

Article

Binary Aquila Optimizer for Selecting Effective Features from Medical Data: A COVID-19 Case Study

Mohammad H. Nadimi-Shahraki ^{1,2,3,*}, Shokooh Taghian ^{1,2}, Seyedali Mirjalili ^{3,4,*} and Laith Abualigah ⁵

¹ Faculty of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad 8514143131, Iran; sh.taghian@sco.iaun.ac.ir

² Big Data Research Center, Najafabad Branch, Islamic Azad University, Najafabad 8514143131, Iran

³ Centre for Artificial Intelligence Research and Optimisation, Torrens University Australia, Brisbane 4006, Australia

⁴ Yonsei Frontier Lab, Yonsei University, Seoul 03722, Korea

⁵ Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan; aligah.2020@gmail.com

* Correspondence: nadimi@iaun.ac.ir (M.H.N.-S.); ali.mirjalili@gmail.com (S.M.)

Abstract: Medical technological advancements have led to the creation of various large datasets with numerous attributes. The presence of redundant and irrelevant features in datasets negatively influences algorithms and leads to decreases in the performance of the algorithms. Using effective features in data mining and analyzing tasks such as classification can increase the accuracy of the results and relevant decisions made by decision-makers using them. This increase can become more acute when dealing with challenging, large-scale problems in medical applications. Nature-inspired metaheuristics show superior performance in finding optimal feature subsets in the literature. As a seminal attempt, a wrapper feature selection approach is presented on the basis of the newly proposed Aquila optimizer (AO) in this work. In this regard, the wrapper approach uses AO as a search algorithm in order to discover the most effective feature subset. S-shaped binary Aquila optimizer (SBAO) and V-shaped binary Aquila optimizer (VBAO) are two binary algorithms suggested for feature selection in medical datasets. Binary position vectors are generated utilizing S- and V-shaped transfer functions while the search space stays continuous. The suggested algorithms are compared to six recent binary optimization algorithms on seven benchmark medical datasets. In comparison to the comparative algorithms, the gained results demonstrate that using both proposed BAO variants can improve the classification accuracy on these medical datasets. The proposed algorithm is also tested on the real-dataset COVID-19. The findings testified that SBAO outperforms comparative algorithms regarding the least number of selected features with the highest accuracy.

Keywords: transfer function; medical data; nature-inspired algorithm; binary metaheuristic algorithm; feature selection

MSC: 68T20



Citation: Nadimi-Shahraki, M.H.; Taghian, S.; Mirjalili, S.; Abualigah, L. Binary Aquila Optimizer for Selecting Effective Features from Medical Data: A COVID-19 Case Study. *Mathematics* **2022**, *10*, 1929. <https://doi.org/10.3390/math10111929>

Academic Editor: Pasi Luukka

Received: 11 May 2022

Accepted: 30 May 2022

Published: 4 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

It is difficult and frequently imprecise to analyze big amounts of data in real-world problems such as engineering, health, biology, or tourism, due to the presence of massive volumes of data. This massive data, often called big data, contains a combination of features with relevant/irrelevant, redundant, or noisy characteristics [1]. Therefore, selecting effective features is an essential pre-processing technique and prominent data analysis task. Feature selection is regarded as a searching or optimization task since its objective is to find effective features that utilize to enhance classification performance [2]. Furthermore, reducing the complexity and ditching irrelevant and redundant features while enhancing the algorithms' performance are the main aims of feature selection approaches. Feature

selection is also often called dimensionality reduction, which has a variety of applications in various areas, including but not limited to image analysis, biomedical problems, text mining, and industrial applications.

The literature classifies feature selection strategies into two categories, filter and wrapper approaches, depending on their interaction with the classifier [3]. The former method finds the optimum subset of features based on measures such as distance, dependency, or consistency of the features. The latter method employs a learning algorithm, such as a classifier, to continuously assess the feature subset during the search process to locate optimal solutions from an exponential collection of features [3]. Although filter techniques are faster than wrapper methods since they do not consume additional processing time to invoke the learning algorithm [4], their major drawback is that features do not affect the classifier's performance [5]. Meanwhile, the wrapper approaches are more accurate but computationally more costly [6]. Thus, it is required to employ an optimization algorithm in order to achieve an optimal set of features.

As mentioned above, the process of selecting features can be considered an optimization issue since it aims to identify a subset of features with near-optimal fitness. Considering the proof that the number of the proper subsets in a dataset with Q features is $2^Q - 1$, feature selection can be regarded as an NP-hard problem. Thus, finding the best subset of features is impractical by using exact search strategies [7], and approximate approaches such as metaheuristic algorithms can be used to select effective features within a reasonable time [8]. Optimization algorithms have numerous advantages over conventional search algorithms, including the capability of identifying the optimal subset of features within a reasonable time. Furthermore, the conventional search algorithms require generating all possible feature subsets to find the optimum solution, which is unsuitable and time-consuming for large datasets [9].

Metaheuristics are designed to solve challenging optimization problems and provide acceptable solutions in a reasonable time [8]. Such general-purpose algorithms have the potential to locate the search space promising regions and the ability to estimate an appropriate solution for a particular optimization issue. In addition, metaheuristic algorithms utilize stochastic strategies to foster population diversity in the early stages of iterations by searching in the target domain. Meanwhile, the algorithm searches for potential solutions locally in the exploitation phase to improve upon the quality of solutions found in the exploration phase. After a limited number of iterations, a reasonable convergence occurs, meaning that the algorithm does not provide more improvement to the solution. Black-box nature, simplicity, ease of use, and high global search capability are the main reasons for the popularity of such algorithms in diverse areas [10]. Some of the trending application areas are: classification [11,12], power and energy management [13–16], structural engineering [17–19], community detection [20], clustering [21–24], image segmentation [25–28], global optimization [29–35], industrial engineering [36,37], life-cycle cost analysis in structural optimization [38], engineering design problem [39–41], task scheduling [42–44] and virtual machine placement [45,46] in cloud computing, navigation planning [47–49], and wind speed prediction [50,51].

Metaheuristic algorithms are mostly nature-inspired and are designed to emulate the biological, physical, and social behaviors of species found in nature. We can group them into three categories: evolution-based, swarm intelligence-based, and physics-based algorithms [52]. Evolution-based algorithms tend to mimic creatures' evolutionary behavior concepts in nature. Some of the most well-regarded evolutionary algorithms are genetic programming (GP) [53], evolution strategy (ES) [54], differential evolution (DE) [55], evolutionary programming (EP) [56], genetic algorithm (GA) [57], and biogeography-based optimization (BBO) [58]. The social and collective behavior of swarms in nature, such as those in colonies of bees, ants, animal herds, and birds' flocks, are the source of inspiration for swarm intelligence algorithms. Among this category, the most popular algorithms are particle swarm optimization (PSO) [59], ant colony optimization (ACO) [60], artificial bee colony (ABC) [61], and bat algorithm (BA) [62].

Algorithms based on the fundamental physical rules existing in nature comprise the last category. Simulated annealing (SA) [63], thermal exchange optimization (TEO) [64], big bang-big crunch (BB-BC) [65], and atom search optimization (ASO) [66] are some notable examples of this category. Although the proposed algorithms are designed and evaluated on different kinds of problems, the No-free-lunch (NFL) theorem [67] states that there is no general-purpose algorithm that is suitable for tackling all problems with various characteristics. Thus, the algorithms such as multi-trial vector-based differential evolution (MTDE) [68], multi-strategy enhanced HHO algorithm (RLHHO) [69], and gaze cues learning-based grey wolf optimizer (GGWO) [70] were proposed to overcome the shortcoming of the existing algorithms.

This paper attempts to find effective features by utilizing the Aquila optimizer (AO) [71] algorithm, which is one of the most recent optimization algorithms to be published. The AO mimics Aquila's hunting strategies in nature. Fast-moving prey hunting techniques reveal the algorithm's global exploration capability, whereas slow-moving prey hunting strategies show the algorithm's local exploitation capability. The canonical AO operates on the continuous solution search space and cannot be performed on problems with discrete, binary, or mixed-integer variables. Thus, we propose a wrapper-based binary metaheuristic algorithm named BAO to select effective features from seven medical datasets. The BAO algorithm calculates the position of solutions in the binary space using S and V-shaped transfer functions while the AO's search space remains continuous.

The following is how the rest of this article is structured: Section 2 provides a literature review of binary metaheuristic algorithms applicable to feature selection. The continuous Aquila optimizer algorithm is discussed in Section 3. Section 4 describes the suggested binary versions of the Aquila optimizer algorithm. Section 5 discusses the suggested algorithms for the feature selection problem, whereas Section 6 shows the experimental findings and statistical analysis on medical datasets. In Section 7, the applicability of the proposed algorithm on the COVID-19 dataset is assessed and compared with comparative algorithms. Finally, Section 8 concludes the work, and future research is suggested.

2. Related Work

In applications that rely on machine learning classifiers such as data mining, classification is considered to be an essential process. However, the type of features used, some of which are usually irrelevant and noisy, has a significant effect on the performance of these classifiers. Feature selection can minimize the dimensionality of data, reduce classifier learning time, and enhance classification result by choosing the most effective features and eliminating the redundant and irrelevant features [72]. Many practical applications have used feature selection including intrusion detection [73,74], software fault prediction [75,76], speech emotion recognition [77,78], bankruptcy prediction [79,80], credit scoring [81], stock trend prediction [82], emotion analysis [83], spam detection [84,85], digital soil mapping [86], diseases prediction and detection [87–90], breath analysis [91], biodiesel property selection [92], gene selection [93,94], and wind forecasting [95,96].

As stated, feature selection has an essential role in dimensionality reduction since it removes irrelevant and redundant features from the original dataset to find an optimal subset of features. In the following paragraphs, our focus is on reviewing the feature selection methods that use metaheuristics as search algorithms and rely on using transfer functions. The challenge of selecting effective features from an entire set of features is a discrete optimization problem that can be handled by metaheuristics. Metaheuristic algorithms, in which the search is directed by information gained during the optimization process, are practical approaches for solving feature selection problems. Most of the well-regarded nature-inspired algorithms were designed to address continuous optimization issues, whereas some problems are binary in nature. Furthermore, other strategies for adapting continuous metaheuristic algorithms into discrete domains have also been proposed, such as normalizing, rounding, and utilizing binary operators [97]. The transfer

function can also be used as another method of converting continuous components into binary values.

Due to the binary nature of feature selection, transfer functions are efficient yet easy ways to limit the result such that 0 means the feature is redundant and not chosen, and 1 represents the feature is useful and chosen. Transfer functions based on their shape have been divided into three groups S-shaped, V-shaped, and U-shaped transfer functions [98–101]. In prominent algorithms such as BPSO [98], binary differential evolution (binDE) [102], and altruistic whale optimization algorithm (AltWOA) [103], the S-shaped transfer function was employed to estimate the probabilities of shifting positions. In [104], a V-shaped transfer function is applied to the velocity parameter of the continuous gravitational search algorithm to handle the feature selection.

In [105], two binary variants were suggested for feature selection with the GWO algorithm as a state-of-the-art algorithm. In the first version, the position changing estimation for each wolf is accomplished by utilizing an S-shaped transfer function to their respective positions. The second variant converted the continuous GWO to a binary version via stochastic crossover. In [106], the binary butterfly optimization algorithm (bBOA) was proposed to address feature selection issues by using both S-shaped and V-shaped transfer functions. Each feature subset is represented as a butterfly, and global and local strategies are used during the search process. However, there are certain flaws with bBOA, such that it cannot balance the exploration and exploitation, and in the local search strategy, butterflies merely change their position at random, which is regarded as insufficient [107]. In another similar work [108], a wrapper-based binary SCA (WBSCA) was proposed that transformed continuous SCA to binary search space by utilizing the V-shaped transfer function. An improved binary PSO algorithm (ISBPSO) [109] is proposed for selecting features and used in classification in recent work. The performance of the canonical PSO algorithm is improved by adapting three strategies to obtain a better initial population, global search, and convergence property. The proposed ISBPSO is used as a wrapper to select features and validated on 12 UCI datasets. The comparison of obtained results by other feature selection algorithms shows that it can obtain higher or similar accuracy with fewer selected features. Another study [110] proposed a wrapper method for selecting features, which used MFO as a search strategy. This method used twelve different transfer functions of S, V, and U-shaped to transform MFO from the continuous version to a binary one. The proposed variants and wrapper-based comparative algorithms were tested and compared on medical datasets to confirm competitive performance.

