

Article

A Quantum-Based Chameleon Swarm for Feature Selection

Mohamed Abd Elaziz ^{1,2,3,4,*} , Mahmoud Ahmadein ^{5,6} , Sabbah Ataya ⁵ , Naser Alsaleh ⁵ ,
Agostino Forestiero ⁷  and Ammar H. Elsheikh ⁶ 

- ¹ Department of Mathematics, Faculty of Science, Zagazig University, Zagazig 44519, Egypt
² Artificial Intelligence Research Center (AIRC), College of Engineering and Information Technology, Ajman University, Ajman P.O. Box 346, United Arab Emirates
³ Faculty of Computer Science and Engineering, Galala University, Suez 435611, Egypt
⁴ Department of Electrical and Computer Engineering, Lebanese American University, Byblos 13-5053, Lebanon
⁵ Mechanical Engineering Department, Imam Mohammad Ibn Saud Islamic University, Riyadh 11564, Saudi Arabia
⁶ Department of Production Engineering and Mechanical Design, Faculty of Engineering, Tanta University, Tanta 31527, Egypt
⁷ Institute for High Performance Computing and Networking, National Research Council of Italy, 87036 Rende, Italy
* Correspondence: abd_el_aziz_m@yahoo.com

Abstract: The Internet of Things is widely used, which results in the collection of enormous amounts of data with numerous redundant, irrelevant, and noisy features. In addition, many of these features need to be managed. Consequently, developing an effective feature selection (FS) strategy becomes a difficult goal. Many FS techniques, based on bioinspired metaheuristic methods, have been developed to tackle this problem. However, these methods still suffer from limitations; so, in this paper, we developed an alternative FS technique, based on integrating operators of the chameleon swarm algorithm (Cham) with the quantum-based optimization (QBO) technique. With the use of eighteen datasets from various real-world applications, we proposed that QCham is investigated and compared to well-known FS methods. The comparisons demonstrate the benefits of including a QBO operator in the Cham because the proposed QCham can efficiently and accurately detect the most crucial features. Whereas the QCham achieves nearly 92.6%, with CPU time(s) nearly 1.7 overall the tested datasets. This indicates the advantages of QCham among comparative algorithms and high efficiency of integrating the QBO with the operators of Cham algorithm that used to enhance the process of balancing between exploration and exploitation.

Keywords: feature selection (FS); metaheuristic (MH); chameleon swarm algorithm (Cham); quantum-based optimization (QBO)

MSC: 68Txx



Citation: Elaziz, M.A.; Ahmadein, M.; Ataya, S.; Alsaleh, N.; Forestiero, A.; Elsheikh, A.H. A Quantum-Based Chameleon Swarm for Feature Selection. *Mathematics* **2022**, *10*, 3606. <https://doi.org/10.3390/math10193606>

Academic Editors: Florin Leon and Hsien-Chung Wu

Received: 19 August 2022

Accepted: 27 September 2022

Published: 2 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

Recently, the Internet of Things (IoT) has more attention, since it can be applied to different applications. For example, medical [1–3], agriculture [4], industrial [5,6], and education [7]. However, the rapid growth in data volume, as well as its complexity, is the main reason for many challenging problems, such as the noisy nature, complex dimensionality, and irrelevance [8]. Consequently, these problems impair the accuracy and efficiency of the machine learning systems and result in a long computational time. Feature selection (FS) methods have been proposed to boost the accuracy of the classification process and reduce the corresponding computational costs [9]. FS approaches are usually employed to capture the properties of data by determining the subset with the most relevant features [10]. Moreover, they are employed to remove noisy irrelevant data [10]. Thus, FS techniques have been extensively used in various engineering fields, such as text

classification [11], human motion detection [12], classification of computerized tomography images (especially for COVID-19 cases) [13], neuromuscular disorders [14], parameter determination of biochemical systems [15], data analysis problems [16], segmentation of magnetic resonance images [17], and other applications [18].

There are three key kinds of FS techniques, namely filter, wrapper, and embedded [18]. Filter-based methods utilize the datasets properties, while wrapper-based techniques utilize the learning approach to evaluate the selected features. Embedded-based methods focus on learning features with best contribution to the robustness and accuracy of the classification system during its creation process. Thus, wrapper-based methods have better efficiency, but higher computational costs, compared with filter-based techniques. Generally speaking, the main criteria used to assess the performance of the FS method are the minimization of the selected features, total number, and maximization of the classifier accuracy [18]. One of the most popular wrapper-based techniques depends on the Metaheuristic (MH) algorithms.

