Algorithm: Harmony Search. *SIMULATION* **2001**, *76*, 60–68. https://doi.org/10.1177/003754970107600201.
101. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. BGSA: Binary gravitational search algorithm. *Nat. Comput.* **2009**, *9*, 727–745. https://doi.org/10.1007/s11047-009-9175-3.

102. Reddy, S.; Panwar, L.K.; Panigrahi, B.K.; Kumar, R. Solution to Unit Commitment in Power System Operation Planning Using Modified Moth Flame Optimization Algorithm (MMFOA): A Flame Selection Based Computational Technique. *J. Comput. Sci.* **2017**, *25*, 298–317.
103. Ghosh, K.K.; Singh, P.K.; Hong, J.; Geem, Z.W.; Sarkar, R. Binary Social Mimic Optimization Algorithm With X-Shaped Transfer Function for Feature Selection. *IEEE Access* **2020**, *8*, 97890–97906. https://doi.org/10.1109/access.2020.2996611.
104. Wang, L.; Fu, X.; Menhas, M.I.; Fei, M. *A Modified Binary Differential Evolution Algorithm*; Springer: Berlin, Germany, 2010, pp. 49–57. https://doi.org/10.1007/978-3-642-15597-0_6.
105. Michalak, K.; Selecting Best Investment Opportunities from Stock Portfolios Optimized by a Multiobjective Evolutionary Algorithm. In Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation. 2015, Association for Computing Machinery, Madrid, Spain, 11–15 July 2015; pp. 1239–1246.
106. Daolio, F.; et al., Global vs local search on multi-objective NK-landscapes: Contrasting the impact of problem features. 2015.
107. Statnikov, A.; Aliferis, C.F.; Tsamardinos, I.; Hardin, D.; Levy, S. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics* **2004**, *21*, 631–643. https://doi.org/10.1093/bioinformatics/bti033.
108. Dua, C.G.D. Machine Learning Repository. 2017. http://archive.ics.uci.edu/ml/index.php.
109. Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. https://doi.org/10.1016/j.swevo.2011.02.002.