




Article

# Quantum Chaotic Honey Badger Algorithm for Feature Selection

Samah Alshathri <sup>1,\*</sup>, Mohamed Abd Elaziz <sup>2,3,4,5</sup>, Dalia Yousri <sup>6</sup>, Osama Farouk Hassan <sup>7</sup>  
and Rehab Ali Ibrahim <sup>4</sup>

- <sup>1</sup> Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia  
<sup>2</sup> Faculty of Computer Science and Engineering, Galala University, Suez 435611, Egypt  
<sup>3</sup> Artificial Intelligence Research Center (AIRC), Ajman University, Ajman P.O. Box 346, United Arab Emirates  
<sup>4</sup> Department of Mathematics, Faculty of Science, Zagazig University, Zagazig 44519, Egypt  
<sup>5</sup> Department of Electrical and Computer Engineering, Lebanese American University, Byblos 13-5053, Lebanon  
<sup>6</sup> Electrical Engineering Department, Faculty of Engineering, Fayoum University, Faiyum 63514, Egypt  
<sup>7</sup> Department of Information System, Faculty of Computers and Informatics, Suez Canal University, Ismailia 41522, Egypt  
\* Correspondence: sealshathry@pnu.edu.sa

**Abstract:** Determining the most relevant features is a critical pre-processing step in various fields to enhance prediction. To address this issue, a set of feature selection (FS) techniques have been proposed; however, they still have certain limitations. For example, they may focus on nearby points, which lowers classification accuracy because the chosen features may include noisy features. To take advantage of the benefits of the quantum-based optimization technique and the 2D chaotic Hénon map, we provide a modified version of the honey badger algorithm (HBA) called QCHBA. The ability of such strategies to strike a balance between exploitation and exploration while identifying the workable subset of pertinent features is the basis for employing them to enhance HBA. The effectiveness of QCHBA was evaluated in a series of experiments conducted using eighteen datasets involving comparison with recognized FS techniques. The results indicate high efficiency of the QCHBA among the datasets using various performance criteria.

**Keywords:** feature selection; honey badger algorithm (HBA); quantum-based optimization technique; 2D chaotic Hénon map



**Citation:** Alshathri, S.; Abd Elaziz, M.; Yousri, D.; Hassan, O.F.; Ibrahim, R.A. Quantum Chaotic Honey Badger Algorithm for Feature Selection. *Electronics* **2022**, *11*, 3463. <https://doi.org/10.3390/electronics11213463>

Academic Editors: D. J. Lee and Dong Zhang

Received: 15 September 2022

Accepted: 20 October 2022

Published: 26 October 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

The enormous increase in the volume of data has resulted in a range of difficulties and issues, including noisy, high-dimensional, and irrelevant data [1]. This results in significant processing costs and adversely affects the effectiveness and accuracy of machine learning systems. Approaches for feature selection (FS) have been used to lower computational costs and increase classification accuracy [2]. Feature selection is the process of limiting the number of collected features into a relevant subset that can be utilized to combat the problem of dimensionality. By choosing a sample of pertinent features, FS techniques are frequently employed to capture data qualities [3]. Additionally, they can eliminate irrelevant and distracting data [3]. FS techniques are frequently used in a variety of disciplines, including COVID-19 CT scan classification [4], text categorization, human activity detection [5,6], MR image segmentation [7,8], data analytics [9,10], parameter estimation of biochemical systems [10], investigation of neuromuscular diseases [11], and other applications [12–14].

There are three types of FS methods: wrapper, filter, and embedded [15]. Filter-based methods rely on dataset characteristics without depending on learning technique, while wrapper-based methods evaluate the chosen features using the learning process. Embedded algorithms identify the elements that most effectively improve model accuracy.

As a result, procedures based on filters are quicker and more effective than methods based on wrappers. A strategy that can limit the amount of features that must be chosen, while maximising classifier accuracy, is the optimal FS method [15].

Numerous MH algorithms have been successfully applied to address FS issues. These include, the particle swarm optimization (PSO) algorithm [16], the differential evolution (DE) algorithm [17], the genetic algorithm (GA) [18], the grasshopper optimization algorithm (GOA) [19], the gravitational search algorithm (GSA) [20], the slime mould algorithm (SMA) [3], the Harris hawk optimization (HHO) algorithm [21], the marine predator algorithm (MPA) [22] and others [7–10,23]. An FS strategy has been proposed that is based on the electric fish optimization (EFO) [24]. It was tested on a variety of challenging real-world datasets and performed very well, with the exception of high dimensional issues where it exhibited delayed convergence and a tendency to become stuck in local minima.

The NFL theorem states that no single algorithm can resolve every issue. As a result, a hybridization approach is frequently used to handle a variety of complicated issues, including FS. We utilize an optimizer known as the honey badger algorithm (HBA) [25] which takes its inspiration from the intelligent foraging behavior of honey badger. The authors of [26] developed the honey badger algorithm (HBA), which is an effective optimization technique for determining the ideal size and location of capacitors and other types of DG to reduce the overall active power. Since there are numerous variable quantities with various non-linear characteristics included in the proton exchange membrane fuel cell (PEMFC) design, which must be properly specified to ensure effective modelling, an alternative method to determine the PEMFC parameters was proposed in [27]. This method relies on use of the honey badger algorithm (HBA) as a novel identification approach to determine the PEMFC parameters. A new approach was suggested in [28] to address certain issues and drawbacks of the HBA which are manifest in trapping in local optima, low convergence, and imbalance between exploration and exploitation stages. The approach, known as mHBA is based on use of the dimensional learning hunting (DLH) method, an effective local search technique, which is incorporated into the HBA. In addition, HBA has proved its efficiency in quantum applications as reported in [29]. The authors proposed an IoT-based optimization scheme for task scheduling for minimizing energy in cloud computing using the HBA. The major objective of the present study was to identify the most significant cloud scheduling key. A committee of decision-makers first chose the standards for evaluation. The optimal solution was then determined after assigning weights to each criterion using an optimization technique. Last, but not least, the evaluation metrics included in the CloudSim toolbox, including makespan, energy consumption, and resource utilisation were applied. The results indicated that it might be challenging to match user demands with available cloud resources to improve performance and minimise energy use. This paper offers a work scheduling method that uses the honey badger algorithm (HBA) in a heterogeneous visualized cloud to save energy.

While application of the HBA has demonstrated a number of benefits, it still suffers from limitations that we were motivated to address. This has involved use of the quantum-based optimization (QBO) technique and the 2D chaotic Hénon map. These methods were used to enhance convergence towards a feasible solution, and to improve balancing of exploitation and exploration. To evaluate the performance of the proposed method, a set of eighteen datasets was used. The results were compared with other well-known FS methods. In general, the proposed technique, named QCHBA, starts by setting the value of a set of solutions using the QBO strategy then evaluates the performance of each using the error classification and the ratio of selected features as the fitness value. Then the best solution having the best fitness value is determined. The operators of the HBA and 2D Hénon maps are integrated to enhance the current set of solutions that represent the relevant features. This process is repeated until the terminal conditions occur. The testing sample is then used to assess the performance of the returned best solution using different performance measures. To the best of our knowledge, no integration of the QBO, the 2D Hénon map,

and the operators of MH techniques has been proposed to date. This motivated us to apply this integration as an FS method.

The main contributions of this study can be summarized as follows:

- A modified version of the honey badger algorithm is proposed and applied as a feature selection technique.
- Quantum-based optimization and the 2D Hénon chaotic map are used to improve the HBA during the process of selecting relevant features.
- The efficiency of the developed method is evaluated using state-of-the-art FS methods applied to eighteen datasets.

## 2. Background

### 2.1. Honey Badger Algorithm (HBA)

The characteristics of the honey badger algorithm (HBA) are described in detail in this section [25]. The design of the HBA was influenced by the foraging behaviour of honey badgers. The honey badger uses its sense of smell as one method of locating its meal, and also uses digging as a second method. The honey badger relies on honey-guide birds to locate and then enter the hives. The “digging phase” was the name used for the first tactic, and the “honey phase” was the name given by the algorithm’s designers to the second tactic. The honey badger’s olfactory sensitivity controls how it moves; when the fragrance is potent, the insect will move quickly, and vice versa [25].

The HBA’s main stages and corresponding equations are as follows:

- Initialization process: During this stage, the problem space’s upper ( $HU$ ) and lower ( $HL$ ) boundaries are used to determine the first potential solution. As a result, the first solutions are stochastic sets that may be created using the following method according to Equation (1) [25].

$$H_i = HL + r_1(1, D) \times (HU - HL), \quad i = 1, 2, \dots, N. \tag{1}$$

where  $D$  refers to the dimension of the solution,  $H$  represents the total of the potential solutions, and  $N$  is the number of solution providers (honey badgers).