In [111], a binary horse herd optimization algorithm (BHOA) is proposed to tackle feature selection problems. In BHOA, four configurations of the S, V, and U-shaped functions are utilized for mapping the HOA to its binary equivalent. Another proposed binary algorithm for feature selection is binary biogeography-based optimization based SVM-RFE (BBO-SVM-RFE) [112]. The support vector machine recursive feature elimination is incorporated into the BBO to enhance the quality of the acquired solutions in the mutation operator, thereby improving the balance between the exploitative and exploratory aspects of the original BBO. On the basis of the accuracy and the number of considered features, the BBO-SVM-REF algorithm surpasses the BBO method and other wrapper and filter methods. The usage of optimization algorithms is advantageous in selecting features because they can produce a solution near to the optimum or ideal in a reasonable amount of time. Contrary to this, conventional exhaustive search methods search through every possible combination of features from the entire feature set, which is time-consuming and regarded as an NP-hard task. Despite the fact that various metaheuristic algorithms for dealing with feature selection have been developed over time, the increasing dimensionality of data presents significant challenges; therefore, it is worthwhile to continue searching for effective strategies to improve the performance of metaheuristic algorithms to deal with high-dimensional feature selection issues.

3. Aquila Optimizer (AO)

In accordance with the natural Aquila hunting process, the Aquila optimizer (AO) [71] is a newly proposed algorithm. The hunting process has four steps: expanded exploration by soaring high with a vertical stoop, narrowed exploration by gliding with a contour flight, expanded exploitation by low-flying descent attack, and narrowed exploitation by walking and catching prey. In order to transit from the exploration stage to the exploitation stage, the AO algorithm uses a variety of behaviors. The exploration stage is simulated in the first two-thirds of iterations, while the exploitation stage is imitated in the last one-third of iterations. The following is a mathematical representation of the AO algorithm.

Initializing: The AO algorithm begins by spreading N solutions in a D -dimensional search space across a preset range $[L, U]$ by applying Equation (1).

$$X_{i,j} = L_j + r \times (U_j - L_j) \quad (1)$$

where $X_{i,j}$ is the j -th dimension of the i -th solution, L_j and U_j refer to the lower and upper bound value of the j -th dimension in the search space, and r is chosen at random from the range of 0 to 1. The position of solutions is kept in matrix $X_{N \times D}$. Then, by $f(X_i)$, the fitness value of each solution is calculated.

Expanded exploration: An Aquila first determines the prey region and picks the optimal hunting location by high-soaring while stooping vertically. This behavior leads to the search space being explored from high altitudes to estimate where the prey can be located. In AO, this behavior is simulated to expand the exploration by Equation (2) and is executed when $iter < (2/3 \times MaxIter)$ and randomly generated value < 0.5 ,

$$X_1(iter + 1) = X_{best}(iter) \times (1 - \frac{iter}{MaxIter}) + (X_M(iter) - X_{best}(iter) \times r) \quad (2)$$

where $X_1(iter + 1)$ is the solution given by the prime method to use in the subsequent iteration and $X_{best}(iter)$ is the best solution found until the current iteration and approximates the position of prey. The $(1 - \frac{iter}{MaxIter})$ term is utilized to regulate the extent of the exploration based on the number of iterations, where $iter$ denotes the current iteration and $MaxIter$ is the number of iterations that can be performed. In the $iter$ -th iteration, $X_M(iter)$ indicates the mean of currently available solutions, as determined by Equation (3),

$$X_M(iter) = \frac{1}{N} \sum_{i=1}^N X_i(iter), \forall j = 1, 2, \dots, D \quad (3)$$

where D denotes the dimension size of the search space and N represents the number of solutions.

Narrowed exploration: In the second step, the hunting behavior named contour flight of a short glide attack is performed. Aquila flies over the targeted prey, prepares to descend, and attacks when spotted from a high height. This behavior allows the Aquila to explore a specific region narrowly. In AO, this behavior is simulated to narrow the exploration by Equation (4) and is done when $iter < (2/3 \times MaxIter)$ and randomly generated value > 0.5 ,

$$X_2(iter + 1) = X_{best}(iter) \times Levy(D) + X_R(iter) + (y - x) \times r \quad (4)$$

where $X_2(iter + 1)$, $X_R(iter)$, and $Levy(D)$ represent the solutions produced by the narrowed exploration strategy, a randomly selected solution from entire solutions in the $iter$ -th iteration, and the Levy flight distribution function calculated by Equation (5), respectively.

$$Levy(D) = s \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left(\frac{\Gamma(1 + \beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}} \right) \quad (5)$$

where $s = 0.01$, $\beta = 1.5$, and u and v are random integer numbers ranging in $[0, 1]$. In Equation (4), the spiral form is represented by y and x calculated using Equation (6),

$$y = p \times \cos(\theta), \quad x = p \times \sin(\theta) \quad (6)$$

where r and θ are calculated by Equations (7) and (8),

$$p = r_1 + U \times D_1 \quad (7)$$

$$\theta = -\omega \times D_1 + \theta_1, \quad \theta_1 = \frac{3 \times \pi}{2} \quad (8)$$

where the number of search cycles is fixed by the random number r_1 which is between 1 and 20, D_1 is an integer value in the range of 1 to D , $U = 0.00565$, and $\omega = 0.005$.

Expanded exploitation: Aquila employs the third strategy to pursue prey during the expanded exploitation step. The Aquila has carefully identified the prey zone and is prepared to alight and attack. In order to determine how the prey would respond to the attack, the Aquila descends vertically and performs the first strike. This behavior is named low-flying descent attack and is performed when $iter > (2/3 \times MaxIter)$ and randomly generated value < 0.5 by Equation (9).

$$X_3(iter + 1) = (X_{best}(iter) - X_M(iter)) \times \alpha - r + ((U - L) \times r + L) \times \delta \quad (9)$$

where $X_3(iter + 1)$ denotes the solutions obtained by the expanded exploitation method and the exploitation adjustment parameters α and δ are set to 0.1.

Narrowed exploitation: The fourth hunting strategy is used during the narrower exploitation step when the Aquila approaches the prey and attacks randomly. This behavior is called walking and grabbing the prey and is done when $iter > (2/3 \times MaxIter)$ and randomly generated value > 0.5 by Equation (10).

$$X_4(iter + 1) = QF(iter) \times X_{best}(iter) - (G_1 \times X(iter) \times r) - G_2 \times Levy(D) + r \times G_1 \quad (10)$$

where $X_4(iter + 1)$ denotes the generated fourth search solutions, $X(iter)$ is the $iter$ -th iteration's current solution, and to balance the search strategy, a quality function called QF is calculated by Equation (11).

$$QF(iter) = t^{\frac{2 \times r - 1}{(1 - MaxIter)^2}} \quad (11)$$

G_1 and G_2 are values to represent the Aquila's prey tracking movements such that the G_2 's value is decreasing from 2 to 0. The calculation of G_1 and G_2 are done by Equations (12) and (13).

$$G_1 = 2 \times r - 1 \quad (12)$$

$$G_2 = 2 \times \left(1 - \frac{iter}{MaxIter}\right) \quad (13)$$

The AO algorithm is inspired by Aquila's natural behavior and shown a competitive performance when adapted to solve optimization problems [113]. The algorithm's global exploration ability is shown by the hunting strategies for fast-moving prey, while the algorithm's local exploitation capability is demonstrated by the hunting approaches for slow-moving prey. The AO algorithm has a high search efficiency, fast convergence speed, and good global exploration but lacks local exploitation, making it susceptible to get trapped in the local optimum [114]. The BAO's major goal is to tackle the restriction of the continuous AO to solve feature selection problems by utilizing its binary form.

4. Binary Aquila Optimizer (BAO) Algorithm

A binary optimization problem's search space can be considered a hypercube, which allows an individual to move from one point to another by altering one or more bits of its

position. Since binary space contains only two values, “0” and “1”, the position updating for binary optimization issues such as feature selection cannot be achieved by continuous strategies. Transfer functions are one of the key components of metaheuristic-based feature selection algorithms, in which the continuous search space is mapped to the discrete space. Transfer functions are used to assess the probability of altering the elements of a position vector to 0 or 1 based on the value of the i -th solution’s vector in the d -th dimension. S-shaped and V-shaped transfer functions are the two most prevalent forms of transfer functions [98,99] such that continuous metaheuristic algorithms can be discretized and utilized to solve binary optimization problems by converting a real vector into a binary vector. In Table 1 and Figure 1, formulation and visual representation of these two families of transfer functions are provided.

Table 1. List of transfer functions.

S-Shaped Transfer Functions		V-Shaped Transfer Functions	
Name	Function	Name	Function
S1	$T(x) = \frac{1}{1+e^{-2(x)}}$	V1	$T(x) = \left \frac{\sqrt{2}}{\pi} \int_0^{(\sqrt{\pi}/2)x} e^{-t^2} dt \right $
S2	$T(x) = \frac{1}{1+e^{-x}}$	V2	$T(x) = \tanh(x) $
S3	$T(x) = \frac{1}{1+e^{(-x/2)}}$	V3	$T(x) = (x)\sqrt{1+x^2} $
S4	$T(x) = \frac{1}{1+e^{(-x/3)}}$	V4	$T(x) = \left \frac{2}{\pi} \arctan\left(\frac{\pi}{2}\right)x \right $

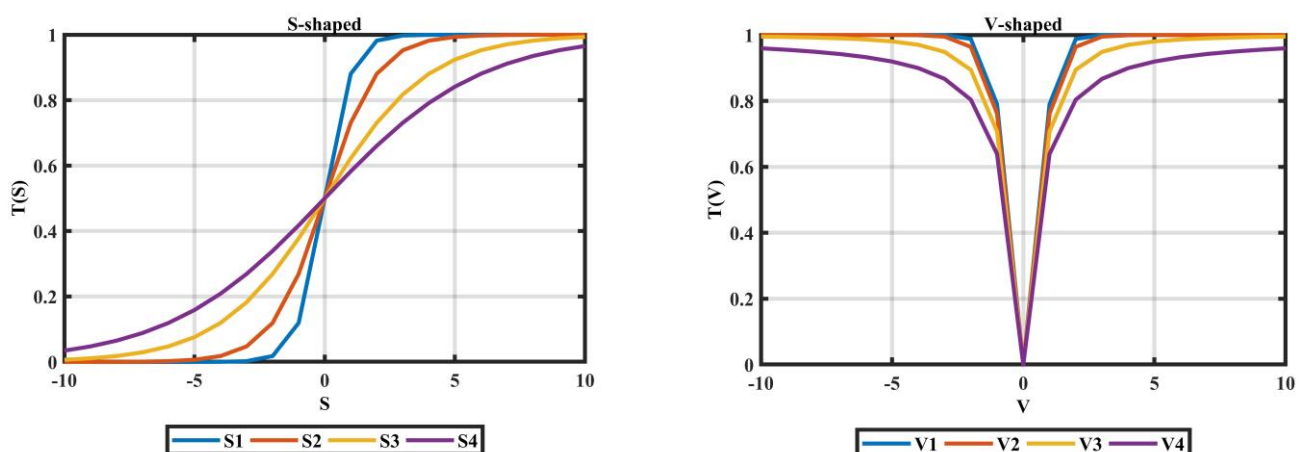


Figure 1. S-shaped and V-shaped transfer functions.

Although the AO algorithm was initially suggested to solve continuous optimization issues, it has to be adapted to accommodate the binary challenge of feature selection. Therefore, in order to solve binary problems through the use of AO, a transfer function needs to be used to convert a solution vector with continuous values into a probability vector. The transfer functions of both families are utilized to adapt continuous AO to binary variants called SBAO and VBAO in this study. Specifically, eight different transfer functions are considered, which result in eight distinct BAO variations.