In general, MH optimizers have been commonly employed to solve complex optimization problems, such as parameter selection of solar cells [19], optimizing neural networks [20], and FS. Several MH optimizers have been developed in the literature as robust tools that are successfully utilized to solve FS problems. The most common used MH optimizers used in this direction are particle swarm optimization (PSO) [21], genetic algorithms (GA) [22], firefly algorithm [23], manta ray foraging optimizer [24], grey wolf optimizer [25], differential evolution (DE) [26], Harris hawk optimizer (HHO) [27], Henry gas optimizer [28,29], political optimizer [30], flower pollination optimizer [31], Aquila optimizer [32], crow search optimizer [33], cat swarm optimizer [34], and ecosystem-based optimizer [35]. According to the no free lunch theorem (NFLT), there is not a single algorithm that can be successfully used for solving all problems. Therefore, the hybridization conceptualization was widely employed to solve several intricate problems, such as FS. In this study, considering the hybridization conceptualization, a new FS approach is proposed, utilizing an innovative algorithm called chameleon swarm optimizer (Cham).

Cham is a newly developed MH approach that mimics the behavior of chameleon during the process of hunting the food sources in jungles or deserts. According to these behaviors, Cham has been applied to handle a set of variant applications. For example, the economic load dispatch problem [36], optimization of automobile cruise control [37], parameters extraction of solid oxide fuel cells [38], global optimization and feature selection [39], and segmentation of plant disease [40]. However, it suffers from getting stuck in local minima as well as it has very slow convergence when the optimization problem has high dimensions. To overcome these drawbacks of the standalone Cham algorithm, quantum-based optimization is used. QBO has been successfully employed in several engineering applications, such as parameter identification of PEM fuel cell [41], wrapper feature selection [42], and inverse problems [43].

In this study, a hybrid MH optimizer called QCham is proposed, which is composed of conventional Cham optimized by QBO. It is utilized to split the tested dataset into two main subsets, namely training, which represents 70% of the whole data, and test sets, which represent 30% of the whole data. At that, the initial value for individuals that characterizes the FS solutions is defined. Boolean versions of individuals set are computed to evaluate their efficacy. The fitness value is also calculated, considering the features of Boolean ones. After that, the individual with the best fitness value is determined and defined as the best solution. After that, the MH operators of Cham and QBO are utilized in the exploration stage to determine those regions with the optimal solutions, which are called the feasible region. This enhances the convergence process to obtain the optimal solution. Furthermore, conventional Cham operators are used during the exploitation stage. The procedures of augmenting individuals' value are carried out until a stopping criterion is fulfilled. Then, the number of the data in the testing set is decreased, considering the features of the best solution. Finally, the performance of the proposed algorithm is calculated using various statistical measures. To our best knowledge, there are no previous studies on the applications of Cham or its improved version in FS problems.

The main contributions and objectives of this study are listed as follows:

- Develop an innovative improved version of Cham using the mathematical operators of QBO to improve exploration capability.
- Employ the enhanced hybrid QCham algorithm as a new FS method to detect and eliminate the irrelevant features, which results in improving the accuracy and efficiency of the classification process.
- Evaluate the efficiency of the proposed QCham using eighteen UCI datasets and compare its performance with other well-known conventional FS approaches.

The remaining parts of this article are structured as follows. Section 2 presents the FS methods applied in the literature to solve the investigated problem. Section 3 discusses the fundamentals and mathematical representation of Cham and ABO. Section 4 presents the main the stages of the proposed approach. Section 5 illustrates and discusses the experimental results. Finally, the main conclusion of the study and future prospects are introduced in Section 6.

2. Related Works

Herein, the main FS methods applied in the literature to solve the investigated FS problem are highlighted. Chaudhuri and Sahu [44] developed an improved crow search algorithm (CSA) and employed it in FS problems. They regulate both stages of the search process (exploitation and exploration) by applying the time-varying flight length of the crows. They assessed many variants of FS models, and they verified the algorithm accuracy using 20 UCI datasets.