- Updating positions: Coordinates of the candidates  $H_{new}$  are updated at this point. For instance, this can involve applying a technique that uses either the digging or honey stages.
  - Digging phase: In this phase, the movements of the potential search subjects are influenced by the potency of the predator’s odor and the separation of the honey badger (agent) from the prey ( $P$ ). The honey badger excavates in a polarized circle. The following is the stated formula for its motion:

$$H_{new} = P + Fg \times \beta \times In \times P + Fg \times r_3 \times (P - H_i) \times (\cos 2\pi r_4) \times (1 - \cos 2\pi r_5) \tag{2}$$

where  $\beta$  measures an insect’s ability to collect food. Hashim et al. [25] reported that the maximum value of  $\beta$  that might exist is 6. The intensity is  $In$ , and the  $r_3, r_4$ , and  $r_5$  are random variables with a range of 0 to 1 that were selected from a uniform distribution. The  $Fg$ , which serves as a search direction indication, is produced as follows:

$$Fg = \begin{cases} 1 & \text{if } r_6 \leq 0.5 \\ -1 & \text{if else} \end{cases} \tag{3}$$

- Honey phase: When looking for beehives, honey badgers utilize the honey phase to alter their position relative to the honey lead bird. The following equation was used by Hashim et al. [25] to determine the honey phase:

$$H_{new} = P + Fg \times r_7 \times \sigma \times (P - H_i) \tag{4}$$

where  $P$  is the best solution obtained so far and  $r_7$  is a random number having values between 0 and 1.

- Modeling intensity  $In$ : Since the honey badger perception of insect scent governs its behavior, Hashim et al. [25] created the following formulation for each contender's scent intensity  $Ini$  of the prey.

$$In_i = r_2 \times \frac{(H_i - H_{i+1})^2}{4\pi(P - H_i)} \quad (5)$$

where  $P$  is the prey's location and  $r_2$  is a stochastic number in the range [0, 1].

- Modeling the density parameter ( $\sigma$ ): According to Hashim et al. [25], the  $\sigma$  value serves as a regulator for transmission among the local and global search stages. Hashim et al. [25] hypothesized that  $\beta$  is depicted throughout the iterations, as shown below:

$$\sigma = C \times \exp\left(\frac{-IT}{IT_{max}}\right) \quad (6)$$

where  $IT$  and  $IT_{max}$  refer to the current and the total number of iterations, respectively.  $C$  is a constant that was recommended to be 2.

- Escaping from local solutions: the algorithm developers [25] employed a warning (Fg) to point out the search direction to avoid getting stuck on local solutions.

The primary structure of the HBA is described in the Algorithm 1 based on the preceding description.

---

#### Algorithm 1 Steps of HBA

---

- 1: **Inputs:** Agents size  $N$ , number of iterations  $Iter_{max}$ .
  - 2: **Outputs:** The optimal solutions.
  - 3: **Step 1:** Calculate the first set of  $N$  solutions  $U$  with dimension  $d$  (i.e., number of unknown variables).
  - 4: Compute the fitness function of Equation (13) and the corresponding swarm matrix as the best solutions ( $P$ ).
  - 5: **while** ( $Iter \leq Iter_{max}$ ) **do**
  - 6:   Upgrade the value of the decreasing factor through Equation (19).
  - 7:   **for** ( $i = 1$  to  $N$ ) **do**
  - 8:     Compute the intensity through Equation (5).
  - 9:     **if**  $r < 0.5$  **then**
  - 10:       Upgrade the location of  $H_{new}$  through Equation (15).
  - 11:     **else**
  - 12:       Upgrade the location of  $H_{new}$  through Equation (4).
  - 13:     Evaluate the new solutions and compute the  $Fit_{t+1}$  and assign  $Fit_{max_{t+1}}$ .
  - 14:     **if**  $Fit_{t+1} \leq Fit_t$  **then**
  - 15:       Set  $H_i = H_{new}$  and  $Fit_t = Fit_{t+1}$ .
  - 16:     **if**  $Fit_{max_{t+1}} \leq Fit_{max_t}$  **then**
  - 17:       Set  $H_{best} = H_{new}$  and  $Fit_{max_t} = Fit_{max_{t+1}}$ .
- 

#### 2.2. Two-Dimensional Hénon Map

The 2D Hénon map is one of the most popular discrete-time dynamical systems that replicates chaotic behavior [30]. The Hénon map, also known as the Hénon-Pomeau attractor/map, is a discrete-time dynamical system. It is one of the most well-researched examples of chaotic behavior in dynamical systems. The Hénon map converts a point  $(x_n, y_n)$  in the plane to a new point. The map was first presented by Michel Hénon as a condensed form of the Poincaré section of the Lorenz model. The Hénon strange attractor is a group of points, under the classical map, from which an initial plane point would either approach or diverge to infinity. A fractal known as the Hénon attractor has a Cantor set in

one direction and is smooth in the other. The mathematical formula of the Hénon map is written as in Equation (7) and its distribution is represented in Figure 1:

$$\begin{cases} x_{t+1} = 1 - 1.4 \cdot x_t^2 + y_t \\ y_{t+1} = 0.3 \cdot x_t \end{cases} \quad (7)$$

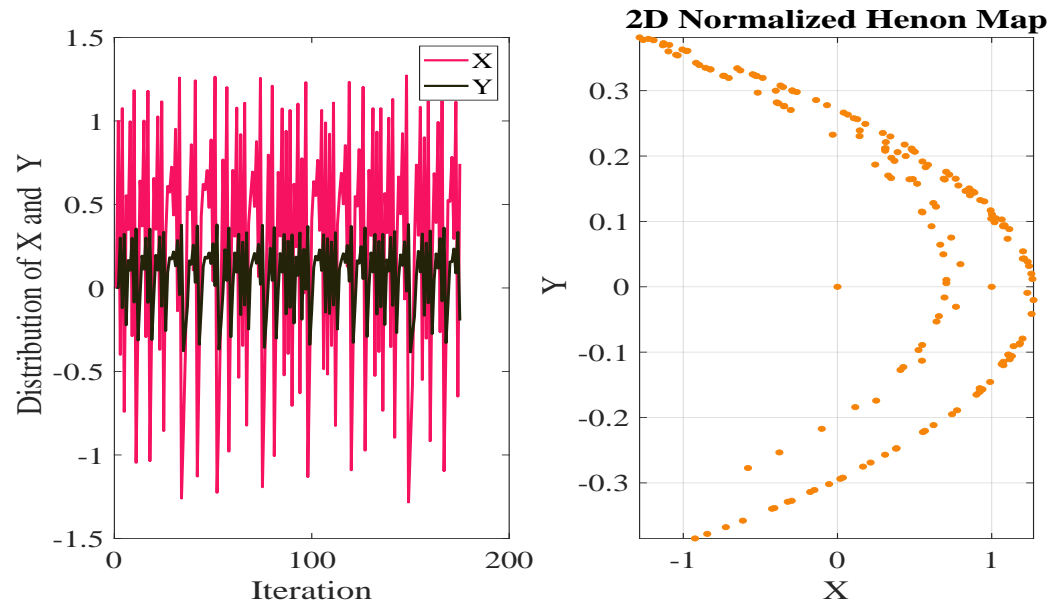


Figure 1. Hénon attractor, and  $x, y$  distributions.

### 3. Quantum Chaotic Honey Badger Algorithm (QCHBA)

In this approach, the HBA is combined with the two-dimensional Hénon map to improve the fundamental performance of the algorithm. Moreover, the quantum-based optimization technique is used to improve the ability to search to balance exploration and exploitation. In general, the proposed QCHAB approach starts by using the training set that represents 70% of the given dataset to determine the relevant features. Then the solution is generated using the quantum-based optimization technique. This is followed by computing the performance of each solution and determining the optimal solution. The next step is to update the solution according to the strengths of the 2D chaotic maps and the QBO. This updating procedure is repeated until terminal criteria are met and then the best solution is the output of this stage. The next stage involves reducing the features in the testing set to represent 30% of the dataset. Then, the performance of the selected features is evaluated using different performance metrics.

#### 3.1. Initial Solutions

The main purpose of this stage is to generate the set of  $N$  solutions (i.e., population) using the QBO technique. These solutions consists of  $D$  Q-bits ( $D$  refers to the number of features) and this can be given as:

$$H_i = [q_{i1}|q_{i2}| \dots |q_{iD}] = [\theta_{i1}|\theta_{i2}| \dots |\theta_{iD}], i=1,2, \dots, N \quad (8)$$

In Equation (8),  $H_i$  denotes the superpositions of probabilities of the features that are selected that correspond to ones and features not selected that correspond to zeros.

### 3.2. Updating Solution

In this stage, the QCHAB starts by obtaining the binary form of solution  $H_i, i = 1, 2, \dots, N$  using the following formula.

$$BH_{i,j} = \begin{cases} 1 & \text{if rand} < |\beta|^2 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

In Equation (9),  $\text{rand} \in [0, 1]$  refers to a random value.

Thereafter, the fitness value for each  $H_i$  is computed using the following equation.

$$Fit_i = \rho \times \gamma + (1 - \rho) \times \left( \frac{|BH_{i,j}|}{D} \right) \quad (10)$$

In Equation (10),  $|BH_{i,j}|$  is the number of features that correspond to ones.  $\gamma$  refers to the error classification obtained from applying the KNN classifier, whereas  $\rho \in [0, 1]$  refers to the parameter that is used to balance the objectives (i.e., the error classification and feature selection).

The next step to compute the fitness value for each  $H_i$  then determines the best solution  $H_b$ . Then, the modified version of HBA based on the 2D Hénon map is used to update the current solutions. The following is a summary of the proposed modification and controlled equations, in which there are two modifications.