4.1. S-Shaped Binary Aquila Optimizer (SBAO) Algorithm

The suggested variants of the BAO algorithm treat the search space as a continuous space where every solution has a real-valued position vector. The continuous values in the search space must be mapped to binary values using our proposed algorithms to derive a binary position vector for the solution. Each dimension of the position is taken into account by applying a particular S-shaped transfer function to compel the solution to move around in binary space. Using the floating-point position values, the transfer function calculates a limited probability in the interval [0, 1] for each solution. A floating-point

vector is then converted to a bit-string position vector based on the obtained probabilities. In Equation (14), the S-shaped function is given for generating probability value.

$$S(X_i^d(iter + 1)) = \frac{1}{1 + e^{(-X_i^d(iter)/3)}} \quad (14)$$

The value of $S(X_i^d(iter + 1))$ denotes the possibility of altering the i -th solution's binary position value in the d -th dimension. After comparing the probability to a threshold value, the binary value is calculated, as shown in Equation (15), where *rand* is a random value between 0 and 1.

$$X_i^d(iter + 1) = \begin{cases} 1 & , \text{if } rand \leq S(X_i^d(iter + 1)) \\ 0 & , \text{otherwise} \end{cases} \quad (15)$$

4.2. V-Shaped Binary Aquila Optimizer (VBAO) Algorithm

The V-shaped transfer function is another function used to calculate the possibility of changing positions. Similar to the S-shaped transfer function, the V-shaped transfer function is also used to calculate the probability of changing the search agent's location by Equation (16).

$$V(X_i^d(iter + 1)) = \left| \frac{2}{\pi} \arctan\left(\frac{\pi}{2}\right)(X_i^d(iter)) \right| \quad (16)$$

Following the changing probability values calculation, each search agent's binary position vector is updated using an updating position equation, as shown in Equation (17), where *rand* is a random value between 0 and 1.

$$X_i^d(iter + 1) = \begin{cases} \text{complement}(X_i^d(iter)) & , \text{if } rand \leq V(X_i^d(iter + 1)) \\ X_i^d(iter) & , \text{otherwise} \end{cases} \quad (17)$$

The continuous Aquila Optimizer (AO) is converted into its binary variant (BAO) by transforming each search agent's dimension to a probability value ranging from 0 to 1 as given in Equation (2). This is done utilizing all variations of S-shaped and V-shaped transfer functions. Therefore, the probability of changing each search agent's position is calculated through the use of either S-shaped or V-shaped transfer functions shown by Equations (14) and (16). Then, the binary position of the search agent is obtained by considering the calculated changing probability value using Equation (15) or (17). To identify the appropriate transfer function, eight different variants of the suggested BAO are examined since transforming a continuous search space to a binary domain greatly affects the performance and results of classifiers. The pseudo-code of the proposed BAO algorithm is shown in Algorithm 1. BAO has an $O(NDT)$ computational complexity, where N , D , and T are the population size, the number of features, and the maximum number of iterations, respectively.

Algorithm 1. The binary Aquila optimizer (BAO)

Input: N (population size), D (the dimension's number), $MaxIter$ (maximum number of iterations)
Output: The best solution (X_{best})

```

1: Begin
2: Initializing  $iter = 1$ ,  $\alpha = 0.1$ ,  $\delta = 0.1$ .
3: Generating a random initial population  $X$ .
4: While  $it \leq MaxIter$ 
5:   Evaluating the fitness function values and set the  $X_{best}(it)$ .
6:   If  $iter < (2/3) \times MaxIter$ 
7:     If  $rand < 0.5$ 
8:       Calculating  $X_1(iter + 1)$  using Equation (2).
9:       Updating  $X(iter + 1)$  and  $X_{best}(iter)$ .
10:    else
11:      Calculating  $X_2(iter + 1)$  using Equation (4).
12:      Updating  $X(iter + 1)$  and  $X_{best}(iter)$ .
13:    End if
14:  else
15:    If  $rand < 0.5$  then
16:      Calculating  $X_3(iter + 1)$  using Equation (9).
17:      Updating  $X(iter + 1)$  and  $X_{best}(iter)$ .
18:    Else
19:      Calculating  $X_4(iter + 1)$  using Equation (10).
20:      Updating  $X(iter + 1)$  and  $X_{best}(iter)$ .
21:    End if
22:  End if
23:  Calculating the probability values using Equation (14) or (16).
24:  Updating binary position.
25:   $iter = iter + 1$ .
26: End while
27: Return the best solution ( $X_{best}$ ).
28: End

```

5. Binary Aquila Optimizer Algorithm for Feature Selection Problem

Feature selection entails identifying relevant features in a dataset to enhance learning capabilities, reduce computation complexity, and improve classification performance. The optimum feature subset is obtained using a binary algorithm depending on the nature of the feature selection problem. Using the binary approach, each solution is represented by binary vectors with D entries, reflecting the number of features in a dataset. Each entry of the solution vector has a value of 0 or 1, where 0 signifies no selection and 1 indicates the selection of that particular feature. The feature selection problem is addressed by utilizing two binary variants of the AO algorithm. As a multi-objective problem, the problem of feature selection requires the fulfillment of two conflicting objectives. This is a conflict between objectively maximizing accuracy and minimizing the selection of features. The weighted sum multi-objective fitness function for evaluating each solution is shown in Equation (18).

$$Fitness = \alpha E_R(D) + \beta \frac{|R|}{|C|} \quad (18)$$

where α and β are two factors that represent the weight of accuracy and the number of selected features and their values are set in the range of $\alpha \in [0, 1]$ and $\beta = 1 - \alpha$ [105]. The classification error, the number of chosen features, and the total number of features are represented by $E_R(D)$, $|R|$, and $|C|$, respectively.

6. Experimental Evaluation and Results

An evaluation of the performance of variants of SBAO and VBAO algorithms is presented in this section by presenting detailed experimental results and statistical analysis.

To assess the efficiency of the BAO algorithms, these experiments include performance evaluation and convergence evaluation. In the comparative experiment, the gained results are assessed along with the state-of-the-art and recently developed nature-inspired algorithms consisting of binary bat algorithm (BBA) [115], binary gravitational search algorithm (BGSA) [104], binary grey wolf optimization (BGWO) [105], binary dragonfly algorithm (BDA) [116], S-shaped binary sine cosine algorithm (SBSCA) [117], and V-shaped binary sine cosine algorithm (VBSCA) [117]. Moreover, a non-parametric statistical test, known as the Friedman test [118], is used to demonstrate the significance of the difference between the gained results and comparative algorithms. Additionally, we performed an experiment on the COVID-19 dataset and compared the gained results by comparative algorithms.

6.1. Datasets Description and Experimental Environment

The proposed BAO algorithms were validated on seven medical datasets selected from [119,120] with varying numbers of features and instances. A detailed description of each dataset is provided in Table 2, including the number of instances, features, and classes. As part of the evaluation process, each dataset was split into two sets: a training set and a testing set, with 80% of the instances used for training and the remainder for testing. In this wrapper approach, the k-nearest neighbors (k-NN) method is employed to estimate the classification error rate of the selected feature subset. The proposed algorithms were developed by Matlab programming environment R2014b, and all experiments were run on a CPU, Intel Core(TM) i7-3770 at 3.4 GHz and 8.00 GB RAM.

Table 2. Statistical information of datasets.

Dataset	No. of Instances	No. of Features	No. of Classes
Heart	270	14	2
Breast Cancer	683	10	2
Pima	768	9	2
Breast-WDBC	569	31	2
Lymphography	148	19	4
Colon	62	2000	2
Leukemia	72	7129	2

6.2. Experimental Setup

Throughout this work, all of the comparative algorithms' parameters were set as in their original works, as indicated in Table 3. The appropriate values for the parameters α and δ are found by some pretests reported in Appendix A. To gain meaningful results, all experiments were done and evaluated by 30 separate runs. All algorithms' initial settings were set to the same maximum iterations (*MaxIter*) and population size (*N*) to 300 and 20 to ensure fair comparisons. The value of α and β parameters in the fitness function in Equation (18) are set by 0.99 and 0.01, and the value of parameter *k* is set with 5 in the k-NN classifier, respectively. As an evaluation criterion, all algorithms use classification accuracy and the number of selected features. The algorithms' performance was measured using mean and standard deviation (SD) of accuracy, the number of selected features, and the gained fitness.

6.3. Performance Evaluation of SBAO and VBAO

In this subsection, seven medical datasets are used to benchmark various variants of the proposed BAO to determine the impact of different transfer functions on the proposed algorithm's performance. The BAO variants utilize eight transfer functions listed in Table 1. The efficacy of the variants is evaluated by the mean and standard deviation of accuracy, the number of selected features, and the gained fitness. Tables 4–6 provide a summarization of the gained results achieved by the algorithms. In Table 4, the mean and SD for the

accuracy of eight variants of BAO with S-shaped and V-shaped transfer functions in 30 runs are tabulated. As per results in Table 4, the variants SBAO-2, SBAO-3, SBAO-4, and VBAO-1, VBAO-3, and VBAO-4 provided the highest accuracy in the Pima dataset. All the variants gained the same and superior accuracy in the Breast cancer dataset. In Heart and Lymphography datasets, the BAO versions that utilize transfer functions S4 and V4 can achieve better accuracy than the other variants. SBAO-3 and VBAO-3 outperform other variants in terms of accuracy in the Breast-WDBC dataset. In the Colon dataset, SBAO-1, SBAO-2, and VBAO-1 and in the Leukemia dataset, SBAO-2 and VBAO-1 obtain the best results that show the binary AO's performance in selecting features from large datasets.

Table 3. Algorithms parameter settings.

Algorithm	Parameter	Value
BSCA	a	2
BBA	A	0.5
	r	0.5
	Q_{\min}	0
	Q_{\max}	2
BGSA	G_0	100
BGWO	a	[2 0]
BAO	α and δ	0.1

Table 4. The accuracy comparison of SBAO and VBAO.

Dataset	Metric	SBAO-1	SBAO-2	SBAO-3	SBAO-4	VBAO-1	VBAO-2	VBAO-3	VBAO-4
Pima	Mean acc.	0.772	0.773	0.773	0.773	0.773	0.772	0.773	0.773
	SD acc.	0.001	0.000	0.000	0.000	0.000	0.003	0.000	0.000
Breast Cancer	Mean acc.	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	SD acc.	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Heart	Mean acc.	0.894	0.896	0.894	0.898	0.887	0.885	0.892	0.896
	SD acc.	0.009	0.009	0.009	0.009	0.010	0.014	0.011	0.012
Lymphography	Mean acc.	0.862	0.864	0.867	0.871	0.862	0.860	0.867	0.869
	SD acc.	0.024	0.026	0.028	0.024	0.024	0.025	0.030	0.015
Breast-WDBC	Mean acc.	0.967	0.967	0.967	0.967	0.964	0.964	0.965	0.965
	SD acc.	0.004	0.004	0.004	0.004	0.002	0.002	0.002	0.000
Colon	Mean acc.	0.950	0.950	0.925	0.925	0.875	0.858	0.850	0.858
	SD acc.	0.056	0.041	0.025	0.045	0.042	0.038	0.033	0.038
Leukemia	Mean acc.	0.9714	0.9929	0.9786	0.9571	0.9429	0.9357	0.9286	0.9357
	SD acc.	0.0356	0.0218	0.0333	0.0356	0.0436	0.0391	0.0325	0.0218

As tabulated in Table 5, the mean and standard deviation of selected features are calculated for 30 independent runs using BAO variants. It can be observed that SBAO-1 and VBAO-2 can be achieved better results in the Pima dataset. The BAO with S-shaped and V-shaped transfer functions except the fourth S-shaped function have achieved the same best number of selected features in the Breast cancer dataset. In the Heart dataset, both transfer functions S1 and V1 in BAO achieved the minimum selection of features. For the Lymphography dataset, SBAO-1, SBAO-3, and SBAO-4 have shown equal performance, whereas VBAO-1 has better performance in comparison to other V-shaped variants. SBAO-2 and VBAO-4 achieved better performance for the Breast-WDBC dataset. In the Colon dataset, SBAO-4 and VBAO-2 and in Leukemia, SBAO-2 and VBAO-1 had better performance than other variants. Table 6 demonstrates the proposed algorithms' results

regarding the mean and SD of the gained fitness values. The results demonstrate the impact of utilizing various transfer functions on the BAO's performance.

Table 5. The number of selected features comparison of SBAO and VBAO.

Dataset	Metric	SBAO-1	SBAO-2	SBAO-3	SBAO-4	VBAO-1	VBAO-2	VBAO-3	VBAO-4
Pima	Mean	4.97	5.00	5.00	5.00	5.00	4.97	5.00	5.00
	SD	0.18	0.00	0.00	0.00	0.00	0.18	0.00	0.00
Breast Cancer	Mean	3.00	3.00	3.00	3.07	3.00	3.00	3.00	3.00
	SD	0.00	0.00	0.00	0.25	0.00	0.00	0.00	0.00
Heart	Mean	4.90	5.60	5.00	5.50	5.00	5.10	5.40	6.30
	SD	1.40	1.45	1.36	1.53	1.20	1.32	1.30	1.02
Lymphography	Mean	5.73	6.27	5.73	5.73	7.20	8.07	7.73	7.47
	SD	1.95	2.18	1.36	1.20	1.45	2.12	1.84	1.57
Breast-WDBC	Mean	5.00	4.13	4.53	4.87	9.80	10.13	9.80	9.07
	SD	2.03	1.22	1.74	1.70	2.41	1.85	1.49	1.08
Colon	Mean	120.10	118.10	133.80	107.60	47.50	35.70	85.60	134.00
	SD	202.62	281.94	342.86	165.75	55.20	27.25	83.63	108.07
Leukemia	Mean	1316.70	1151.80	1739.80	2111.30	228.50	1151.90	1457.60	2651.80
	SD	520.27	1300.27	1220.66	1177.89	161.25	1321.51	1483.56	1099.86

Table 6. The fitness comparison of SBAO and VBAO.