Butterfly optimizer (BO) and information theory are integrated to form a hybrid FS algorithm [45]. The hybrid algorithm overcomes the main drawbacks of the conventional BO model, and it exhibited excellent performance, compared to other well-known MH models. Maleki et al. [46] employed a conventional genetic algorithm as a FS technique, along with k-nearest neighbors classifier in classification of lung cancer disease. The application of GA in this classification process enhanced classification. Song et al. [47] proposed PSO algorithm with a new variant as a FS approach, this new algorithm is called bare bones PSO. The key idea of the proposed algorithm is to utilize a swarm-based initialization procedure based on label correlation. The exploitation process was also enhanced using two operators called deletion and supplementary operators. Furthermore, an adaptive mutation operator was used to avert getting stuck in local minima. The proposed algorithm was applied for many datasets, along with kNN, and it had a superior performance, compared with other well-known MH algorithms.

In another study, a hybrid FS model composed of the grey wolf optimizer (GWO) and rough set was developed to analyze mammogram images [48]. The proposed hybrid algorithm well-known outperformed the FS techniques, based on the comparison investigation carried out. Tubishat et al. [49] established a FS algorithm considering a dynamic version of salp swarm optimizer (SSO). The new version of SSO was established based on two approaches. The first approach was employed to control the position updating of salps, while the second approach was employed to enhance the search capabilities of the conventional SSA. The proposed SSA was employed, along with a conventional kNN classifier, and assessed using recognized datasets. Moreover, it was compared with the conventional SSA and other well-known MH algorithms and exhibited outperformance.

Dhiman et al. [50] employed a modified binary version of emperor penguin optimizer (EPO) to solve FS problems. The proposed approach was compared with other well-known approaches, considering twenty-five datasets. Generally, the binary EPO exhibited excellent performance compared with the conventional EPO. A hybrid Elastic Net and genetic algorithm was employed for FS [51]. Neggaz et al. [52] proposed a FS approach considering the Henry gas solubility optimizer (HGSO). The authors in [10] developed a FS method considering two MH approaches, namely the firefly and slime mould algorithms. The developed method was assessed using various datasets, such as the QSAR datasets.

Yousri et al. [13] presented a FS technique to improve the classifications process of COVID-19 CT images. A modified cuckoo search (CS) optimizer was employed for this target. The heavy-tailed distributions and fractional-order (FO) calculus were employed to enhance the conventional CS algorithm. In general, there are many standalone and hybrid MH approaches that have been established for different FS applications, and they outperformed other traditional techniques, as discussed in [53,54].

3. Background

3.1. Chameleon Swarm Optimizer

Chameleon swarm optimizer [55] is a bioinspired model that simulates the natural behavior of chameleon during their hunting process in different environments. Chameleons have the excellent ability to change their color to match their surroundings. They have outstanding eyesight with a clear vision range of 10 m, which enables them to monitor their preys. These preys are grasshoppers, small birds, snails, lizards, crickets, and mantis. However, chameleons themselves may be preys to snakes and birds, and they protect themselves from hunting via adapting their color with surroundings. Chameleons move in deserts and climb trees to hunt for prey. Their excellent eyesight enables them to discover the search space to find and trace their prey. They have ability to look in two directions at the same time. Each eye has its independent motion during tracking process of prey. This help chameleons to detect and track two preys simultaneously. They have a superior capability to track prey, with a 360° full stereoscopic view. Chameleons feed via clingy tongues, which stick to the prey’s body, while employing different surface operations, such as entanglement and wet adhesion with very high tongue acceleration (2590 m/s²). The developed algorithm mimics the natural hunting process of chameleons, including capturing, pursuing, and tracking. The mathematical formulation of that optimizer is discussed in this section.

First the optimizer defines its initial population in a search space with n-dimensions using *m* chameleons. Each chameleon represents a solution to the optimization problem and has a position at a certain iteration *t*, given by [55]:

$$x_t^j = [x_{t,1}^j, x_{t,2}^j, x_{t,3}^j, \dots, x_{t,n}^j] \tag{1}$$

where *j* = 1, 2, 3, . . . , *m*, and *m* are the total number of chameleons, *t* denotes the iteration number, *n* denotes the problem dimensions, and *x*_{*t*,*n*}^{*j*} denotes the chameleon position.

The population is randomly initiated, according to the following formula [55]:

$$x^j = LB_j + r \times (UB_j - LB_j) \tag{2}$$

where *x*^{*j*} denotes the initial position of *j*th chameleon, *r* denotes random number created in the range of [0, 1], and *UB* and *LB* denote the upper and lower bounds of the search space, respectively.