- First modification: The two-dimensional Hénon map is applied to adjust the parameters of  $C$  and  $\beta$  of Equation (11), respectively, to improve the functionality of the fundamental HBA optimizer. The updated values of  $C$  and  $\beta$  follow the equation shown below:

$$\begin{aligned} C(t) &= 4 * y_{t+1} \\ \beta(t) &= 7 * H_{t+1} \end{aligned} \quad (11)$$

where  $H_{t+1}$  and  $y_{t+1}$  are vectors of the Hénon map and  $t$  is the current iteration. The Hénon map vectors are normalized using the numbers 4 and 7 to place them inside the HBA developer's suggested range. The developers chose to use  $\beta$  and  $C$  values of 6 and 2, as mentioned in the previous section on the fundamental HBA.

The variables  $\beta$  and  $C$  undergo changes during the course of the iterations with values ranging from 0 to 7 and 0 to 4, respectively.

The CHBA uses the values 4 and 7 to provide broad diversity. When implemented, the Hénon map's initialization is 0 ( $x(1) = 0; y(1) = 0$ ). Figure 1 shows how the map's attractor works.

Then the digging phase and the density variable are modeled as

$$H_{new} = P + Fg \times \beta(t) \times In \times P + Fg \times r_3 \times (P - H_i) \times (\cos 2\pi r_4) \times (1 - \cos 2\pi r_5) \quad (12)$$

$$\sigma = C(t) \times \exp\left(\frac{-Iter}{Iter_{max}}\right) \quad (13)$$

The process of updating the solutions is conducted until the stop conditions are met. Then the best solution  $H_b$  is used as the output of this stage.

### 3.3. Evaluate Quality of $H_b$

Within this phase, the best solution  $H_b$  is used to remove irrelevant features from the testing set. Then, this reduced testing set is used as input to the learned KNN classifier to predict the target of the testing set. The performance of the predicted target is then computed using a set of performance criteria. The QCHBA algorithm steps are listed in Algorithm 2.



**Algorithm 2** Pseudo code of QCHBA

---

```

1: Inputs: Agents size  $N$ , number of iterations  $Iter_{max}$ , the dataset.
2: Outputs: The optimal solutions.
3: Step 1: Calculate the first set of  $N$  solutions  $U$  with dimension  $d$  (i.e., number of
  unknown variables using QBO as in Equation (9)).
4: Compute the fitness function as in Equation (10) and the corresponding swarm matrix
  as the best solutions ( $P$ ).
5: Calculate  $C$ , and  $\beta$  based on Hénon map, using Equation (11) with dimensions of
   $1 * Iter_{max}$ .
6: while ( $Iter \leq Iter_{max}$ ) do
7:   Upgrade the value of the decreasing factor through Equation (13).
8:   for ( $i = 1$  to  $N$ ) do
9:     Compute the intensity through Equation (5).
10:    if  $r < 0.5$  then
11:      Upgrade the location of  $H_{new}$  through Equation (12).
12:    else
13:      Upgrade the location of  $H_{new}$  through Equation (4).
14:    Evaluate the new solutions and compute the  $Fit_{t+1}$  and assign  $Fit_{max_{t+1}}$ .
15:    if  $Fit_{t+1} \leq Fit_t$  then
16:      Set  $H_i = H_{new}$  and  $Fit_t = Fit_{t+1}$ .
17:    if  $Fit_{max_{t+1}} \leq Fit_{max_t}$  then
18:      Set  $H_{best} = H_{new}$  and  $Fit_{max_t} = Fit_{max_{t+1}}$ .
19: Evaluate the performance of the best solution using the testing set.

```

---

**4. Experimental Results***4.1. Data Description*

The evaluation of the performance of QCHBA was performed using eighteen UCI datasets [31]. The characteristics of these datasets are detailed in Table 1. From these characteristics, it can be observed that there are different classes, features, and samples, and that they derive from different domains.

**Table 1.** Dataset descriptions.

| Datasets          | Number of Features | Number of Instances | Number of Classes | Data Category   |
|-------------------|--------------------|---------------------|-------------------|-----------------|
| Breastcancer (S1) | 9                  | 699                 | 2                 | Biology         |
| BreastEW (S2)     | 30                 | 569                 | 2                 | Biology         |
| CongressEW (S3)   | 16                 | 435                 | 2                 | Politics        |
| Exactly (S4)      | 13                 | 1000                | 2                 | Biology         |
| Exactly2 (S5)     | 13                 | 1000                | 2                 | Biology         |
| HeartEW (S6)      | 13                 | 270                 | 2                 | Biology         |
| IonosphereEW (S7) | 34                 | 351                 | 2                 | Electromagnetic |
| KrvskpEW (S8)     | 36                 | 3196                | 2                 | Game            |
| Lymphography (S9) | 18                 | 148                 | 2                 | Biology         |
| M-of-n (S10)      | 13                 | 1000                | 2                 | Biology         |
| PenglungEW (S11)  | 325                | 73                  | 2                 | Biology         |
| SonarEW (S12)     | 60                 | 208                 | 2                 | Biology         |
| SpectEW (S13)     | 22                 | 267                 | 2                 | Biology         |
| Tic-tac-toc (S14) | 9                  | 958                 | 2                 | Game            |
| Vote (S15)        | 16                 | 300                 | 2                 | Politics        |
| WaveformEW (S16)  | 40                 | 5000                | 3                 | Physics         |
| Water (S17)       | 13                 | 178                 | 3                 | Chemistry       |
| Zoo (S18)         | 16                 | 101                 | 6                 | Artificial      |

#### 4.2. Performance Measures

Six performance metrics are presented in this section that were used to assess how well the proposed methods performed. These metrics included the averages of the accuracy, standard deviation (*Std*), fitness value, minimum fitness value (*Min*), and maximum fitness value (*Max*). All algorithms were used 25 times to obtain the performance metric averages.

- Accuracy (*Acc*): The corrected classified data ratio was calculated using this metric. It was determined using Equation (14).

$$Acc = \frac{TP + TN}{TP + FN + FP + TN} \quad (14)$$

where *FN*, *TN*, *FP* and *TP* denote false negative, true negative, false positive, and true positive, respectively.

- Fitness value: This metric assesses the effectiveness of the techniques using the fitness function as in Equation (10).
- Maximum of the fitness value: This metric captures the highest result that the fitness function for each method can achieve.

$$Max = \max_{1 \leq i \leq Nr} Fit_b^i \quad (15)$$

- Minimum of the fitness value: This metric captures the lowest result that the fitness function for each method can achieve.

$$Min = \min_{1 \leq i \leq Nr} Fit_b^i \quad (16)$$

- Selected features: This metric keeps track of how many chosen features each algorithm is able to produce.
- Standard deviation: This metric assesses an algorithm's consistency throughout numerous executions. The calculation is as in Equation (17).

$$Std = \sqrt{\frac{1}{Nr} \sum_{i=1}^{Nr} (Fit_i - Fit_a)^2} \quad (17)$$

The parameter *Nr* refers to the number of runs and *Fit<sub>i</sub>* refers to the given fitness value. Its average is given by *Fit<sub>a</sub>*. Whereas, *Fit<sub>b</sub><sup>i</sup>* refers to the best *Fit* at run *i*.

The results of the QCHBA were compared with the electric fish optimization (EFO) [24], the sinusoidal parameter adaptation incorporated with L-SHADE (LSEpSin) [32], the grey-wolf optimization algorithm (GWO) [33], the reptile search algorithm (RSA) [34], the L-SHADE with semi-parameter adaptation (LSHADECS) [35], and the self-adaptive differential evolution (SaDE) [36].

#### 4.3. Results and Discussion

The results obtained for the proposed QCHBA and the other FS methods are introduced in this section. The comparison was performed with the traditional HBA, the chaotic HAB, GWO, EFO, RSA, LSHADE, LSHADECS, and SaDE. In the initial investigations, the parameter settings of these FS algorithms were set as for the original implementation. The common parameters were set as  $N = 15$  and  $t_{max} = 50$ . To ensure a fair comparison, each method was run 25 times. The dataset was split into a training set representing 80% of the total and a testing set representing 20% of the total.

The results for the QCHBA and the other methods based on the average fitness values are given in Table 2. It can be seen from these results that the QCHBA had the smallest value at seven datasets, which represented nearly 39% of the the tested data. Both the CHBA and the HBA showed best performance of nearly 22% of the datasets. Figure 2 shows the average of each algorithm among the tested datasets based on the average fitness



value. From this figure, it can be seen that the QCHBA had the smallest average overall compared to the other algorithms. This was followed by the traditional HBA and CHBA which had the second and third best fitness value averages, respectively. LSHADECS was the worst algorithm based on the average fitness value obtained in this study.

The stability of the competitive methods is given in Table 3. From the values for the standard deviation (STD) of the fitness values for each method, it can be seen that the LSHADE was the most stable method overall. The STD results for the QCHBA indicated that its stability was better than the remaining methods with the second best performance. The EFO, SaDE, HBA, RSA, and LSHADECS methods occupied the subsequent ranks, respectively, based on their STD values. The same conclusions may be drawn from Figure 3.

Based on the results for the best fitness values given in Table 4 and Figure 4, the following observations can be made: First, the QCHBA method had the lowest results at nearly 61% from the eighteen datasets. This was followed by the HBA and CHBA methods which had best values of nearly 27% and 22%, respectively. The GWO and RSA methods achieved better performance in terms of best fitness values at 16% and 11%, respectively. This indicates that the operators of the HBA had the ability to reach smaller fitness values than the other algorithms. However, adding quantum and chaotic maps to these operators increased the probability of being optimal cases.