Dataset	Metric	SBAO-1	SBAO-2	SBAO-3	SBAO-4	VBAO-1	VBAO-2	VBAO-3	VBAO-4
Pima	Mean fitness	0.231	0.231	0.231	0.231	0.231	0.231	0.231	0.231
	SD fitness	0.001	0.000	0.000	0.000	0.000	0.002	0.000	0.000
Breast Cancer	Mean fitness	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003
	SD fitness	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Heart	Mean fitness	0.108	0.107	0.108	0.105	0.115	0.117	0.110	0.107
	SD fitness	0.007	0.008	0.007	0.008	0.009	0.013	0.010	0.011
Lymphography	Mean fitness	0.139	0.137	0.135	0.130	0.140	0.143	0.136	0.133
	SD fitness	0.023	0.025	0.026	0.023	0.023	0.024	0.029	0.014
Breast-WDBC	Mean fitness	0.034	0.034	0.033	0.034	0.038	0.038	0.037	0.037
	SD fitness	0.004	0.003	0.004	0.003	0.002	0.002	0.002	0.000
Colon	Mean fitness	0.050	0.050	0.074	0.074	0.124	0.140	0.148	0.140
	SD fitness	0.055	0.041	0.025	0.045	0.041	0.038	0.033	0.038
Leukemia	Mean fitness	0.030	0.008	0.023	0.045	0.056	0.065	0.072	0.067
	SD fitness	0.034	0.021	0.032	0.034	0.043	0.039	0.032	0.022

6.4. Performance Comparison with State-of-the-Art Algorithms

In this subsection, the obtained results of the proposed binary versions of the AO, SBAO, and VBAO for each dataset are compared with other binary state-of-the-art algorithms that are widely used to solve the feature selection problem. To compare with other comparative algorithms, for each dataset, one variant of S-shaped and V-shaped transfer functions was selected such that the variant provides better performance in terms of accuracy. This selection of SBAO and VBAO for each dataset is as follows: Pima (SBAO-1,

VBAO-2), Breast cancer (SBAO-1, VBAO-1), Heart (SBAO-4, VBAO-4), Lymphography (SBAO-4, VBAO-4), Breast-WDBC (SBAO-3, VBAO-3), Colon (SBAO-2, VBAO-1), and Leukemia (SBAO-2, VBAO-1). The experimental results are shown in Tables 7–9, in which the best-obtained results are remarked in boldface. The Friedman test [118] was used to examine the accuracy gained by SBAO, VBAO, and comparative algorithms. A comparison of binary AO and comparative algorithms was also performed in order to determine their convergence behavior.

Table 7. The accuracy comparison of SBAO and VBAO with other binary metaheuristic algorithms.

Dataset	Metric	BBA	BGSA	BGWO	BDA	SBSCA	VBSCA	SBAO	VBAO
Pima	Mean acc.	0.754	0.773	0.766	0.669	0.773	0.773	0.773	0.773
	SD acc.	0.012	0.000	0.009	0.112	0.000	0.000	0.000	0.000
Breast cancer	Mean acc.	0.998	1.000	0.999	0.865	1.000	1.000	1.000	1.000
	SD acc.	0.003	0.000	0.001	0.103	0.000	0.000	0.000	0.000
Heart	Mean acc.	0.852	0.877	0.871	0.697	0.896	0.892	0.899	0.896
	SD acc.	0.017	0.009	0.025	0.237	0.009	0.007	0.009	0.012
Lymphography	Mean acc.	0.797	0.834	0.830	0.797	0.877	0.863	0.871	0.869
	SD acc.	0.023	0.020	0.026	0.217	0.025	0.020	0.024	0.015
Breast-WDBC	Mean acc.	0.951	0.959	0.953	0.955	0.967	0.965	0.967	0.965
	SD acc.	0.006	0.004	0.006	0.067	0.004	0.002	0.003	0.002
Colon	Mean acc.	0.766	0.788	0.794	0.866	0.877	0.838	0.950	0.875
	SD acc.	0.034	0.042	0.042	0.041	0.042	0.021	0.041	0.042
Leukemia	Mean acc.	0.831	0.864	0.888	0.971	0.971	0.914	0.993	0.942
	SD acc.	0.078	0.047	0.064	0.035	0.035	0.029	0.021	0.043
Friedman test-Average rank		7.50	5.00	6.14	6.29	2.21	3.93	1.79	3.14
Overall rank		8	5	6	7	2	4	1	3

Table 8. The number of selected features comparison of SBAO and VBAO with comparative algorithms.

Dataset	Metric	BBA	BGSA	BGWO	BDA	SBSCA	VBSCA	SBAO	VBAO
Pima	Mean	3.00	5.00	5.10	5.00	5.00	5.00	5.00	5.00
	SD	1.53	0.00	0.31	0.00	0.00	0.00	0.00	0.00
Breast Cancer	Mean	3.27	3.20	4.27	3.00	3.00	3.00	3.00	3.00
	SD	1.41	0.41	1.08	0.00	0.00	0.00	0.00	0.00
Heart	Mean	5.07	4.97	7.03	5.33	5.27	5.27	5.50	6.30
	SD	2.07	1.16	0.76	1.42	1.46	1.41	1.53	1.02
Lymphography	Mean	6.33	7.63	9.73	6.03	6.13	7.23	5.73	7.47
	SD	3.07	1.73	2.21	1.33	1.55	2.06	1.20	1.57
Breast-WDBC	Mean	11.10	12.77	11.93	4.27	4.20	9.33	4.53	9.80
	SD	3.39	2.51	2.46	1.14	1.06	2.25	1.74	1.49
Colon	Mean	687.00	973.03	1061.60	738.23	917.37	109.90	118.10	47.50
	SD	84.99	28.66	75.54	54.02	21.55	65.74	281.94	55.20
Leukemia	Mean	2520.17	3537.93	4585.00	3032.17	3426.40	1579.40	1151.80	228.50
	SD	325.76	41.25	340.83	217.44	44.33	996.49	1300.27	161.25
Friedman test-Average rank		4.00	5.79	7.86	3.79	3.86	3.57	3.21	3.93
Overall rank		6	7	8	3	4	2	1	5

Table 9. The fitness comparison of SBAO and VBAO with other binary metaheuristic algorithms.

Dataset	Metric	BBA	BGSA	BGWO	BDA	SBSCA	VBSCA	SBAO	VBAO
Pima	Mean fitness	0.249	0.231	0.237	0.330	0.231	0.231	0.231	0.231
	SD fitness	0.011	0.000	0.009	0.108	0.000	0.000	0.000	0.002
Breast Cancer	Mean fitness	0.005	0.003	0.005	0.135	0.003	0.003	0.003	0.003
	SD fitness	0.003	0.000	0.001	0.102	0.000	0.000	0.000	0.000
Heart	Mean fitness	0.149	0.125	0.132	0.302	0.106	0.110	0.105	0.107
	SD fitness	0.017	0.008	0.025	0.233	0.008	0.006	0.008	0.011
Lymphography	Mean fitness	0.204	0.168	0.173	0.219	0.125	0.139	0.130	0.133
	SD fitness	0.022	0.020	0.026	0.256	0.024	0.019	0.023	0.014
Breast-WDBC	Mean fitness	0.050	0.044	0.050	0.045	0.033	0.037	0.033	0.037
	SD fitness	0.005	0.004	0.006	0.067	0.004	0.002	0.004	0.002
Colon	Mean fitness	0.233	0.213	0.208	0.135	0.125	0.160	0.050	0.124
	SD fitness	0.033	0.041	0.041	0.041	0.041	0.020	0.041	0.041
Leukemia	Mean fitness	0.170	0.139	0.117	0.032	0.033	0.087	0.008	0.056
	SD fitness	0.077	0.046	0.063	0.035	0.035	0.027	0.021	0.043

As demonstrated in Table 7, the gained results of SBAO, VBAO, and comparative algorithms on the basis of mean and standard deviation of the estimated accuracy for 30 separate runs on each of seven datasets are summarized. The BAO algorithm produces superior and competitive results on the datasets evaluated, as shown in Table 7. In the Pima and Breast cancer datasets, BAO produces the same results as BGSA, SBSCA, and VBSCA, while outperforming the rest of the algorithms on the Heart, Colon, and Leukemia datasets. According to the results, the proposed BAO algorithm was significantly more accurate than the comparative algorithms.

In Figure 2, the proposed BAO algorithm compares to the comparative algorithms in terms of the best-obtained classification accuracy in 30 runs. Additionally, the mean and standard deviation for the number of selected features are shown in Table 8. The plotted results show how the BAO can explore the search space for the most effective feature subset with the greatest classification accuracy. The suggested BAO beats the comparative algorithms when it comes to determining the effective subset with the fewest number of features, as shown in Table 8. It can be observed that SBAO and VBAO, like BDA, SBSCA, and VBSCA, performed best on the Breast cancer dataset, but SBAO and VBAO performed better on the Lymphography, Colon, and Leukemia datasets.

Figures 3 and 4 show a comparison between the proposed BAO and the comparative algorithms on four small datasets Breast cancer, Heart, Lymphography, and Breast-WDBC, and large datasets Colon and Leukemia. These figures show the number of selected features by considering the best-achieved accuracy. Table 9 shows the gained results of the SBAO and VBAO compared to comparative algorithms in terms of fitness value. BAO's ability to intensively explore promising regions of feature space and intensively exploit near solutions is the reason for this performance.

The average fitness convergence curves obtained by the suggested variations of BAO and comparative algorithms are shown in Figure 5. For the proposed BAO and comparative algorithms, the convergence curves are based on the optimal fitness function as well as mean convergence curves. The minimum fitness functions' convergence curves illustrate the highly qualified performance of the proposed BAO algorithm. The plotted curves

demonstrate that BAO can develop more effective solutions and strike a more favorable balance between exploration and exploitation.

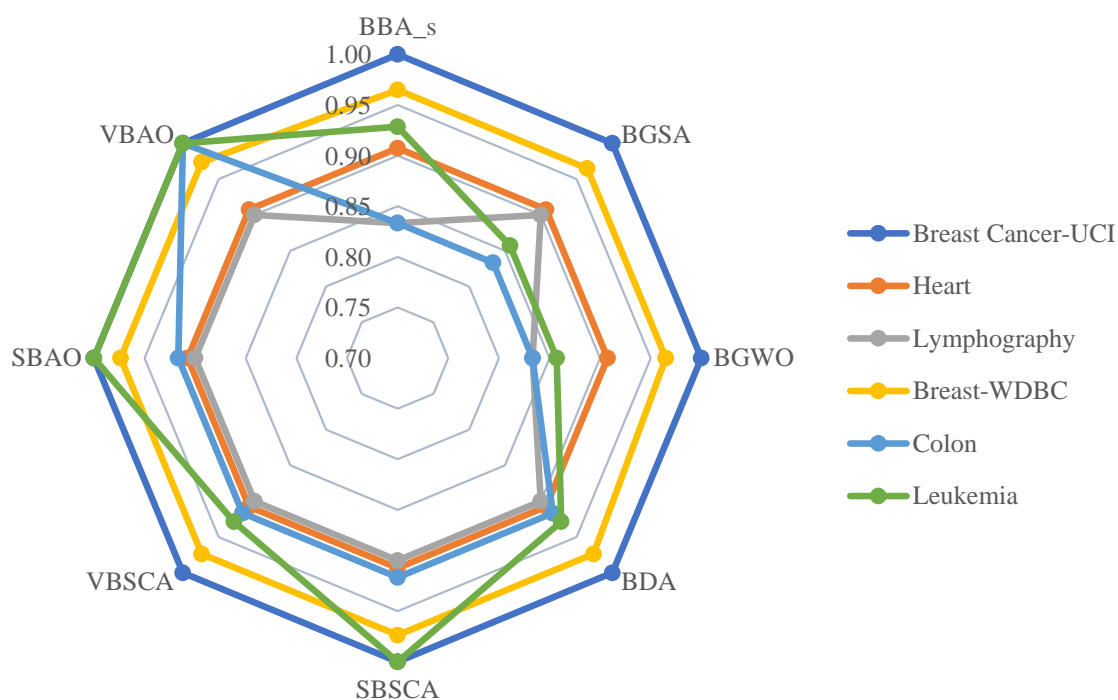


Figure 2. Average accuracy gained by BAO and comparative algorithms.