The solution quality is evaluated based on a fitness function, computed at each new chameleon position. Then, the recent position is updated to augment the solution quality. The chameleon does not move if the new position has lower quality, compared with the recent one. During execution, the search process on preys, the chameleon position is updated, according to the following formula:

$$x_{t+1}^{j,i} = \begin{cases} x_t^{j,i} + s_1(S_t^{j,i} - W_t^i)r_2 + s_2(W_t^i - x_t^{j,i})r_1 & r_i \geq S_s \\ x_t^{j,i} + z((UB^i - LB^i)r_3 - LB_b^i)sgn(rand - 0.5) & r_i < S_s \end{cases} \tag{3}$$

where *x*_{*t*+1}^{*j,i*} denotes the updated position at the *t* + 1 iteration, *x*_{*t*}^{*j,i*} denotes the present position at the *t* iteration, *S*_{*t*}^{*j,i*} denotes the best position at iteration *t*, *W*_{*t*}^{*i*} denotes the global best solution at iteration *t*, *s* denotes a positive control parameters, *r*₁, *r*₃, and *r*_{*i*}

denote random value created in $[0, 1]$, S_s denotes perceiving probability, and z is time decay number.

The line connects S and W is given by:

$$L(r_1) = r_1S + (1 - r_1)W \tag{4}$$

The positions of chameleons could be characterized considering line path as follows:

$$F(r_1, r_2) = r_1L + (1 - r_2)E \quad 0 \leq r_2 \leq 1 \tag{5}$$

where $E = x_t^{j,i}$ is a position in an affine plane with other two positions S and W .

Chameleons can track the prey position by exploiting its excellent vision capabilities via rotating their eyes over 360° . A chameleon updates its position via sequential movements and rotation steps to reach the prey. The algorithm changes chameleon original position to its gravity center. Then, it computes the rotation matrix of the prey. Chameleon position is updated based on the computed rotation matrix at its gravity center. The chameleon is translated back to the previous position. The updated position of the chameleon is computed according to the following formula:

$$x_{t+1}^j = xr_t^j + \bar{x}_t^j \tag{6}$$

where x_{t+1}^j denotes the updated position after rotation, \bar{x}_t^j denotes the original gravity center of the chameleon, and xr_t^j denotes the coordinates of rotation center, which is given as:

$$xr_t^j = k \times xc_t^j \tag{7}$$

where k denotes the rotation matrix, and xc_t^j denotes the coordinates of rotation center at iteration t , and it is given by:

$$xc_t^j = x_t^j - \bar{x}_t^j \tag{8}$$

where x_t^j denotes the current position at iteration t . Based on the velocity of chameleon around the prey, its new position is computed as follows:

$$x_{t+1}^{j,i} = x_t^{j,i} + \left((v_t^{j,i})^2 - (v_{t-1}^{j,i})^2 \right) / a_x, \quad a_x = 2 \left(2590 \times \left(1 - e^{-\log(t)} \right) \right) \tag{9}$$

where $v_t^{j,i}$ and $v_{t-1}^{j,i}$ denote chameleon acceleration, chameleon velocity at iterations t and $t - 1$, respectively. The procedures of the CGO algorithm is given in Algorithm 1.

3.2. Quantum-Based Optimization (QBO)

In the quantum-based optimization (QBO) technique, the binary number is used to represent the features, and this indicates whether they will be selected (1) or removed (0). In QBO, each feature is denoted by a quantum bit (i.e., Q-bit (q)), where q refers to superposition of binary value (i.e., '1' and '0'). The mathematical formulation of Q-bit(q) can be formulated using the following equation [39].

$$q = \alpha + i\beta = e^{i\theta}, \quad |\alpha|^2 + |\beta|^2 = 1 \tag{10}$$

where α and β denote the probability of the value of the Q-bit being '0' and '1', respectively. Whereas θ refers the angle of q and it is updated using $\tan^{-1}(\alpha/\beta)$.

The QBO aims to determine the change in the value of q by computing the value of $\Delta\theta$, and this is formulated as:

$$q(t + 1) = q(t) \times R(\Delta\theta) = [\alpha(t) \beta(t)] \times R(\Delta\theta) \tag{11}$$

$$R(\Delta\theta) = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \tag{12}$$

where $\Delta\theta_{ij}$ refers to the rotation angle of i th Q-bit of j th Q-solution, and it is determined according to the best solution X_b ad, predefined in Table 1, based on the experimental tests conducted on the knapsack problems [56].

Algorithm 1. A pseudo-code of CGO algorithm.