From the worst fitness value results for each method that are presented in Table 5 and Figure 5, the following observations can be made: The HBA and RSA showed better fitness values in the worst instances of each compared to the other methods. However, the proposed QCHBA approach maintained its superiority over the other methods with better fitness values for eight datasets, which represented nearly 45% of the total number of datasets tested. The convergence curves of the algorithms are shown in Figure 6. These curves indicate the high efficiency of the proposed method, which converged faster towards the optimal subset of features than the other methods.

**Table 2.** Results of QCHBA based on average of fitness values.

|     | HBA           | CHAB          | QCHBA         | lGWO          | EFO    | RSA           | LSHADE | LSHADECS | SaDE   |
|-----|---------------|---------------|---------------|---------------|--------|---------------|--------|----------|--------|
| S1  | 0.0568        | <b>0.0551</b> | 0.0736        | 0.0627        | 0.0823 | 0.0824        | 0.0833 | 0.1325   | 0.0917 |
| S2  | 0.0672        | 0.0552        | <b>0.0318</b> | 0.0796        | 0.0884 | 0.0752        | 0.1254 | 0.1996   | 0.1114 |
| S3  | 0.0443        | 0.0707        | 0.0849        | 0.0473        | 0.0989 | <b>0.0191</b> | 0.0655 | 0.1609   | 0.1063 |
| S4  | 0.0578        | 0.0710        | <b>0.0539</b> | 0.0989        | 0.0793 | 0.1103        | 0.2504 | 0.2942   | 0.3853 |
| S5  | 0.2214        | <b>0.1783</b> | 0.2496        | 0.2152        | 0.3134 | 0.2868        | 0.2258 | 0.3748   | 0.2285 |
| S6  | 0.1883        | <b>0.1557</b> | 0.1835        | 0.1920        | 0.1662 | 0.1733        | 0.2019 | 0.3583   | 0.2417 |
| S7  | 0.0794        | <b>0.0705</b> | 0.0923        | 0.0973        | 0.1457 | 0.0887        | 0.1160 | 0.1660   | 0.1148 |
| S8  | <b>0.0737</b> | 0.0842        | 0.0840        | 0.1024        | 0.0992 | 0.0953        | 0.3904 | 0.4010   | 0.3658 |
| S9  | 0.1944        | 0.1878        | 0.1150        | <b>0.1050</b> | 0.2242 | 0.1424        | 0.2567 | 0.2667   | 0.2516 |
| S10 | 0.0546        | 0.0689        | <b>0.0519</b> | 0.0766        | 0.0808 | 0.1181        | 0.2118 | 0.3503   | 0.2706 |
| S11 | 0.0341        | 0.0815        | <b>0.0080</b> | 0.0810        | 0.2006 | 0.1308        | 0.3200 | 0.3330   | 0.3500 |
| S12 | 0.0823        | 0.0764        | <b>0.0655</b> | 0.0840        | 0.1374 | 0.1213        | 0.2833 | 0.4167   | 0.3333 |
| S13 | <b>0.1188</b> | 0.2002        | 0.1518        | 0.1235        | 0.2345 | 0.2058        | 0.1630 | 0.2731   | 0.2417 |
| S14 | 0.2290        | 0.2409        | 0.2342        | 0.2678        | 0.2420 | <b>0.2276</b> | 0.2635 | 0.3234   | 0.2992 |
| S15 | 0.0714        | 0.0572        | 0.0504        | 0.1114        | 0.1043 | <b>0.0353</b> | 0.0567 | 0.1450   | 0.0850 |
| S16 | 0.2728        | 0.2810        | <b>0.2708</b> | 0.2914        | 0.3115 | 0.2969        | 0.3574 | 0.4506   | 0.4094 |
| S17 | <b>0.0565</b> | 0.0732        | 0.0647        | 0.0634        | 0.0796 | 0.0692        | 0.1833 | 0.1819   | 0.1583 |
| S18 | <b>0.0303</b> | 0.0400        | <b>0.0303</b> | 0.0481        | 0.0425 | 0.0338        | 0.3333 | 0.2133   | 0.0833 |

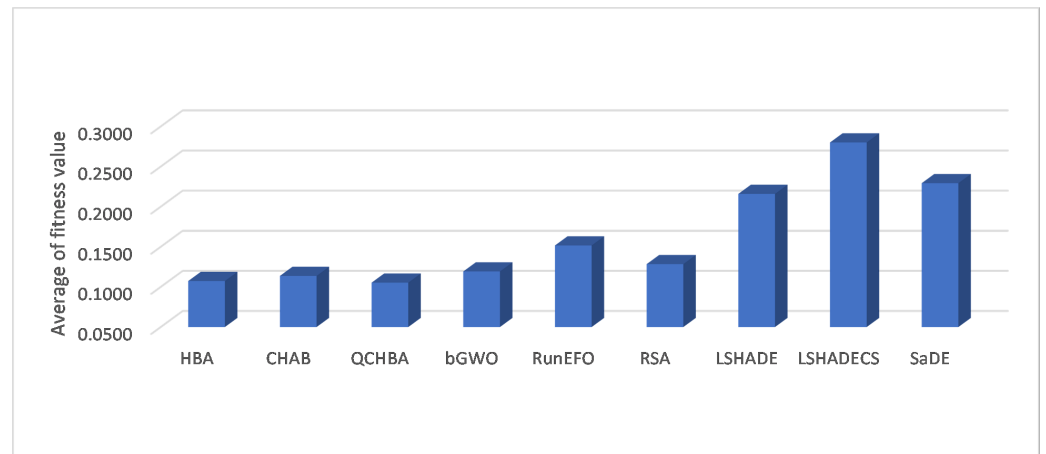


Figure 2. Competitive results of QCHBA and other methods in terms of average fitness values.

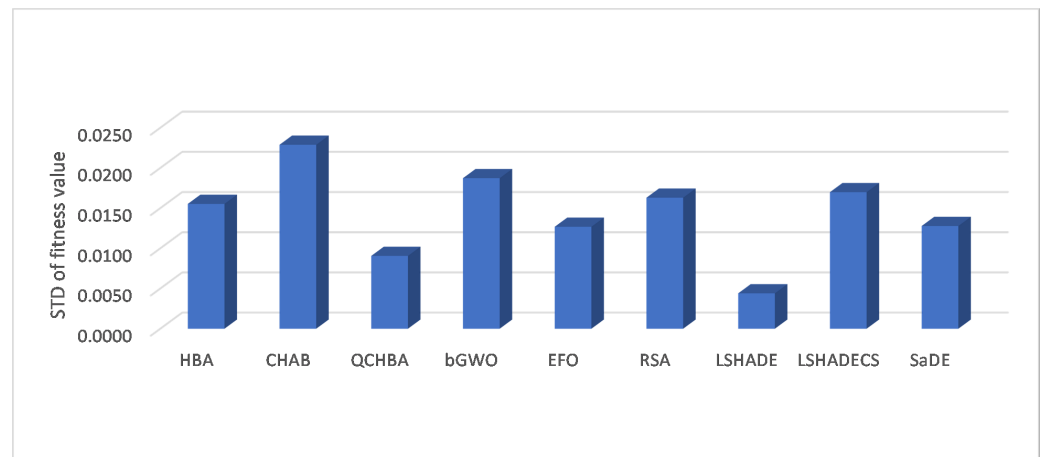


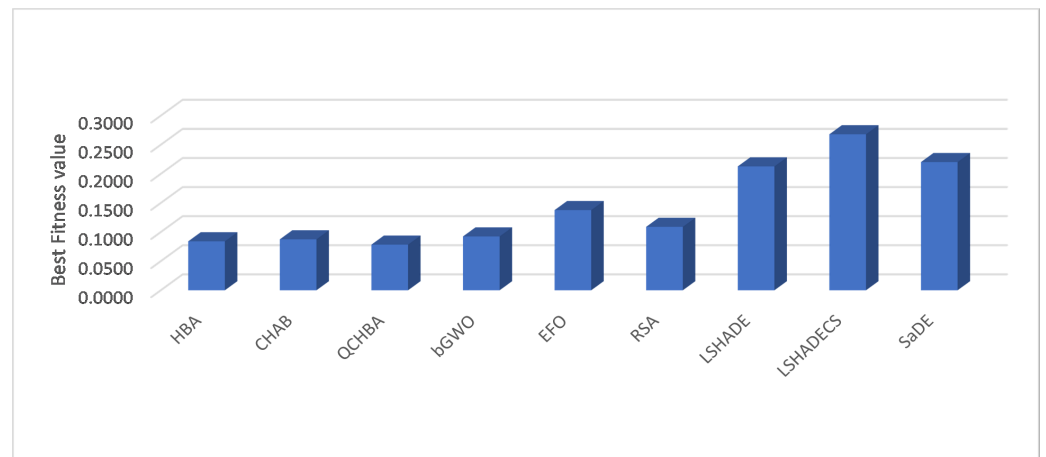
Figure 3. Average of stability results of QCHBA and other algorithms overall for the tested datasets.