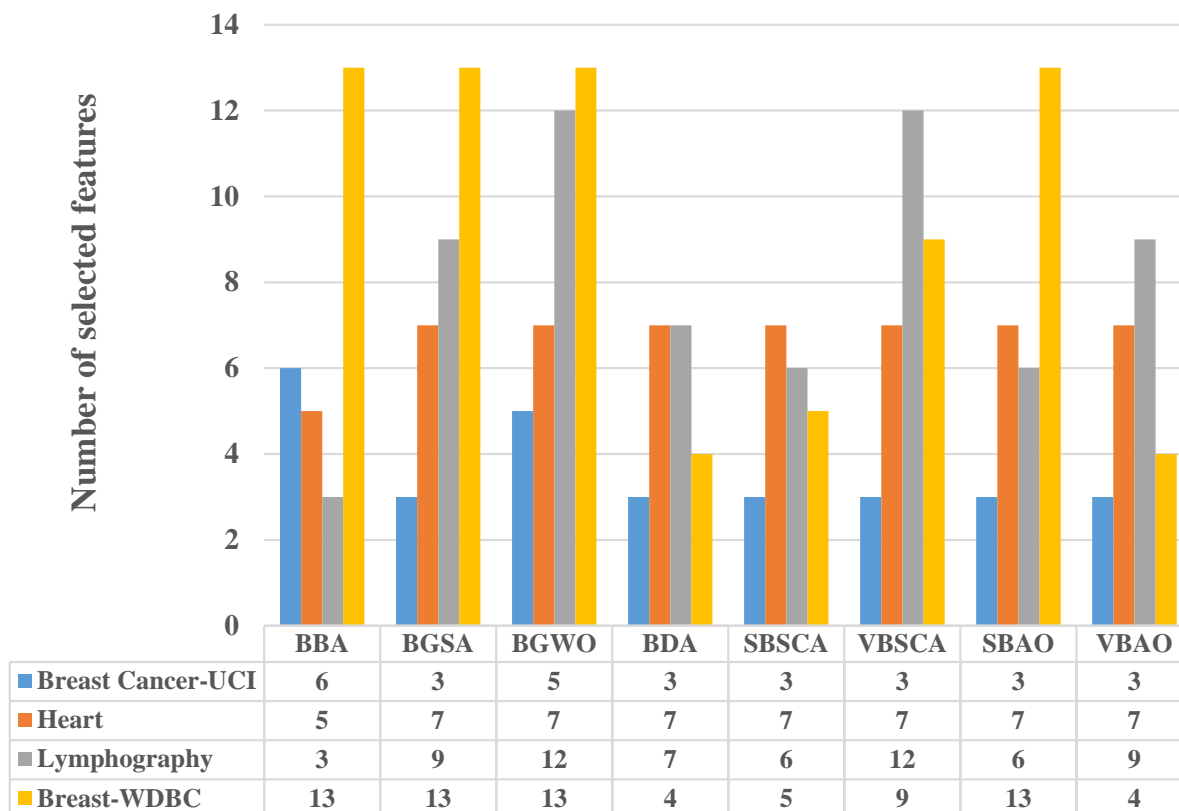


Figure 3. The Number of selected features comparison on small datasets.

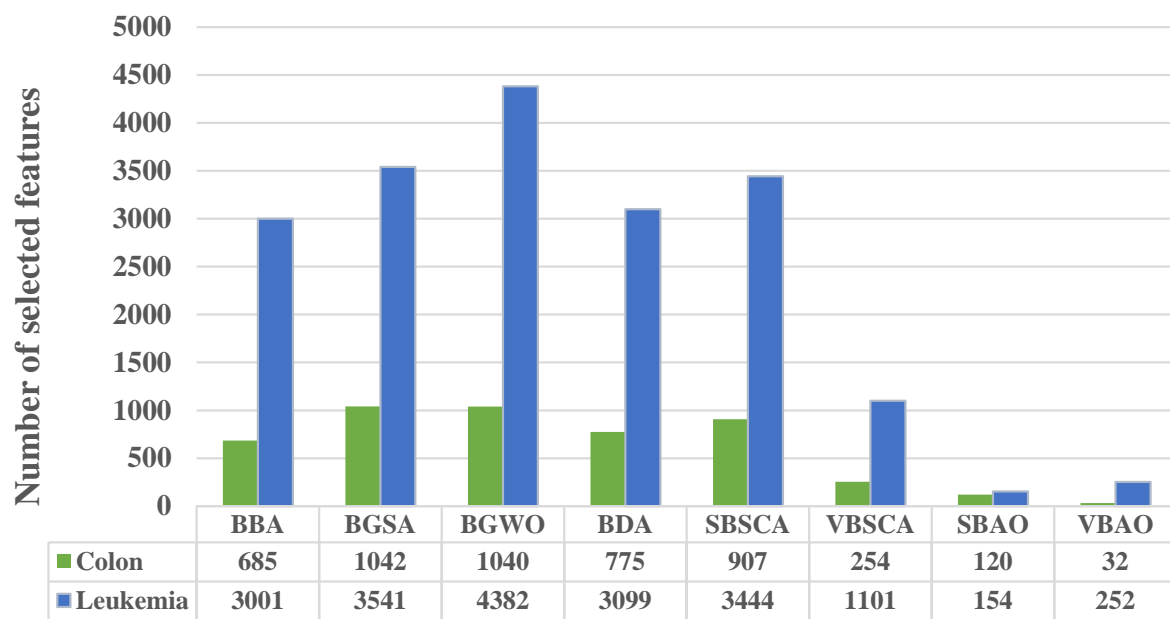


Figure 4. The Number of selected features comparison on large datasets.

Figure 5 shows with the exception of the Lymphography dataset, two variations of BAO discover superior fitness than other algorithms in all datasets. BGSA, unlike BAOs, shows a substantial decrease in fitness value in the early iterations, but the fitness value stayed constant for almost remained iterations, and eventually, it exhibits a significant loss in fitness value in the last iterations. While the BBA and BGWO algorithms demonstrate a steady convergence, the search process results in stagnation in non-optimal solutions. Finally, both SBSCA and VBSCA algorithms demonstrate a similar behavior to BAOs with a delayed convergence characteristic throughout the search phase.

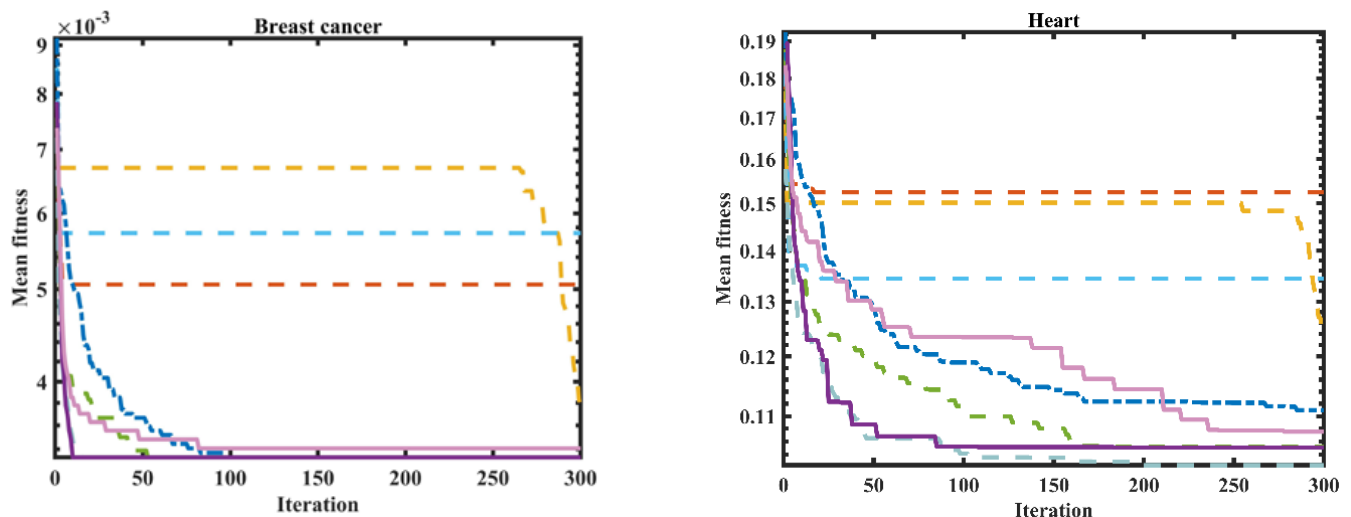


Figure 5. Cont.

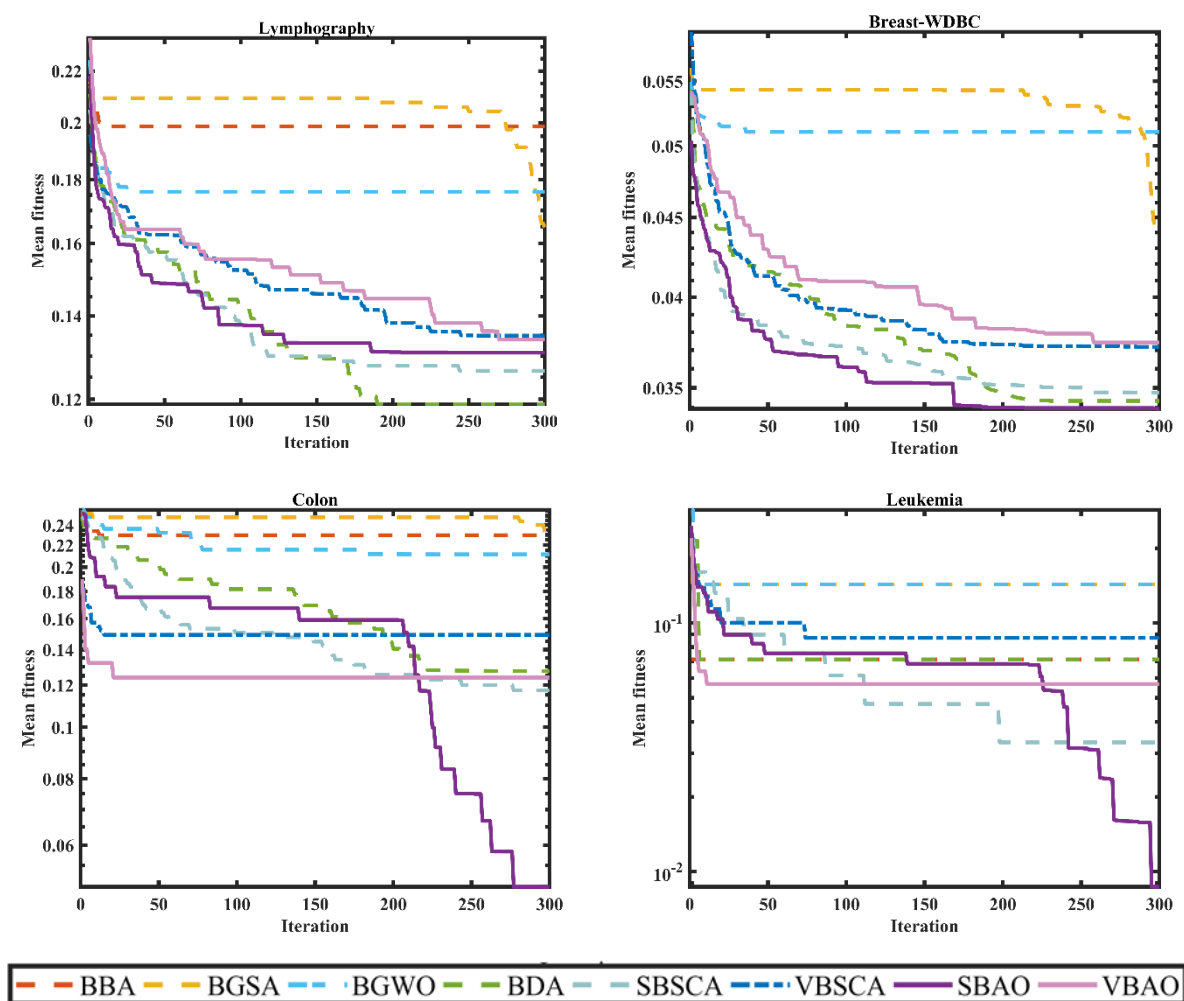


Figure 5. The convergence curves of BAO and comparative algorithms on all datasets.