1. Set $S_s = 0.1$
2. The coordinates of rotation center of chameleon j at iteration t is given by $xr_t^j = k \times xc_t^j$
3. Initialize the positions and velocities of chameleons.
4. While ($t < T$) do
5. for $j = 1$ to m do
6. for $i = 1$ to n do
7. if $r^j \geq S_s$ then
8. $x_{t+1}^{j,i} = x_t^{j,i} + s_1(S_t^{j,i} - W_t^i)r_2 + s_2(W_t^i - x_t^{j,i})r_1$
9. else
10. $x_{t+1}^{j,i} = x_t^{j,i} + z((UB_j - LB_j)r_3 - LB_j)sgn(rand - 0.5)$
11. end if
12. end for
13. end for
14. for $j = 1$ to m do
15. for $i = 1$ to n do
16. $x_{t+1}^{j,i} = x_t^{j,i} + ((v_t^{j,i})^2 - (v_{t-1}^{j,i})^2) / (5180 \times (1 - e^{-\log(t)}))$
17. end for
18. end for
19. Update positions of chameleon based on predefined LB and UB
20. Set $t = t + 1$
21. end while

Table 1. Predefined value of $\Delta\theta_{ji}$.

$x_t^{j,i}$	x_b	$f(x_t^j) \geq f(x_b)$	$\Delta\theta_{ji}$
0	0	F	0
0	1	F	0.01π
1	0	F	-0.01π
1	1	F	0
0	0	T	0
0	1	T	0
1	0	T	0
1	1	T	0

4. Proposed QCham Method

Figure 1 shows the procedures of the developed FS approach, which is based on boosting Cham efficiency employing quantum-based technique (QBO). The primary goal of employing QBO is to improve the ability of balance between the exploration and exploitation during process of searching for a feasible solution. The proposed FS method, QCham, starts by separating the data into 70% and 30% training and testing sets, respectively. The random numerical values for N chameleons are then allocated, and the fitness value for each of them is calculated. Then, as the best chameleon, the person with the highest fitness value is used. Following this, the solution is updated using Cham exploitation. The procedure for updating individuals is repeated until the stop criteria are met. Following that, the

dimension of the testing set is decreased, based on the best solution, and the implemented QCham as FS is evaluated using several metrics. The QCham is described in depth in the following paragraphs.

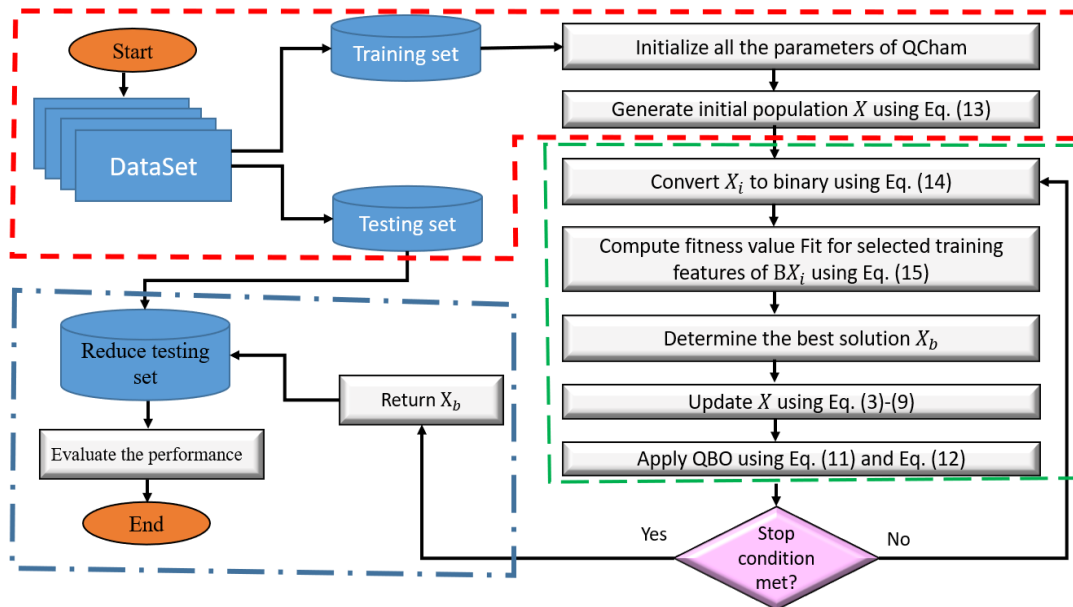


Figure 1. Procedures of the QCham-based FS technique.