Table 3. Stability of competitive algorithms based on fitness values.

|     | HBA    | CHAB   | QCHBA  | bGWO   | EFO    | RSA    | LSHADE        | LSHADECS      | SaDE          |
|-----|--------|--------|--------|--------|--------|--------|---------------|---------------|---------------|
| S1  | 0.0110 | 0.0074 | 0.0092 | 0.0088 | 0.0037 | 0.0073 | <b>0.0000</b> | 0.0106        | <b>0.0000</b> |
| S2  | 0.0139 | 0.0129 | 0.0108 | 0.0112 | 0.0105 | 0.0122 | <b>0.0000</b> | 0.0192        | 0.0236        |
| S3  | 0.0130 | 0.0101 | 0.0016 | 0.0102 | 0.0138 | 0.0056 | <b>0.0000</b> | 0.0293        | 0.0236        |
| S4  | 0.0180 | 0.0452 | 0.0041 | 0.0476 | 0.0282 | 0.0521 | 0.0645        | 0.0000        | 0.0000        |
| S5  | 0.0100 | 0.0364 | 0.0175 | 0.0242 | 0.0095 | 0.0073 | <b>0.0000</b> | 0.0148        | 0.0236        |
| S6  | 0.0240 | 0.0281 | 0.0141 | 0.0315 | 0.0340 | 0.0143 | <b>0.0000</b> | 0.0223        | <b>0.0092</b> |
| S7  | 0.0212 | 0.0236 | 0.0025 | 0.0107 | 0.0033 | 0.0140 | <b>0.0000</b> | 0.0236        | 0.0070        |
| S8  | 0.0126 | 0.0151 | 0.0107 | 0.0107 | 0.0109 | 0.0119 | 0.0054        | 0.0089        | <b>0.0000</b> |
| S9  | 0.0399 | 0.0348 | 0.0183 | 0.0224 | 0.0094 | 0.0070 | 0.0094        | 0.0236        | 0.0096        |
| S10 | 0.0083 | 0.0337 | 0.0018 | 0.0308 | 0.0208 | 0.0565 | <b>0.0000</b> | 0.0422        | 0.0211        |
| S11 | 0.0137 | 0.0491 | 0.0152 | 0.0346 | 0.0007 | 0.0399 | <b>0.0000</b> | 0.0123        | 0.0236        |
| S12 | 0.0211 | 0.0239 | 0.0196 | 0.0190 | 0.0113 | 0.0097 | <b>0.0000</b> | 0.0000        | 0.0000        |
| S13 | 0.0061 | 0.0239 | 0.0024 | 0.0215 | 0.0138 | 0.0155 | <b>0.0000</b> | 0.0144        | 0.0196        |
| S14 | 0.0093 | 0.0162 | 0.0020 | 0.0121 | 0.0119 | 0.0084 | <b>0.0000</b> | 0.0206        | 0.0026        |
| S15 | 0.0216 | 0.0166 | 0.0165 | 0.0133 | 0.0106 | 0.0014 | <b>0.0000</b> | 0.0471        | 0.0236        |
| S16 | 0.0160 | 0.0139 | 0.0106 | 0.0094 | 0.0077 | 0.0080 | <b>0.0000</b> | 0.0032        | 0.0000        |
| S17 | 0.0140 | 0.0107 | 0.0021 | 0.0112 | 0.0172 | 0.0061 | <b>0.0000</b> | 0.0137        | 0.0196        |
| S18 | 0.0058 | 0.0100 | 0.0043 | 0.0081 | 0.0112 | 0.0163 | <b>0.0000</b> | <b>0.0000</b> | 0.0236        |

**Table 4.** Best fitness values generated using competitive methods.

| Min | HBA           | CHAB          | QCHBA         | bGWO          | EFO    | RSA           | LSHADE | LSHADECS | SaDE   |
|-----|---------------|---------------|---------------|---------------|--------|---------------|--------|----------|--------|
| S1  | 0.0462        | <b>0.0415</b> | 0.0608        | 0.0462        | 0.0766 | 0.0766        | 0.0833 | 0.1250   | 0.0917 |
| S2  | 0.0482        | 0.0291        | <b>0.0133</b> | 0.0616        | 0.0782 | 0.0595        | 0.1254 | 0.1860   | 0.0947 |
| S3  | 0.0332        | 0.0560        | 0.0664        | 0.0291        | 0.0791 | <b>0.0166</b> | 0.0655 | 0.1402   | 0.0897 |
| S4  | <b>0.0462</b> | <b>0.0462</b> | <b>0.0462</b> | 0.0538        | 0.0538 | 0.0538        | 0.2048 | 0.2942   | 0.3853 |
| S5  | 0.2192        | <b>0.1562</b> | 0.2327        | 0.2057        | 0.2968 | 0.2794        | 0.2258 | 0.3643   | 0.2118 |
| S6  | 0.1474        | 0.1551        | <b>0.1205</b> | 0.1321        | 0.1282 | 0.1551        | 0.2019 | 0.3426   | 0.2352 |
| S7  | 0.0489        | 0.0518        | <b>0.0303</b> | 0.0781        | 0.1408 | 0.0751        | 0.1160 | 0.1493   | 0.1099 |
| S8  | <b>0.0586</b> | 0.0628        | 0.0615        | 0.0853        | 0.0864 | 0.0758        | 0.3866 | 0.3947   | 0.3658 |
| S9  | 0.1233        | 0.1400        | 0.0810        | <b>0.0800</b> | 0.2163 | 0.1322        | 0.2500 | 0.2500   | 0.2448 |
| S10 | <b>0.0462</b> | <b>0.0462</b> | <b>0.0462</b> | <b>0.0462</b> | 0.0660 | 0.0538        | 0.2118 | 0.3205   | 0.2557 |
| S11 | <b>0.0025</b> | 0.0083        | 0.0037        | 0.0246        | 0.1997 | 0.0760        | 0.3200 | 0.3242   | 0.3333 |
| S12 | 0.0300        | 0.0317        | <b>0.0217</b> | 0.0548        | 0.1262 | 0.1057        | 0.2833 | 0.4167   | 0.3333 |
| S13 | 0.0955        | 0.1515        | 0.1061        | <b>0.0848</b> | 0.2212 | 0.1818        | 0.1630 | 0.2630   | 0.2278 |
| S14 | 0.2214        | 0.2278        | <b>0.2009</b> | 0.2524        | 0.2243 | 0.2196        | 0.2635 | 0.3089   | 0.2974 |
| S15 | 0.0425        | 0.0338        | <b>0.0275</b> | 0.0888        | 0.0950 | 0.0338        | 0.0567 | 0.1117   | 0.0683 |
| S16 | 0.2501        | 0.2595        | <b>0.2440</b> | 0.2711        | 0.3031 | 0.2877        | 0.3574 | 0.4483   | 0.4094 |
| S17 | <b>0.0308</b> | 0.0538        | <b>0.0308</b> | 0.0462        | 0.0635 | 0.0615        | 0.1833 | 0.1722   | 0.1444 |
| S18 | 0.0250        | 0.0250        | <b>0.0188</b> | 0.0250        | 0.0250 | <b>0.0188</b> | 0.3333 | 0.2133   | 0.0667 |



**Figure 4.** The efficiency of QCHBA and other methods according to the best fitness value measure.

**Table 5.** Worst fitness values generated using competitive methods.

| Max | HBA           | CHAB          | QCHBA         | bGWO   | EFO    | RSA           | LSHADE | LSHADECS | SaDE   |
|-----|---------------|---------------|---------------|--------|--------|---------------|--------|----------|--------|
| S1  | 0.0795        | <b>0.0731</b> | 0.0941        | 0.0766 | 0.0860 | 0.0941        | 0.0833 | 0.1400   | 0.0917 |
| S2  | 0.0995        | 0.0782        | <b>0.0646</b> | 0.1032 | 0.1049 | 0.0907        | 0.1254 | 0.2132   | 0.1281 |
| S3  | 0.0769        | 0.0914        | 0.1203        | 0.0685 | 0.1164 | <b>0.0291</b> | 0.0655 | 0.1816   | 0.1230 |
| S4  | <b>0.1020</b> | 0.2164        | <b>0.1020</b> | 0.1939 | 0.1142 | 0.1894        | 0.2960 | 0.2942   | 0.3853 |
| S5  | 0.2640        | 0.2577        | 0.3167        | 0.3058 | 0.3199 | 0.2961        | 0.2258 | 0.3853   | 0.2452 |
| S6  | 0.2359        | 0.2103        | 0.2115        | 0.2385 | 0.2103 | <b>0.1910</b> | 0.2019 | 0.3741   | 0.2481 |
| S7  | 0.1272        | 0.1125        | <b>0.1093</b> | 0.1172 | 0.1496 | 0.1376        | 0.1160 | 0.1826   | 0.1197 |
| S8  | 0.1103        | 0.1060        | <b>0.1049</b> | 0.1214 | 0.1129 | 0.1116        | 0.3942 | 0.4073   | 0.3658 |
| S9  | 0.2522        | 0.2522        | <b>0.1489</b> | 0.1630 | 0.2389 | 0.1598        | 0.2633 | 0.2833   | 0.2583 |
| S10 | <b>0.0795</b> | 0.1600        | 0.0872        | 0.1804 | 0.1097 | 0.1804        | 0.2118 | 0.3802   | 0.2855 |
| S11 | <b>0.0646</b> | 0.1917        | 0.1065        | 0.1492 | 0.2015 | 0.1843        | 0.3200 | 0.3417   | 0.3667 |
| S12 | 0.1090        | 0.1310        | <b>0.1045</b> | 0.1195 | 0.1526 | 0.1295        | 0.2833 | 0.4167   | 0.3333 |
| S13 | <b>0.1212</b> | 0.2318        | 0.1939        | 0.1682 | 0.2561 | 0.2212        | 0.1630 | 0.2833   | 0.2556 |
| S14 | <b>0.2401</b> | 0.2870        | 0.2734        | 0.3057 | 0.2559 | 0.2418        | 0.2635 | 0.3380   | 0.3010 |
| S15 | 0.1225        | 0.0925        | 0.0925        | 0.1363 | 0.1225 | <b>0.0363</b> | 0.0567 | 0.1783   | 0.1017 |
| S16 | 0.3104        | 0.3081        | <b>0.2964</b> | 0.3093 | 0.3210 | 0.3062        | 0.3574 | 0.4528   | 0.4094 |
| S17 | 0.0788        | 0.0942        | 0.1038        | 0.0885 | 0.1019 | <b>0.0769</b> | 0.1833 | 0.1917   | 0.1722 |
| S18 | <b>0.0438</b> | 0.0563        | <b>0.0438</b> | 0.0625 | 0.0500 | 0.0563        | 0.3333 | 0.2133   | 0.1000 |