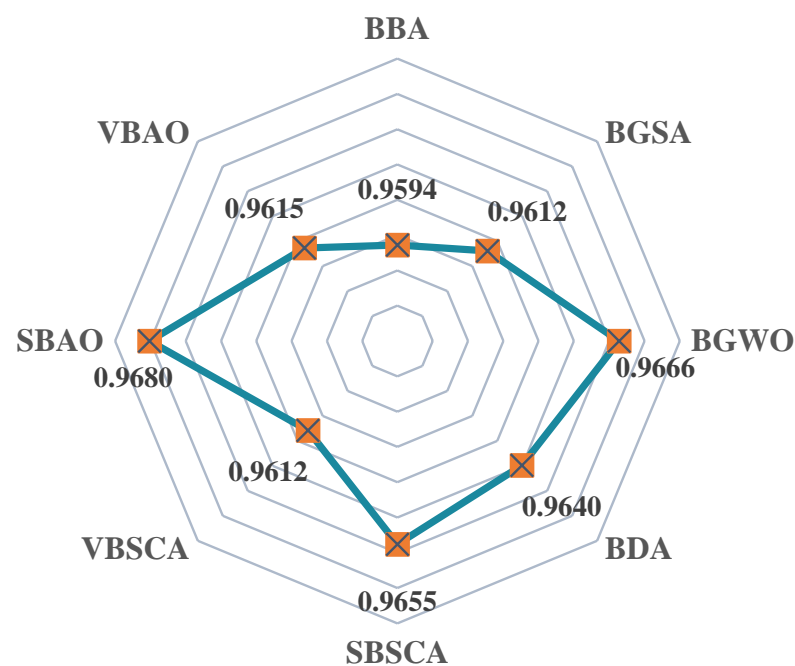
7. Using BAO for COVID-19 Case Study

The suggested BAO is used to diagnose COVID-19 patient health in this section. The COVID-19 patient dataset was obtained from [121]. The dataset includes 864 instances and 15 features, as shown in Table 10. This experiment aims to anticipate the patients' mortality and recovery status based on the supplied criteria. In the main dataset, patients whose death or recovery status is missing are removed. The feature id was removed from the dataset, and all features were transformed to numeric form. In this case, we use k-fold cross-validation with $K = 10$ for the validation process. Figures 6 and 7 demonstrate the accuracy and selected feature size of the proposed BAO and comparative algorithms on the COVID-19 dataset. SBAO, which took into account seven features, had the greatest classification accuracy of 96.80%. On the other hand, the gained results reveal that VBAO only needed three features to diagnose the patient's health.

The BAO's major goal is to overcome the restriction of the canonical continuous AO algorithm in solving the feature selection issue, the use of its binary form resolves this. The algorithm provides fast convergence speed, strong global exploration capabilities, and high search efficiency during the search process, resulting in a better performance than other algorithms. Furthermore, the BAO's performance demonstrates its capability to search the feature space for effective features while maintaining balance exploration and exploitation over several iterations.

Table 10. Description of COVID-19 dataset.

No.	Features	Description
1	Id	The patient's id
2	Location	The location where patient belongs to
3	Country	The country where patient belongs to
4	Gender	The patient's gender
5	Age	The patient's age
6	Sym-on	The date patient started noticing the symptoms
7	Hosp_vis	The date patient visited the hospital
8	Vis_wuhan	Whether the patient visited Wuhan, China
9	From_wuhan	Whether the patient from Wuhan, China
10	Symptom1	Symptom of patient (Fever)
11	Symptom2	Symptom of patient (Cough)
12	Symptom3	Symptom of patient (Cold)
13	Symptom4	Symptom of patient (Fatigue)
14	Symptom5	Symptom of patient (Body pain)
15	Symptom6	Symptom of patient (Malaise)
	Death	Whether the patient passed away due to COVID-19
	Recov	Whether the patient recovered

**Figure 6.** Accuracy gained by BAO and comparative algorithms.

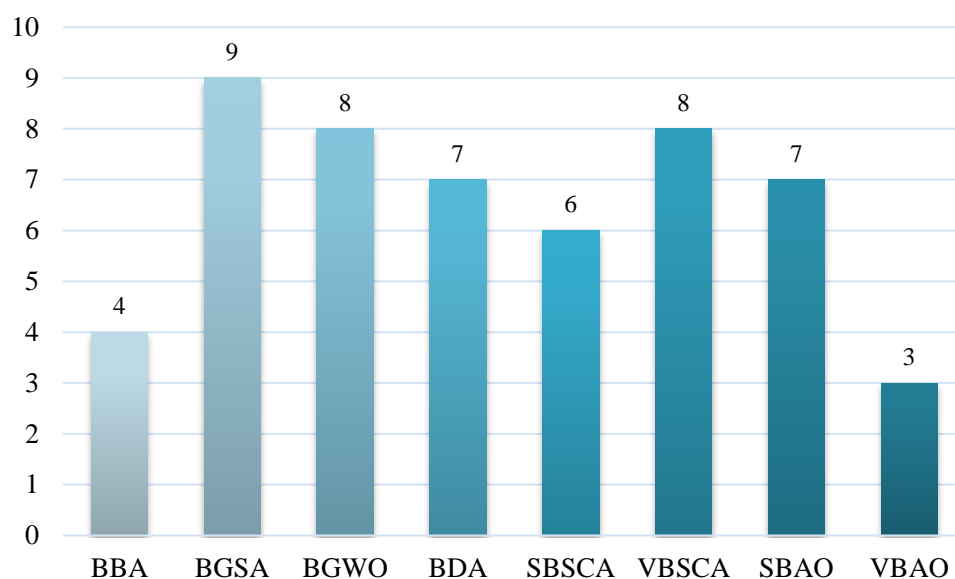


Figure 7. The number of selected features gained by BAO and comparative algorithms.

8. Conclusions and Future Works

In this article, two binary variations of the Aquila optimizer (AO) were introduced in this article and utilized to discover the wrapper approach's effective features. S-shaped and V-shaped transfer functions are utilized to alter the continuous version of AO into two binary algorithms, SBAO and VBAO. Then, the effective features from the medical datasets were selected for disease detection using the proposed algorithms. The gained results of the SBAO and VBAO were compared to six binary algorithms on seven medical datasets. The experimental findings demonstrate that the SBAO method can compete and/or achieve superior results on most datasets. In addition, the proposed algorithm was tested using the COVID-19 real-dataset. As a result of the findings, SBAO significantly outperforms other comparative algorithms and has shown to be more accurate than other comparative algorithms regarding predicting accuracy and minimizing the number of selected features. The BAO's performance over numerous iterations indicates its ability to find effective features while balancing exploration and exploitation. The algorithm provides speedy convergence, global exploration capabilities, and good search efficiency during the search process. In future research, the BAO variants can be applied to a variety of datasets and real-world situations by using various classifiers. Using the BAO to solve issues with many objectives might also be interesting.

Author Contributions: Conceptualization, M.H.N.-S. and S.T.; methodology, M.H.N.-S. and S.T.; software, M.H.N.-S. and S.T.; literature search, M.H.N.-S. and S.T.; validation, M.H.N.-S. and S.T.; formal analysis, M.H.N.-S. and S.T.; investigation, M.H.N.-S. and S.T.; resources, M.H.N.-S., S.T., S.M. and L.A.; data curation, M.H.N.-S., S.T., S.M. and L.A.; writing, M.H.N.-S. and S.T.; original draft preparation, M.H.N.-S. and S.T.; writing—review and editing, M.H.N.-S., S.T., S.M. and L.A.; visualization, M.H.N.-S. and S.T.; supervision, M.H.N.-S. and S.M.; project administration, M.H.N.-S. and 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: The data and code used in the research may be obtained from the corresponding author upon request.

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

Appendix A

The appropriate values for the parameters α and δ used in the BAO are found using some pretests on the Leukemia dataset. Table A1. shows the comparison of mean accuracy and selected number of features by the different variants of SBAO and VBAO using values from set {0.1, 0.5, and 0.9}. The results indicate that the BAO can achieve better performance when $\alpha = 0.1$ and $\delta = 0.1$.

Table A1. The pretests' results for tuning the parameters α and δ .

BAO Variants	Metric	$\alpha = 0.1$			$\alpha = 0.5$			$\alpha = 0.9$		
		$\delta = 0.1$	$\delta = 0.5$	$\delta = 0.9$	$\delta = 0.1$	$\delta = 0.5$	$\delta = 0.9$	$\delta = 0.1$	$\delta = 0.5$	$\delta = 0.9$
SBAO-1	Acc.	0.9714	0.9762	1.0000	1.0000	0.9762	1.0000	1.0000	0.9762	0.9762
	#SF.	1316.70	486.00	2147.33	1581.67	1318.66	1375.00	2236.33	1513.00	1809.67
SBAO-2	Acc.	0.9929	0.9762	1.0000	1.0000	0.9524	1.0000	1.0000	1.0000	0.9762
	#SF.	1151.80	3049.00	2079.00	1955.33	1310.667	1717.00	1738.00	1284.67	1516.00
SBAO-3	Acc.	0.9786	0.9524	0.9762	1.0000	1.0000	0.9762	0.9762	1.0000	0.9762
	#SF.	1739.80	1043.00	2858.67	2845.67	2703.667	1948.00	3050.00	2092.67	1791.67
SBAO-4	Acc.	0.9571	0.9762	0.9762	1.0000	1.0000	1.0000	1.0000	0.9762	0.9524
	#SF.	2111.30	2804.00	2383.00	2412.67	3626	2306.67	3518.00	2143.67	2123.00
VBAO-1	Acc.	0.9429	0.9286	0.9524	0.9524	0.9524	0.9048	0.9286	0.9286	0.9286
	#SF.	228.50	1292.00	265.33	1278.67	464.6667	501.33	163.00	1752.67	1497.67
VBAO-2	Acc.	0.9357	0.9524	0.9048	0.9286	0.9286	0.9286	0.9224	0.9286	0.9124
	#SF.	1151.90	745.67	1195.67	308.67	2570.667	741.33	347.33	2073.00	161.67
VBAO-3	Acc.	0.9286	0.9124	0.9286	0.9162	0.9286	0.9086	0.9048	0.9286	0.9286
	#SF.	1457.60	1975.33	1603.33	1471.00	1822.667	1854.33	2413.33	1490.00	1624.33
VBAO-4	Acc.	0.9357	0.9286	0.9286	0.9048	0.9286	0.9286	0.9286	0.9286	0.9286
	#SF.	2651.80	2045.33	2003.33	1396.67	1563.333	2866.67	2818.33	1940.33	701.00

References

- Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
- Liu, H.; Motoda, H. *Feature Selection for Knowledge Discovery and Data Mining*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 454.
- Kohavi, R.; John, G.H. Wrappers for feature subset selection. *Artif. Intell.* **1997**, *97*, 273–324. [\[CrossRef\]](#)
- Liu, H.; Motoda, H. *Feature Extraction, Construction and Selection: A Data Mining Perspective*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1998; Volume 453.
- Dhaenens, C.; Jourdan, L. *Metaheuristics for Big Data*; John Wiley & Sons: Hoboken, NJ, USA, 2016.
- Luukka, P. Feature selection using fuzzy entropy measures with similarity classifier. *Expert Syst. Appl.* **2011**, *38*, 4600–4607. [\[CrossRef\]](#)
- Dash, M.; Liu, H. Feature selection for classification. *Intell. Data Anal.* **1997**, *1*, 131–156. [\[CrossRef\]](#)
- Talbi, E.-G. *Metaheuristics: From Design to Implementation*; John Wiley & Sons: Hoboken, NJ, USA, 2009; Volume 74.
- El-Hasnony, I.M.; Barakat, S.I.; Elhoseny, M.; Mostafa, R.R. Improved feature selection model for big data analytics. *IEEE Access* **2020**, *8*, 66989–67004. [\[CrossRef\]](#)
- Yang, X.-S. *Nature-Inspired Metaheuristic Algorithms*; Luniver Press: Bristol, UK, 2010.
- Lopez-Garcia, P.; Masegosa, A.D.; Osaba, E.; Onieva, E.; Perallos, A. Ensemble classification for imbalanced data based on feature space partitioning and hybrid metaheuristics. *Appl. Intell.* **2019**, *49*, 2807–2822. [\[CrossRef\]](#)
- Shukla, A.K.; Singh, P.; Vardhan, M. Gene selection for cancer types classification using novel hybrid metaheuristics approach. *Swarm Evol. Comput.* **2020**, *54*, 100661. [\[CrossRef\]](#)
- Oliva, D.; Cuevas, E.; Pajares, G. Parameter identification of solar cells using artificial bee colony optimization. *Energy* **2014**, *72*, 93–102. [\[CrossRef\]](#)
- Nadimi-Shahraki, M.H.; Taghian, S.; Mirjalili, S.; Abualigah, L.; Abd Elaziz, M.; Oliva, D. EWOA-OPF: Effective Whale Optimization Algorithm to Solve Optimal Power Flow Problem. *Electronics* **2021**, *10*, 2975. [\[CrossRef\]](#)
- Zhang, Z.; Hong, W.-C. Application of variational mode decomposition and chaotic grey wolf optimizer with support vector regression for forecasting electric loads. *Knowl. Based Syst.* **2021**, *228*, 107297. [\[CrossRef\]](#)