4.1. First Phase

The initial chameleons, which represent the solutions’ population, are created at this step. Where each solution contains D Q-bits (D is the number of features). Therefore, the solution X_j can be formulated as in Equation (13).

$$X_j = [q_{i1}|q_{i2}| \dots |q_{iD}] = [\theta_{j1}|\theta_{j2}| \dots |\theta_{jD}], j = 1, 2, \dots, m \tag{13}$$

In this equation, X_j refers to a set of superpositions of probabilities of those feature that are either selected or not.

4.2. Second Phase

Within this phase of QCham, the solutions are updated until they reach the terminal criteria. The first step to accomplish this process is to convert X_j into a binary form ($BX_{j,i}$) using Equation (14):

$$BX_{j,i} = \begin{cases} 1 & \text{if } \text{rand} < |\beta|^2 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

where $\text{rand} \in [0, 1]$ is a random value. The next step is to learn the KNN classifier using the training features allocated at the indices that correspond to the ones in $BX_{j,i}$ and calculate the fitness value, which is given as:

$$Fit_j = \rho \times \gamma + (1 - \rho) \times \left(\frac{|BX_{j,i}|}{D} \right) \tag{15}$$

In Equation (15), $|BX_{j,i}|$ refers to the total number of selected features, and γ denotes the error classification using the KNN classifier. $\rho \in [0, 1]$ is the value that equalizes the fitness value of two parts. The key reason for choosing KNN is that it is simple and efficient, and it has just one parameter. The next step is to allocate the best solution X_b that has the smallest Fit_b . Then, use the operators of Cham, as discussed in Equations (3)–(9).

4.3. Third Phase

Within this third phase, the test set is decreased by choosing only the values corresponding ones in BX_b (step 12 in Algorithm 2). At that, the reduced test set is fed into the classifier (KNN). The next procedure is to assess the output quality using various measures (step 13 in Algorithm 2). Algorithm 2 shows the procedures of QCham.

Algorithm 2. Procedures of QCham

1. Input: Number of iterations ($tmax$), tested dataset with D features, number of solutions (N), and other parameters

First Stage

2. Construct training and testing sets, which represents 70% and 30%.

3. Apply Equation (13) to construct the population X .

Second Stage

4. $t = 1$

5. While ($t < tmax$)

6.

7. Using Equation (14) to obtain the Quantum version of X_j .

8. Calculate fitness value of X_j according to training sample as in Equation (15).

9. Allocate the best solution X_b .

10. Using Equations (3)–(9) to update X

11. $t = t + 1$

12. EndWhile

Third Stage

13. Remove irrelevant features from testing set using X_b .

14. Assess the efficiency of QCham using different measures.

5. Experimental Results

This section assesses the QCham approach's performance on eighteen benchmark datasets. In addition, the results are compared with other methods, including LSHADE [57], self-adaptive differential evolution (SaDE) [58], teaching-learning-based optimizer (TLBO) [59], L-SHADE with semi-parameter adaptation (LSHCMA) [60], grey wolf optimizer (GWO) [61], genetic algorithm (GA), and whale optimizer (WOA) [62].

5.1. Description of Dataset and Setting of Parameter

Table 2 lists the descriptions of eighteen UCI datasets. These datasets were obtained from several real-world applications, and they have distinct features, as shown in the table. Furthermore, Cham, LSHADE, SaDE, LSPACMA, GWO, GA, TLBO, and WOA are compared to the developed Qcham. Each algorithm's parameter is determined depending on its original implementation. The number of iterations and chameleons are common parameters among these approaches, and we set them to 30 and 20, respectively. Furthermore, each of the FS techniques is tested 25 times to ensure that the fair of comparison. We used six criteria to measure the performance: average, standard deviation (Std) of the fitness, best (MIN), worst (MAX), and accuracy (Acc).

5.2. Experimental Results and Discussion

In this section, the results of QCham and other are discussed in this section. Table 3 shows the FS outcomes for all techniques, using the fitness function values as an average. We can see from the chart that the suggested QCham was ranked top in thirteen of the 18 datasets (D1–D4, D6–D8, D10, D12, D14, and D16–D18). The WOA approach came in second with three out of the 18 datasets (D3, D9, and D18). The Cham, GWO, and LSPACMA has the best value at only one dataset, namely D9, D5, and D13. Moreover, Figure 2 shows the average of each algorithm over all the 18 datasets. It can be seen from this average that the developed QCham has the smallest fitness value among the good algorithms. This is followed by traditional Cham and WOA, which provided better fitness values than other methods.