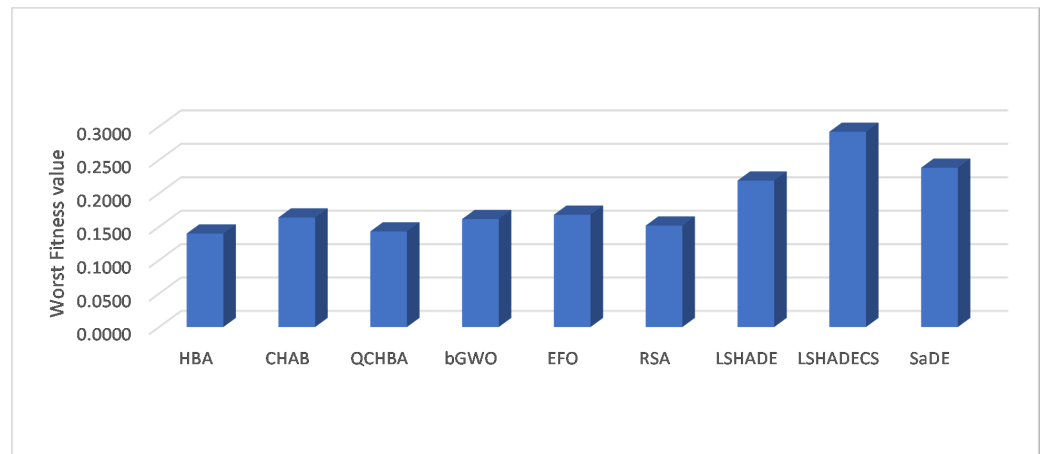


Figure 5. Performance of QCHBA and other methods in terms of worst fitness values.

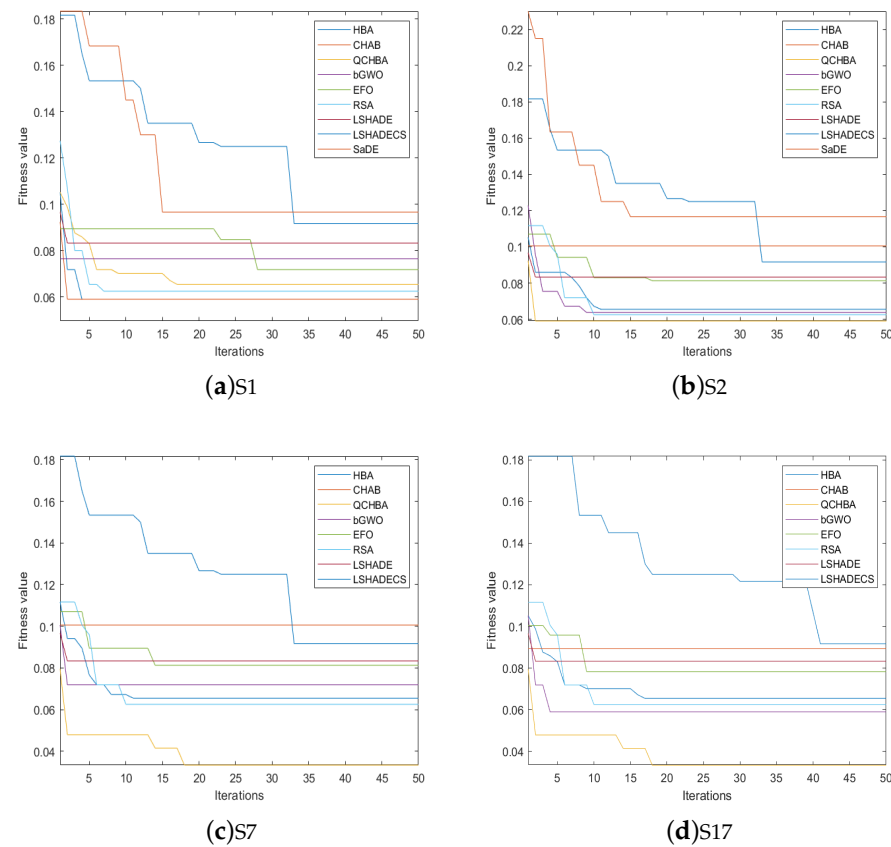


Figure 6. Convergence curves of the competitive algorithms for S1, S2, S7, and S17.

The efficiency of the performance of the QCHBA based on the accuracy measure is shown in Table 6. For example, it obtained the highest accuracy level at 10 datasets, which represents nearly 56% of the total. Each of the HBA, CHBA, GWO, EFO, and RSA methods had best accuracy for only two datasets, including the dataset S18 (which was common to all of them). The average accuracy of the tested datasets is given in Figure 7. It can be seen that the accuracy of the QCHBA was better than the close second algorithm, CHAB, with a difference of nearly 1.45%.

Table 7 and Figure 8 show the average of selected features from each dataset using the FS method. From these results, it can be seen that the QCHBA had the ability to select the smallest number of features, at nearly 50% of the total. RSA was the second best algorithm according to the number of selected features, with the smallest number of features at three

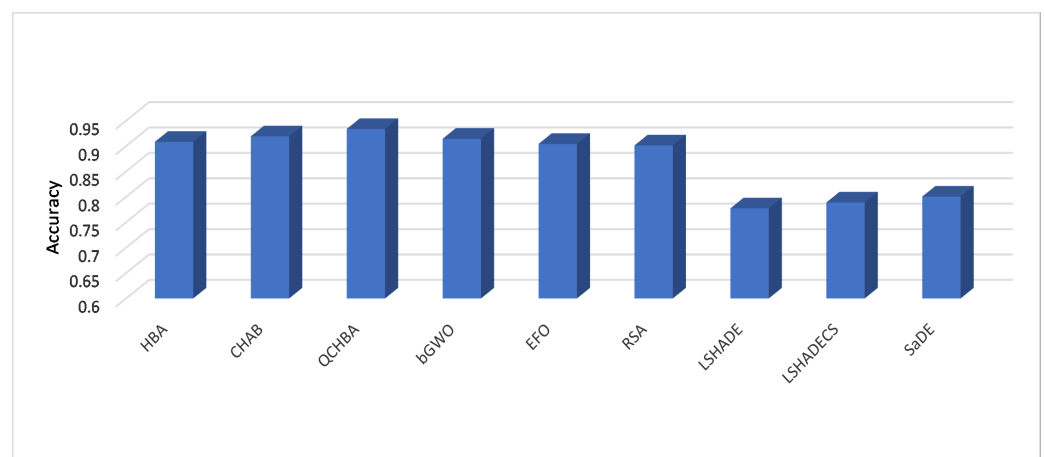
datasets. This was followed by the GWO and CHBA methods with the smallest number of features for two datasets.