16. Ali, M.H.; Kamel, S.; Hassan, M.H.; Tostado-Véliz, M.; Zawbaa, H.M. An improved wild horse optimization algorithm for reliability based optimal DG planning of radial distribution networks. *Energy Rep.* **2022**, *8*, 582–604. [\[CrossRef\]](#)
17. Sharma, S.; Saha, A.K.; Lohar, G. Optimization of weight and cost of cantilever retaining wall by a hybrid metaheuristic algorithm. *Eng. Comput.* **2021**, 1–27. [\[CrossRef\]](#)
18. Mergos, P.E. Optimum design of 3D reinforced concrete building frames with the flower pollination algorithm. *J. Build. Eng.* **2021**, *44*, 102935. [\[CrossRef\]](#)
19. Etaati, B.; Dehkordi, A.A.; Sadollah, A.; El-Abd, M.; Neshat, M. A Comparative State-of-the-Art Constrained Metaheuristics Framework for TRUSS Optimisation on Shape and Sizing. *Math. Probl. Eng.* **2022**, 6078986. [\[CrossRef\]](#)
20. Nadimi-Shahraki, M.H.; Moeini, E.; Taghian, S.; Mirjalili, S. DMFO-CD: A Discrete Moth-Flame Optimization Algorithm for Community Detection. *Algorithms* **2021**, *14*, 314. [\[CrossRef\]](#)
21. Xie, H.; Zhang, L.; Lim, C.P.; Yu, Y.; Liu, C.; Liu, H.; Walters, J. Improving K-means clustering with enhanced firefly algorithms. *Appl. Soft Comput.* **2019**, *84*, 105763. [\[CrossRef\]](#)
22. Masdari, M.; Barshandeh, S. Discrete teaching–learning-based optimization algorithm for clustering in wireless sensor networks. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 5459–5476. [\[CrossRef\]](#)
23. Rahnema, N.; Gharehchopogh, F.S. An improved artificial bee colony algorithm based on whale optimization algorithm for data clustering. *Multimed. Tools Appl.* **2020**, *79*, 32169–32194. [\[CrossRef\]](#)
24. Trinh, C.; Huynh, B.; Bidaki, M.; Rahmani, A.M.; Hosseinzadeh, M.; Masdari, M. Optimized fuzzy clustering using moth-flame optimization algorithm in wireless sensor networks. *Artif. Intell. Rev.* **2022**, *55*, 1915–1945. [\[CrossRef\]](#)
25. Oliva, D.; Hinojosa, S.; Cuevas, E.; Pajares, G.; Avalos, O.; Gálvez, J. Cross entropy based thresholding for magnetic resonance brain images using Crow Search Algorithm. *Expert Syst. Appl.* **2017**, *79*, 164–180. [\[CrossRef\]](#)
26. Chakraborty, S.; Saha, A.K.; Nama, S.; Debnath, S. COVID-19 X-ray image segmentation by modified whale optimization algorithm with population reduction. *Comput. Biol. Med.* **2021**, *139*, 104984. [\[CrossRef\]](#)
27. Houssein, E.H.; Helmy, B.E.-D.; Oliva, D.; Jangir, P.; Premkumar, M.; Elngar, A.A.; Shaban, H. An efficient multi-thresholding based COVID-19 CT images segmentation approach using an improved equilibrium optimizer. *Biomed. Signal Process. Control* **2022**, *73*, 103401. [\[CrossRef\]](#)
28. Mohakud, R.; Dash, R. Skin cancer image segmentation utilizing a novel EN-GWO based hyper-parameter optimized FCEDN. *J. King Saud Univ. Comput. Inf. Sci.* **2022**. [\[CrossRef\]](#)
29. Chakraborty, S.; Sharma, S.; Saha, A.K.; Chakraborty, S. SHADE-WOA: A metaheuristic algorithm for global optimization. *Appl. Soft Comput.* **2021**, *113*, 107866. [\[CrossRef\]](#)
30. Nadimi-Shahraki, M.H.; Taghian, S.; Mirjalili, S.; Ewees, A.A.; Abualigah, L.; Abd Elaziz, M. MTV-MFO: Multi-trial vector-based moth-flame optimization Algorithm. *Symmetry* **2021**, *13*, 2388. [\[CrossRef\]](#)
31. Singh, H.; Singh, B.; Kaur, M. An improved elephant herding optimization for global optimization problems. *Eng. Comput.* **2021**, 1–33. [\[CrossRef\]](#)
32. Gharehchopogh, F.S. An Improved Tunicate Swarm Algorithm with Best-random Mutation Strategy for Global Optimization Problems. *J. Bionic Eng.* **2022**, 1–26. [\[CrossRef\]](#)
33. Mergos, P.E.; Yang, X.-S. Flower pollination algorithm with pollinator attraction. *Evol. Intell.* **2022**, *15*, 1–17. [\[CrossRef\]](#)
34. Nadimi-Shahraki, M.H.; Fatahi, A.; Zamani, H.; Mirjalili, S.; Abualigah, L. An Improved Moth-Flame Optimization Algorithm with Adaptation Mechanism to Solve Numerical and Mechanical Engineering Problems. *Entropy* **2021**, *23*, 1637. [\[CrossRef\]](#)
35. Yang, Q.; Hua, L.; Gao, X.; Xu, D.; Lu, Z.; Jeon, S.-W.; Zhang, J. Stochastic Cognitive Dominance Leading Particle Swarm Optimization for Multimodal Problems. *Mathematics* **2022**, *10*, 761. [\[CrossRef\]](#)
36. Sayarshad, H.R. Using bees algorithm for material handling equipment planning in manufacturing systems. *Int. J. Adv. Manuf. Technol.* **2010**, *48*, 1009–1018. [\[CrossRef\]](#)
37. Zhou, Y.; Yang, X.; Tao, L.; Yang, L. Transformer Fault Diagnosis Model Based on Improved Gray Wolf Optimizer and Probabilistic Neural Network. *Energies* **2021**, *14*, 3029. [\[CrossRef\]](#)
38. Varae, H.; Shishegaran, A.; Ghasemi, M.R. The life-cycle cost analysis based on probabilistic optimization using a novel algorithm. *J. Build. Eng.* **2021**, *43*, 103032. [\[CrossRef\]](#)
39. Rodríguez, A.; Camarena, O.; Cuevas, E.; Aranguren, I.; Valdivia-G, A.; Morales-Castañeda, B.; Zaldívar, D.; Pérez-Cisneros, M. Group-based synchronous-asynchronous Grey Wolf Optimizer. *Appl. Math. Model.* **2021**, *93*, 226–243. [\[CrossRef\]](#)
40. Asghari, K.; Masdari, M.; Gharehchopogh, F.S.; Saneifard, R. Multi-swarm and chaotic whale-particle swarm optimization algorithm with a selection method based on roulette wheel. *Expert Syst.* **2021**, *38*, e12779. [\[CrossRef\]](#)
41. Ghasemi, M.R.; Varae, H. Enhanced IGMM optimization algorithm based on vibration for numerical and engineering problems. *Eng. Comput.* **2018**, *34*, 91–116. [\[CrossRef\]](#)
42. Oussalah, M.; Hessami, A.; Navimipour, N.J.; Rahmani, A.M.; Navin, A.H.; Hosseinzadeh, M. Job scheduling in the Expert Cloud based on genetic algorithms. *Kybernetes* **2014**, *43*, 1262–1275.
43. Alboaneen, D.; Tianfield, H.; Zhang, Y.; Pranggono, B. A metaheuristic method for joint task scheduling and virtual machine placement in cloud data centers. *Future Gener. Comput. Syst.* **2021**, *115*, 201–212. [\[CrossRef\]](#)
44. Attiya, I.; Abualigah, L.; Elsadek, D.; Chelloug, S.A.; Abd Elaziz, M. An Intelligent Chimp Optimizer for Scheduling of IoT Application Tasks in Fog Computing. *Mathematics* **2022**, *10*, 1100. [\[CrossRef\]](#)

45. Dashti, S.E.; Rahmani, A.M. Dynamic VMs placement for energy efficiency by PSO in cloud computing. *J. Exp. Theor. Artif. Intell.* **2016**, *28*, 97–112. [\[CrossRef\]](#)
46. Satpathy, A.; Addya, S.K.; Turuk, A.K.; Majhi, B.; Sahoo, G. Crow search based virtual machine placement strategy in cloud data centers with live migration. *Comput. Electr. Eng.* **2018**, *69*, 334–350. [\[CrossRef\]](#)
47. Banaie-Dezfouli, M.; 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 (IDAP), Malatya, Turkey, 28–30 September 2018; pp. 1–6.
48. Cai, J.; Zhang, F.; Sun, S.; Li, T. A meta-heuristic assisted underwater glider path planning method. *Ocean. Eng.* **2021**, *242*, 110121. [\[CrossRef\]](#)
49. Jiang, Y.; Wu, Q.; Zhang, G.; Zhu, S.; Xing, W. A diversified group teaching optimization algorithm with segment-based fitness strategy for unmanned aerial vehicle route planning. *Expert Syst. Appl.* **2021**, *185*, 115690. [\[CrossRef\]](#)
50. Neshat, M.; Nezhad, M.M.; Abbasnejad, E.; Mirjalili, S.; Tjernberg, L.B.; Garcia, D.A.; Alexander, B.; Wagner, M. A deep learning-based evolutionary model for short-term wind speed forecasting: A case study of the Lillgrund offshore wind farm. *Energy Convers. Manag.* **2021**, *236*, 114002. [\[CrossRef\]](#)
51. Neshat, M.; Alexander, B.; Wagner, M. A hybrid cooperative co-evolution algorithm framework for optimising power take off and placements of wave energy converters. *Inf. Sci.* **2020**, *534*, 218–244. [\[CrossRef\]](#)
52. Nadimi-Shahraki, M.H.; Taghian, S.; Mirjalili, S. An improved grey wolf optimizer for solving engineering problems. *Expert Syst. Appl.* **2021**, *166*, 113917. [\[CrossRef\]](#)
53. Koza, J.R. Genetic programming. In *Search Methodologies*; Springer: Berlin/Heidelberg, Germany, 1997.
54. Rechenberg, I. Evolution Strategy: Optimization of Technical systems by means of biological evolution. *Holzboog Stuttg.* **1973**, *104*, 15–16.
55. Storn, R.; Price, K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [\[CrossRef\]](#)
56. Yao, X.; Liu, Y.; Lin, G. Evolutionary programming made faster. *IEEE Trans. Evol. Comput.* **1999**, *3*, 82–102. [\[CrossRef\]](#)
57. Holland, J.H. Genetic algorithms. *Sci. Am.* **1992**, *267*, 66–73. [\[CrossRef\]](#)
58. Simon, D. Biogeography-based optimization. *IEEE Trans. Evol. Comput.* **2008**, *12*, 702–713. [\[CrossRef\]](#)
59. Eberhart, R.; Kennedy, J. A new optimizer using particle swarm theory. In Proceedings of the MHS'95. Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995; pp. 39–43.
60. Dorigo, M.; Di Caro, G. Ant colony optimization: A new meta-heuristic. In Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 6–9 July 1999; pp. 1470–1477.
61. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471. [\[CrossRef\]](#)
62. Yang, X.-S. A new metaheuristic bat-inspired algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 65–74.
63. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [\[CrossRef\]](#)
64. Kaveh, A.; Dadas, A. A novel meta-heuristic optimization algorithm: Thermal exchange optimization. *Adv. Eng. Softw.* **2017**, *110*, 69–84. [\[CrossRef\]](#)
65. Erol, O.K.; Eksin, I. A new optimization method: Big bang–big crunch. *Adv. Eng. Softw.* **2006**, *37*, 106–111. [\[CrossRef\]](#)
66. Zhao, W.; Wang, L.; Zhang, Z. Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. *Knowl. Based Syst.* **2019**, *163*, 283–304. [\[CrossRef\]](#)
67. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [\[CrossRef\]](#)
68. 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. [\[CrossRef\]](#)
69. Li, C.; Li, J.; Chen, H.; Jin, M.; Ren, H. Enhanced Harris hawks optimization with multi-strategy for global optimization tasks. *Expert Syst. Appl.* **2021**, *185*, 115499. [\[CrossRef\]](#)
70. Nadimi-Shahraki, M.H.; Taghian, S.; Mirjalili, S.; Zamani, H.; Bahreininejad, A. GGWO: Gaze cues learning-based grey wolf optimizer and its applications for solving engineering problems. *J. Comput. Sci.* **2022**, *61*, 101636. [\[CrossRef\]](#)
71. Abualigah, L.; Yousri, D.; Abd Elaziz, M.; Ewees, A.A.; Al-qaness, M.A.; Gandomi, A.H. Aquila Optimizer: A novel meta-heuristic optimization Algorithm. *Comput. Ind. Eng.* **2021**, *157*, 107250. [\[CrossRef\]](#)
72. Faris, H.; Habib, M.; Almomani, I.; Eshtay, M.; Aljarah, I. Optimizing extreme learning machines using chains of salps for efficient Android ransomware detection. *Appl. Sci.* **2020**, *10*, 3706. [\[CrossRef\]](#)
73. Alazzam, H.; Shariieh, A.; Sabri, K.E. A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer. *Expert Syst. Appl.* **2020**, *148*, 113249. [\[CrossRef\]](#)
74. Zhou, Y.; Cheng, G.; Jiang, S.; Dai, M. Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Comput. Netw.* **2020**, *174*, 107247. [\[CrossRef\]](#)
75. Turabieh, H.; Mafarja, M.; Li, X. Iterated feature selection algorithms with layered recurrent neural network for software fault prediction. *Expert Syst. Appl.* **2019**, *122*, 27–42. [\[CrossRef\]](#)
76. Catal, C.; Diri, B. Investigating the effect of dataset size, metrics sets, and feature selection techniques on software fault prediction problem. *Inf. Sci.* **2009**, *179*, 1040–1058. [\[CrossRef\]](#)