Table 2. Description of datasets.

Data Code	Datasets	No. Instances	No. Features	No. Classes	Data Code	Datasets	No. Instances	No. Features	No. Classes
S1	Breastcancer	699	9	2	S2	BreastEW	569	30	2
S3	CongressEW	435	16	2	S4	Exactly	1000	13	2
S5	Exactly2	1000	13	2	S6	HeartEW	270	13	2
S7	IonosphereEW	351	34	2	S8	KrvskpEW	3196	36	2
S9	Lymphography	148	18	2	S10	M-of-n	1000	13	2
S11	PenglungEW	73	325	2	S12	SonarEW	208	60	2
S13	SpectEW	267	22	2	S14	tic-tac-toe	958	9	2
S15	Vote	300	16	2	S16	WaveformEW	5000	40	3
S17	WaterEW	178	13	3	S18	Zoo	101	16	6

Table 3. Results of fitness values for QCham and others.

	QCham	Cham	LSHADE	SaDE	LSPACMA	GWO	GA	TLBO	WOA
S1	0.0555	0.0669	0.0833	0.0917	0.1067	0.0679	0.1018	0.0925	0.0738
S2	0.0638	0.0680	0.1254	0.1114	0.1342	0.0809	0.1280	0.0941	0.0699
S3	0.0373	0.0613	0.0655	0.1063	0.0575	0.1075	0.1018	0.0902	0.0738
S4	0.0467	0.0842	0.2504	0.3853	0.2977	0.1415	0.1923	0.2588	0.1598
S5	0.2357	0.2809	0.2258	0.2285	0.2223	0.1998	0.3306	0.3158	0.2170
S6	0.1457	0.1617	0.2019	0.2417	0.2519	0.2038	0.1958	0.2478	0.2160
S7	0.0352	0.0594	0.1160	0.1148	0.1561	0.0817	0.1206	0.1573	0.0993
S8	0.0784	0.0908	0.3904	0.3658	0.3584	0.0955	0.1148	0.1132	0.0971
S9	0.0938	0.0929	0.2567	0.2516	0.2167	0.1564	0.1818	0.1889	0.1287
S10	0.0491	0.0666	0.2118	0.2706	0.3197	0.0998	0.1179	0.1998	0.1176
S11	0.0546	0.1471	0.3200	0.3500	0.2474	0.0489	0.2022	0.0418	0.0408
S12	0.0571	0.0886	0.2833	0.3333	0.3917	0.0965	0.0890	0.1471	0.0673
S13	0.1568	0.1661	0.1630	0.2417	0.1370	0.2353	0.2047	0.2027	0.2336
S14	0.2148	0.2462	0.2635	0.2992	0.3208	0.2548	0.2279	0.2659	0.2572
S15	0.0568	0.0909	0.0567	0.0850	0.1142	0.0533	0.1052	0.0881	0.0457
S16	0.2568	0.2843	0.3574	0.4094	0.4381	0.3026	0.3075	0.3137	0.2996
S17	0.0267	0.0477	0.1833	0.1583	0.1597	0.0571	0.0878	0.0956	0.0699
S18	0.0196	0.0234	0.3333	0.0833	0.2333	0.0660	0.0563	0.0515	0.0533

Table 4 shows the fitness values with the best results. We can observe from this table that the suggested QCham approach produced competitive results, when compared to the TLBO. The QCham had the best values in twenty datasets, while the Cham had the best results in four datasets, followed by WOA, which has the smallest value at three datasets. This indicates that the developed method still provides better fitness value than others.