**Table 6.** Average of accuracy obtained using QCHNA and other methods.

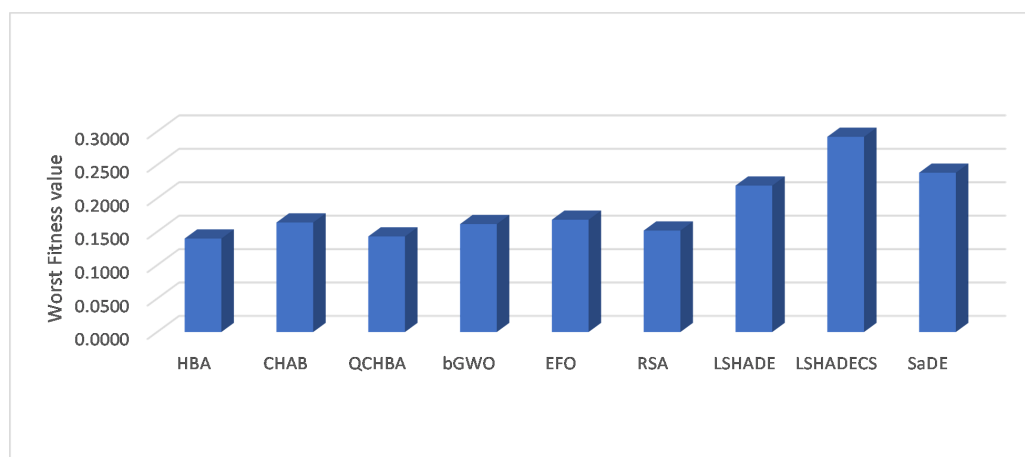
|     | HBA           | CHAB          | QCHBA         | bGWO          | EFO           | RSA           | LSHADE | LSHADECS      | SaDE          |
|-----|---------------|---------------|---------------|---------------|---------------|---------------|--------|---------------|---------------|
| S1  | 0.9479        | 0.9739        | <b>0.9764</b> | 0.9643        | 0.9629        | 0.9529        | 0.9286 | 0.9536        | 0.9429        |
| S2  | 0.9605        | 0.9702        | <b>0.9930</b> | 0.9491        | 0.9684        | 0.9579        | 0.8684 | 0.8816        | 0.9035        |
| S3  | 0.9310        | 0.9638        | 0.9747        | 0.9759        | 0.9609        | <b>0.9885</b> | 0.9540 | 0.9368        | 0.9655        |
| S4  | 0.9935        | 0.9823        | <b>0.9965</b> | 0.9615        | 0.9820        | 0.9510        | 0.7375 | 0.6750        | 0.6700        |
| S5  | 0.7655        | <b>0.8233</b> | 0.7470        | 0.7750        | 0.7270        | 0.7600        | 0.7250 | 0.6550        | 0.7300        |
| S6  | 0.8352        | 0.8546        | <b>0.8898</b> | 0.8269        | 0.8889        | 0.8519        | 0.7593 | 0.7500        | 0.7593        |
| S7  | 0.9254        | 0.9423        | <b>0.9577</b> | 0.9211        | 0.9127        | 0.9296        | 0.9296 | 0.9296        | 0.9437        |
| S8  | 0.9286        | 0.9609        | 0.9598        | 0.9513        | <b>0.9713</b> | 0.9275        | 0.5852 | 0.6414        | 0.5594        |
| S9  | 0.8333        | 0.8317        | <b>0.9361</b> | 0.9340        | 0.8299        | 0.9133        | 0.8000 | 0.8333        | 0.8333        |
| S10 | <b>0.9988</b> | 0.9838        | 0.9975        | 0.9845        | 0.9820        | 0.9440        | 0.7450 | 0.6900        | 0.7100        |
| S11 | 0.9733        | 0.9262        | <b>0.9967</b> | 0.9431        | 0.8667        | 0.8614        | 0.7333 | 0.7386        | 0.8846        |
| S12 | 0.9595        | 0.9643        | <b>0.9738</b> | 0.9560        | 0.9381        | 0.9000        | 0.6905 | 0.5476        | 0.5357        |
| S13 | 0.8185        | 0.8583        | 0.8769        | <b>0.8954</b> | 0.8111        | 0.8037        | 0.8148 | 0.8241        | 0.8519        |
| S14 | 0.8108        | 0.8207        | 0.8070        | 0.7766        | 0.8052        | 0.8188        | 0.7188 | 0.7760        | <b>0.8750</b> |
| S15 | 0.9467        | 0.9708        | 0.9783        | 0.9217        | 0.9467        | 0.9733        | 0.9667 | <b>0.9833</b> | 0.9083        |
| S16 | 0.7451        | 0.7525        | <b>0.7624</b> | 0.7368        | 0.7394        | 0.7118        | 0.5370 | 0.5230        | 0.5170        |
| S17 | 0.9764        | <b>0.9944</b> | 0.9819        | 0.9847        | 0.9833        | 0.9778        | 0.8333 | 0.9306        | 0.9861        |
| S18 | <b>1.0000</b> | <b>1.0000</b> | <b>1.0000</b> | <b>1.0000</b> | <b>1.0000</b> | <b>1.0000</b> | 0.6667 | 0.9333        | 0.8333        |

**Table 7.** Number of selected features.

|     | HBA         | CHAB        | QCHBA        | bGWO        | EFO  | RSA         | LSHADE | LSHADECS | SaDE  |
|-----|-------------|-------------|--------------|-------------|------|-------------|--------|----------|-------|
| S1  | <b>2.85</b> | 3.2         | <b>2.4</b>   | 2.75        | 4.4  | 3.6         | 5      | 5        | 7     |
| S2  | 8.5         | 9.5         | <b>7.65</b>  | 10.15       | 18   | 11.2        | 13     | 13.5     | 14.5  |
| S3  | 6.1         | <b>3.45</b> | 3.65         | 4.1         | 10.2 | 5.4         | 6      | 6.5      | 7.5   |
| S4  | 7.15        | 6.75        | 6.6          | 8.35        | 8.2  | 8.6         | 8      | 6        | 12    |
| S5  | 2.5         | 4.35        | 2.85         | <b>1.65</b> | 8.8  | 9.2         | 8      | 9        | 9.5   |
| S6  | 7.35        | 5.2         | 6.85         | <b>4.7</b>  | 8.6  | 5.2         | 6      | 8.5      | 8     |
| S7  | 9.95        | 14.05       | <b>8.55</b>  | 8.95        | 22.8 | 8.6         | 10     | 9.5      | 13    |
| S8  | 17.65       | 16.35       | 17.2         | 21.1        | 26.4 | <b>10.8</b> | 20     | 19.5     | 21    |
| S9  | <b>6.55</b> | 14          | 10.35        | 8.2         | 12.8 | 11.6        | 11.5   | 10.5     | 15    |
| S10 | 7.05        | <b>6.6</b>  | 6.8          | 8.15        | 8.4  | 8.8         | 10     | 11       | 12.5  |
| S11 | 48.9        | 26.4        | 23.95        | 96.95       | 262  | <b>19.6</b> | 22.5   | 21       | 26    |
| S12 | 26.55       | 27.5        | <b>25.15</b> | 26.6        | 49   | 26.8        | 32.5   | 31.5     | 36    |
| S13 | 8.1         | 6.75        | <b>5.35</b>  | 6.45        | 14.2 | 6.4         | 7.5    | 9        | 9.5   |
| S14 | 6.35        | 6.1         | <b>5.45</b>  | 6           | 6    | 5.8         | 8.5    | 9.5      | 12.5  |
| S15 | 4.95        | 4.75        | <b>3.95</b>  | 6.55        | 9    | 6.8         | 9.7    | 10.7     | 11.2  |
| S16 | 20.7        | 19.35       | 22.75        | 21.8        | 30.8 | <b>15</b>   | 20.4   | 20.9     | 23.4  |
| S17 | 6.75        | 6.7         | <b>6.3</b>   | 6.45        | 8.4  | 6.4         | 8.5    | 8.5      | 11.5  |
| S18 | 6.4         | <b>4.85</b> | <b>4.85</b>  | 7.7         | 6.8  | 5.4         | 11.45  | 11.45    | 13.95 |



**Figure 7.** Performance of ACHBA and others based on the average accuracy.



**Figure 8.** Efficiency of QCHBA and other methods based on the average number of selected features.

From the previous results and discussion, it is evident that integration of the quantum-based optimization (QBO) and the 2D chaotic Hénon map with the honey badger algorithm led to an enhanced ability to discover the subset of relevant features. The QBO and 2D Hénon maps have the ability to balance switching between exploration and exploitation during the search process for the relevant features. Despite these advantages, the proposed QCHBA still suffers from certain limitations that can effect its performance. For example, the stability of QCHBA still requires improvement, and this is critical. In addition, the time complexity of the QCHBA needs to be decreased.

## 5. Conclusions

This paper proposes a modified version of the honey badger algorithm (HBA) using a quantum optimization technique and a 2D chaotic Hénon map. The 2D Hénon map was applied to adjust the parameters of  $C$  and  $\beta$ , as well as to update the digging phase and the density variable. The quantum-based optimization was applied to improve convergence towards the optimal solution. To assess the performance of the proposed approach, a set of experimental investigations was conducted. This was achieved through the use of eighteen datasets with the results compared with other well-known FS-based methods, including, for example, RSA, EFO, GWO, LASHADE, LSHADECS, SaDE, and the traditional HBA. The comparative results indicated strong performance of the proposed method. Moreover, the influence of the QBO and chaotic maps for enhancing the performance of the proposed method was apparent.

In the future, the proposed QCHBA can be applied to solve different optimization problems. For example, it can be used to improve the performance of medical classification, the quality of service in cloud computing environments, investigation of plant diseases and other applications. In addition, it can be applied to handle multi-objective optimization problems.

**Author Contributions:** Conceptualization, D.Y., R.A.I., O.F.H. and M.A.E.; methodology, D.Y., R.A.I. and M.A.E.; software, D.Y., R.A.I., O.F.H. and M.A.E.; validation, D.Y., R.A.I. and M.A.E.; formal analysis, S.A., D.Y., R.A.I. and M.A.E.; investigation, S.A., D.Y., R.A.I. and M.A.E.; writing—original draft preparation, D.Y., R.A.I. and M.A.E.; writing—review and editing, S.A., D.Y., R.A.I., O.F.H. and M.A.E.; visualization, S.A., D.Y., R.A.I. and M.A.E.; supervision, M.A.E., R.A.I. and O.F.H.; project administration, S.A., D.Y., R.A.I. and M.A.E.; funding acquisition, S.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R197), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

**Data Availability Statement:** The data available upon request from corresponding author.



**Conflicts of Interest:** The authors declare that there are no conflicts of interest regarding the publication of this paper.