77. Ververidis, D.; Kotropoulos, C. Fast and accurate sequential floating forward feature selection with the Bayes classifier applied to speech emotion recognition. *Signal Process.* **2008**, *88*, 2956–2970. [\[CrossRef\]](#)
78. Liu, Z.-T.; Wu, M.; Cao, W.-H.; Mao, J.-W.; Xu, J.-P.; Tan, G.-Z. Speech emotion recognition based on feature selection and extreme learning machine decision tree. *Neurocomputing* **2018**, *273*, 271–280. [\[CrossRef\]](#)
79. Wang, G.; Ma, J.; Yang, S. An improved boosting based on feature selection for corporate bankruptcy prediction. *Expert Syst. Appl.* **2014**, *41*, 2353–2361. [\[CrossRef\]](#)
80. Ravi, V.; Pramodh, C. Threshold accepting trained principal component neural network and feature subset selection: Application to bankruptcy prediction in banks. *Appl. Soft Comput.* **2008**, *8*, 1539–1548. [\[CrossRef\]](#)
81. Jadhav, S.; He, H.; Jenkins, K. Information gain directed genetic algorithm wrapper feature selection for credit rating. *Appl. Soft Comput.* **2018**, *69*, 541–553. [\[CrossRef\]](#)
82. Lee, M.-C. Using support vector machine with a hybrid feature selection method to the stock trend prediction. *Expert Syst. Appl.* **2009**, *36*, 10896–10904. [\[CrossRef\]](#)
83. Hosseinalipour, A.; Gharehchopogh, F.S.; Masdari, M.; Khademi, A. A novel binary farmland fertility algorithm for feature selection in analysis of the text psychology. *Appl. Intell.* **2021**, *51*, 4824–4859. [\[CrossRef\]](#)
84. Zhang, Y.; Wang, S.; Phillips, P.; Ji, G. Binary PSO with mutation operator for feature selection using decision tree applied to spam detection. *Knowl. Based Syst.* **2014**, *64*, 22–31. [\[CrossRef\]](#)
85. Mohammadzadeh, H.; Gharehchopogh, F.S. A novel hybrid whale optimization algorithm with flower pollination algorithm for feature selection: Case study Email spam detection. *Comput. Intell.* **2021**, *37*, 176–209. [\[CrossRef\]](#)
86. Behrens, T.; Zhu, A.-X.; Schmidt, K.; Scholten, T. Multi-scale digital terrain analysis and feature selection for digital soil mapping. *Geoderma* **2010**, *155*, 175–185. [\[CrossRef\]](#)
87. Akay, M.F. Support vector machines combined with feature selection for breast cancer diagnosis. *Expert Syst. Appl.* **2009**, *36*, 3240–3247. [\[CrossRef\]](#)
88. Shaban, W.M.; Rabie, A.H.; Saleh, A.I.; Abo-Elsoud, M. A new COVID-19 Patients Detection Strategy (CPDS) based on hybrid feature selection and enhanced KNN classifier. *Knowl. Based Syst.* **2020**, *205*, 106270. [\[CrossRef\]](#)
89. Chatterjee, S.; Biswas, S.; Majee, A.; Sen, S.; Oliva, D.; Sarkar, R. Breast cancer detection from thermal images using a Grunwald-Letnikov-aided Dragonfly algorithm-based deep feature selection method. *Comput. Biol. Med.* **2022**, *141*, 105027. [\[CrossRef\]](#) [\[PubMed\]](#)
90. Ewees, A.A.; Al-qaness, M.A.; Abualigah, L.; Oliva, D.; Algamal, Z.Y.; Anter, A.M.; Ali Ibrahim, R.; Ghoniem, R.M.; Abd Elaziz, M. Boosting Arithmetic Optimization Algorithm with Genetic Algorithm Operators for Feature Selection: Case Study on Cox Proportional Hazards Model. *Mathematics* **2021**, *9*, 2321. [\[CrossRef\]](#)
91. Yan, K.; Zhang, D. Feature selection and analysis on correlated gas sensor data with recursive feature elimination. *Sens. Actuators B Chem.* **2015**, *212*, 353–363. [\[CrossRef\]](#)
92. Huang, Y.; Li, F.; Bao, G.; Xiao, Q.; Wang, H. Modeling the effects of biodiesel chemical composition on iodine value using novel machine learning algorithm. *Fuel* **2022**, *316*, 123348. [\[CrossRef\]](#)
93. Jain, I.; Jain, V.K.; Jain, R. Correlation feature selection based improved-binary particle swarm optimization for gene selection and cancer classification. *Appl. Soft Comput.* **2018**, *62*, 203–215. [\[CrossRef\]](#)
94. Lu, H.; Chen, J.; Yan, K.; Jin, Q.; Xue, Y.; Gao, Z. A hybrid feature selection algorithm for gene expression data classification. *Neurocomputing* **2017**, *256*, 56–62. [\[CrossRef\]](#)
95. Feng, C.; Cui, M.; Hodge, B.-M.; Zhang, J. A data-driven multi-model methodology with deep feature selection for short-term wind forecasting. *Appl. Energy* **2017**, *190*, 1245–1257. [\[CrossRef\]](#)
96. Li, S.; Wang, P.; Goel, L. Wind power forecasting using neural network ensembles with feature selection. *IEEE Trans. Sustain. Energy* **2015**, *6*, 1447–1456. [\[CrossRef\]](#)
97. Taghian, S.; Nadimi-Shahraki, M.H.; Zamani, H. Comparative Analysis of Transfer Function-based Binary Metaheuristic Algorithms for Feature Selection. In Proceedings of the 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turkey, 28–30 September 2018; pp. 1–6.
98. Kennedy, J.; Eberhart, R.C. A discrete binary version of the particle swarm algorithm. In Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Orlando, FL, USA, 12–15 October 1997; pp. 4104–4108.
99. Mirjalili, S.; Lewis, A. S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm Evol. Comput.* **2013**, *9*, 1–14. [\[CrossRef\]](#)
100. Mirjalili, S.; Zhang, H.; Mirjalili, S.; Chalup, S.; Noman, N. A Novel U-Shaped Transfer Function for Binary Particle Swarm Optimisation. In Proceedings of the 9th International Conference on Soft Computing for Problem Solving, SocProS, Liverpool, UK, 2–4 September 2019; pp. 241–259.
101. He, Y.; Zhang, F.; Mirjalili, S.; Zhang, T. Novel binary differential evolution algorithm based on Taper-shaped transfer functions for binary optimization problems. *Swarm Evol. Comput.* **2021**, *69*, 101022. [\[CrossRef\]](#)
102. Engelbrecht, A.P.; Pampara, G. Binary differential evolution strategies. In Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2007, Piscataway, NJ, USA, 25–28 September 2007; pp. 1942–1947.
103. Kundu, R.; Chattopadhyay, S.; Cuevas, E.; Sarkar, R. AltWOA: Altruistic Whale Optimization Algorithm for feature selection on microarray datasets. *Comput. Biol. Med.* **2022**, *144*, 105349. [\[CrossRef\]](#)

104. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. BGSa: Binary gravitational search algorithm. *Nat. Comput.* **2010**, *9*, 727–745. [[CrossRef](#)]
105. Emary, E.; Zawbaa, H.M.; Hassanien, A.E. Binary grey wolf optimization approaches for feature selection. *Neurocomputing* **2016**, *172*, 371–381. [[CrossRef](#)]
106. Arora, S.; Anand, P. Binary butterfly optimization approaches for feature selection. *Expert Syst. Appl.* **2019**, *116*, 147–160. [[CrossRef](#)]
107. Zhang, B.; Yang, X.; Hu, B.; Liu, Z.; Li, Z. OEbBOA: A novel improved binary butterfly optimization approaches with various strategies for feature selection. *IEEE Access* **2020**, *8*, 67799–67812. [[CrossRef](#)]
108. Taghian, S.; Nadimi-Shahraki, M.H. A Binary Metaheuristic Algorithm for Wrapper Feature Selection. *Int. J. Comput. Sci. Eng. (IJCSE)* **2019**, *8*, 168–172.
109. Li, A.-D.; Xue, B.; Zhang, M. Improved binary particle swarm optimization for feature selection with new initialization and search space reduction strategies. *Appl. Soft Comput.* **2021**, *106*, 107302. [[CrossRef](#)]
110. 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. [[CrossRef](#)]
111. Awadallah, M.A.; Hammouri, A.I.; Al-Betar, M.A.; Braik, M.S.; Abd Elaziz, M. Binary Horse herd optimization algorithm with crossover operators for feature selection. *Comput. Biol. Med.* **2022**, *141*, 105152. [[CrossRef](#)]
112. Albashish, D.; Hammouri, A.I.; Braik, M.; Atwan, J.; Sahran, S. Binary biogeography-based optimization based SVM-RFE for feature selection. *Appl. Soft Comput.* **2021**, *101*, 107026. [[CrossRef](#)]
113. Hussan, M.R.; Sarwar, M.I.; Sarwar, A.; Tariq, M.; Ahmad, S.; Shah Noor Mohamed, A.; Khan, I.A.; Ali Khan, M.M. Aquila Optimization Based Harmonic Elimination in a Modified H-Bridge Inverter. *Sustainability* **2022**, *14*, 929. [[CrossRef](#)]
114. Wang, S.; Jia, H.; Abualigah, L.; Liu, Q.; Zheng, R. An improved hybrid aquila optimizer and harris hawks algorithm for solving industrial engineering optimization problems. *Processes* **2021**, *9*, 1551. [[CrossRef](#)]
115. Nakamura, R.Y.; Pereira, L.A.; Costa, K.; Rodrigues, D.; Papa, J.P.; Yang, X.-S. BBA: A binary bat algorithm for feature selection. In Proceedings of the 25th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Ouro Preto, Brazil, 22–25 August 2012; pp. 291–297.
116. 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; pp. 12–17.
117. Taghian, S.; Nadimi-Shahraki, M.H. Binary Sine Cosine Algorithms for Feature Selection from Medical Data. *arXiv* **2019**, arXiv:1911.07805. [[CrossRef](#)]
118. 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. [[CrossRef](#)]
119. Blake, C.L.; Merz, C.J. *UCI Repository of Machine Learning Databases*; University of California: Oakland, CA, USA, 1998.
120. Zhu, Z.; Ong, Y.-S.; Dash, M. Markov blanket-embedded genetic algorithm for gene selection. *Pattern Recognit.* **2007**, *40*, 3236–3248. [[CrossRef](#)]
121. Iwendi, C.; Bashir, A.K.; Peshkar, A.; Sujatha, R.; Chatterjee, J.M.; Pasupuleti, S.; Mishra, R.; Pillai, S.; Jo, O. COVID-19 patient health prediction using boosted random forest algorithm. *Front. Public Health* **2020**, *8*, 357. [[CrossRef](#)] [[PubMed](#)]