## References

1. Tubishat, M.; Idris, N.; Shuib, L.; Abushariah, M.A.; Mirjalili, S. Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection. *Expert Syst. Appl.* **2020**, *145*, 113–122. [[CrossRef](#)]
2. Hancer, E.; Xue, B.; Karaboga, D.; Zhang, M. A binary ABC algorithm based on advanced similarity scheme for feature selection. *Appl. Soft Comput.* **2015**, *36*, 334–348. [[CrossRef](#)]
3. Ewees, A.A.; Abualigah, L.; Yousri, D.; Algarni, Z.Y.; Al-qaness, M.A.; Ibrahim, R.A.; Abd Elaziz, M. Improved Slime Mould Algorithm based on Firefly Algorithm for feature selection: A case study on QSAR model. *Eng. Comput.* **2021**, *38*, 2407–2421. [[CrossRef](#)]
4. Yousri, D.; Abd Elaziz, M.; Abualigah, L.; Oliva, D.; Al-Qaness, M.A.; Ewees, A.A. COVID-19 X-ray images classification based on enhanced fractional-order cuckoo search optimizer using heavy-tailed distributions. *Appl. Soft Comput.* **2021**, *101*, 107052. [[CrossRef](#)] [[PubMed](#)]
5. Al-qaness, M.A. Device-free human micro-activity recognition method using WiFi signals. *Geo-Spat. Inf. Sci.* **2019**, *22*, 128–137. [[CrossRef](#)]
6. Dahou, A.; Elaziz, M.A.; Zhou, J.; Xiong, S. Arabic sentiment classification using convolutional neural network and differential evolution algorithm. *Comput. Intell. Neurosci.* **2019**, *2019*, 2537689. [[CrossRef](#)]
7. Rundo, L.; Tangherloni, A.; Cazzaniga, P.; Nobile, M.S.; Russo, G.; Gilardi, M.C.; Vitabile, S.; Mauri, G.; Besozzi, D.; Militello, C. A novel framework for MR image segmentation and quantification by using MedGA. *Comput. Methods Programs Biomed.* **2019**, *176*, 159–172. [[CrossRef](#)]
8. Ortiz, A.; Górriz, J.; Ramírez, J.; Salas-Gonzalez, D.; Llamas-Elvira, J.M. Two fully-unsupervised methods for MR brain image segmentation using SOM-based strategies. *Appl. Soft Comput.* **2013**, *13*, 2668–2682. [[CrossRef](#)]
9. Cheng, S.; Ma, L.; Lu, H.; Lei, X.; Shi, Y. Evolutionary computation for solving search-based data analytics problems. *Artif. Intell. Rev.* **2021**, *54*, 1321–1348. [[CrossRef](#)]
10. Nobile, M.S.; Tangherloni, A.; Rundo, L.; Spolaor, S.; Besozzi, D.; Mauri, G.; Cazzaniga, P. Computational intelligence for parameter estimation of biochemical systems. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
11. Ibrahim, R.A.; Abualigah, L.; Ewees, A.A.; Al-Qaness, M.A.; Yousri, D.; Alshathri, S.; Abd Elaziz, M. An electric fish-based arithmetic optimization algorithm for feature selection. *Entropy* **2021**, *23*, 1189. [[CrossRef](#)]
12. Ibrahim, R.; Ewees, A.; Oliva, D.; Abd Elaziz, M.; Lu, S. Improved salp swarm algorithm based on particle swarm optimization for feature selection. *J. Ambient. Intell. Hum. Comput.* **2019**, *10*, 3155–3169. [[CrossRef](#)]
13. Aziz, M.A.E.; Hassanien, A.E. Modified cuckoo search algorithm with rough sets for feature selection. *Neural Comput. Appl.* **2018**, *29*, 925–934. [[CrossRef](#)]
14. Abd Elaziz, M.; Moemen, Y.S.; Hassanien, A.E.; Xiong, S. Toxicity risks evaluation of unknown FDA biotransformed drugs based on a multi-objective feature selection approach. *Appl. Soft Comput.* **2020**, *97*, 105509. [[CrossRef](#)]
15. Ibrahim, R.A.; Abd Elaziz, M.; Ewees, A.A.; El-Abd, M.; Lu, S. New feature selection paradigm based on hyper-heuristic technique. *Appl. Math. Model.* **2021**, *98*, 14–37. [[CrossRef](#)]
16. Xue, B.; Zhang, M.; Browne, W.N. Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE Trans. Cybern.* **2012**, *43*, 1656–1671. [[CrossRef](#)] [[PubMed](#)]
17. Hancer, E. Differential evolution for feature selection: A fuzzy wrapper-filter approach. *Soft Comput.* **2019**, *23*, 5233–5248. [[CrossRef](#)]
18. Tsai, C.F.; Eberle, W.; Chu, C.Y. Genetic algorithms in feature and instance selection. *Knowl.-Based Syst.* **2013**, *39*, 240–247. [[CrossRef](#)]
19. Sayed, G.I.; Hassanien, A.E.; Azar, A.T. Feature selection via a novel chaotic crow search algorithm. *Neural Comput. Appl.* **2019**, *31*, 171–188. [[CrossRef](#)]
20. Oh, I.S.; Lee, J.S.; Moon, B.R. Hybrid genetic algorithms for feature selection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 1424–1437.
21. Abdel-Basset, M.; Ding, W.; El-Shahat, D. A hybrid Harris Hawks optimization algorithm with simulated annealing for feature selection. *Artif. Intell. Rev.* **2021**, *54*, 593–637. [[CrossRef](#)]
22. Sahlol, A.T.; Yousri, D.; Ewees, A.A.; Al-Qaness, M.A.; Damasevicius, R.; Elaziz, M.A. COVID-19 image classification using deep features and fractional-order marine predators algorithm. *Sci. Rep.* **2020**, *10*, 15364. [[CrossRef](#)] [[PubMed](#)]
23. 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](#)]
24. Yilmaz, S.; Sen, S. Electric fish optimization: a new heuristic algorithm inspired by electrolocation. *Neural Comput. Appl.* **2020**, *32*, 11543–11578. [[CrossRef](#)]
25. Hashim, F.A.; Houssein, E.H.; Hussain, K.; Mabrouk, M.S.; Al-Atabany, W. Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems. *Math. Comput. Simul.* **2022**, *192*, 84–110. [[CrossRef](#)]

26. Elseify, M.A.; Kamel, S.; Abdel-Mawgoud, H.; Elattar, E.E. A Novel Approach Based on Honey Badger Algorithm for Optimal Allocation of Multiple DG and Capacitor in Radial Distribution Networks Considering Power Loss Sensitivity. *Mathematics* **2022**, *10*, 2081. [[CrossRef](#)]
27. Almodfer, R.; Abd Elaziz, M.; Alshathri, S.; Abualigah, L.; Mudhsh, M.; Shahzad, K.; Issa, M. Improving Parameters Estimation of Fuel Cell Using Honey Badger Optimization Algorithm. *Front. Energy Res.* **2022**, *10*, 875332. [[CrossRef](#)]
28. Nassef, A.M.; Houssein, E.H.; Helmy, B.E.D.; Rezk, H. Modified honey badger algorithm based global MPPT for triple-junction solar photovoltaic system under partial shading condition and global optimization. *Energy* **2022**, *254*, 124363. [[CrossRef](#)]
29. Kumar, D.S.R.; Kumar, K.P.; Raju, K.G.; Gowsalya, S.; Balraj, L.; Srivastava, A.K. An IoT-based Optimization scheme on task scheduling for minimizing energy in Cloud Computing. In Proceedings of the 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 5–26 March 2022; Volume 1, pp. 1–6.
30. Hénon, M. A two-dimensional mapping with a strange attractor. In *The Theory of Chaotic Attractors*; Springer: New York, NY, USA, 1976; pp. 94–102.
31. Asuncion, A.; Newman, D. UCI Machine Learning Repository. 2010. Available online: <https://archive.ics.uci.edu/ml/datasets/SML2010> (accessed on 14 September 2022).
32. Awad, N.H.; Ali, M.Z.; Suganthan, P.N.; Reynolds, R.G. An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving CEC2014 benchmark problems. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; pp. 2958–2965.
33. Ibrahim, R.A.; Abd Elaziz, M.; Lu, S. Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization. *Expert Syst. Appl.* **2018**, *108*, 1–27. [[CrossRef](#)]
34. Abualigah, L.; Abd Elaziz, M.; Sumari, P.; Geem, Z.W.; Gandomi, A.H. Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer. *Expert Syst. Appl.* **2022**, *191*, 116158. [[CrossRef](#)]
35. Mohamed, A.W.; Hadi, A.A.; Fattouh, A.M.; Jambi, K.M. LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems. In Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC), San Sebastián, Spain, 5–8 June 2017; pp. 145–152.
36. Qin, A.K.; Suganthan, P.N. Self-adaptive differential evolution algorithm for numerical optimization. In Proceedings of the 2005 IEEE Congress on Evolutionary Computation, Scotland, UK, 2–5 September 2005; Volume 2, pp. 1785–1791.