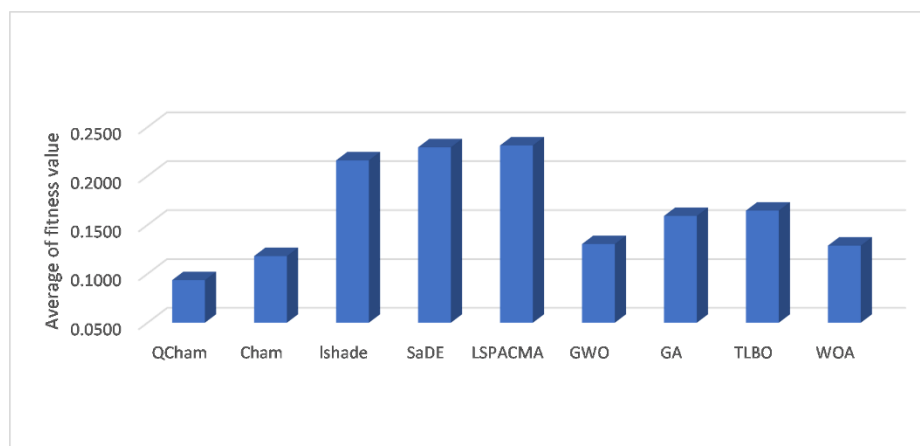


Figure 2. Fitness value mean of all the investigated datasets.

Table 4. Best fitness values for QCham and others.

	QCham	Cham	LSHADE	SaDE	LSPACMA	GWO	GA	TLBO	WOA
S1	0.0526	0.0590	0.0833	0.0917	0.0967	0.0573	0.0830	0.0830	0.0590
S2	0.0458	0.0470	0.1254	0.0947	0.1342	0.0607	0.1107	0.1107	0.0491
S3	0.0373	0.0476	0.0655	0.0897	0.0575	0.0664	0.0769	0.0769	0.0560
S4	0.0462	0.0462	0.2048	0.3853	0.2977	0.0462	0.0615	0.0615	0.0462
S5	0.2327	0.2558	0.2258	0.2118	0.2223	0.1967	0.3032	0.3032	0.2102
S6	0.1205	0.1462	0.2019	0.2352	0.2278	0.1628	0.1692	0.1692	0.1731
S7	0.0176	0.0362	0.1160	0.1099	0.1493	0.0674	0.1048	0.1048	0.0742
S8	0.0645	0.0781	0.3866	0.3658	0.3254	0.0683	0.1003	0.1003	0.0660
S9	0.0556	0.0556	0.2500	0.2448	0.2167	0.1065	0.1322	0.1322	0.0471
S10	0.0462	0.0462	0.2118	0.2557	0.3135	0.0538	0.0615	0.0615	0.0615
S11	0.0071	0.0714	0.3200	0.3333	0.1872	0.0203	0.1982	0.1982	0.0031
S12	0.0300	0.0233	0.2833	0.3333	0.3333	0.0648	0.0750	0.0750	0.0481
S13	0.1394	0.1121	0.1630	0.2278	0.1370	0.1939	0.1773	0.1773	0.2182
S14	0.2120	0.2243	0.2635	0.2974	0.3208	0.2307	0.2120	0.2120	0.2354
S15	0.0275	0.0700	0.0567	0.0683	0.0917	0.0338	0.0625	0.0625	0.0363
S16	0.2354	0.2562	0.3574	0.4094	0.4290	0.2847	0.2951	0.2951	0.2730
S17	0.0154	0.0308	0.1833	0.1444	0.1194	0.0385	0.0692	0.0692	0.0462
S18	0.0188	0.0125	0.3333	0.0667	0.2333	0.0438	0.0438	0.0438	0.0375

In terms of the worst fitness values, as shown in Table 5, the proposed QCham outperformed the other examined approaches, achieving the best results in 77% of all datasets and showing competitive results in the remaining datasets. The LSHADE, LSPACMA, GWO, and TLBO came in second, third, fourth, and fifth place, respectively.

Table 6 also lists the average of the relevant selected features. As seen in Table 6, the QCham has the fewest number of features, at nearly 62% from the tested datasets. The LSHADE, SaDE, and LSPACMA allocated the second rank, which has results better than others, according to the number of selected features. However, GA is the worst algorithm among the tested datasets. The same observation can be noticed from Figure 3.

Table 5. Worst fitness values for QCham and others.

	QCham	Cham	LSHADE	SaDE	LSPACMA	GWO	GA	TLBO	WOA
S1	0.0655	0.0719	0.0833	0.0917	0.1167	0.0818	0.1228	0.1023	0.0976
S2	0.0819	0.0895	0.1254	0.1281	0.1342	0.1011	0.1365	0.1065	0.0856
S3	0.0373	0.1080	0.0655	0.1230	0.0575	0.1431	0.1330	0.1017	0.1037
S4	0.0538	0.2524	0.2960	0.3853	0.2977	0.2372	0.3096	0.2845	0.2822
S5	0.2775	0.3000	0.2258	0.2452	0.2223	0.2121	0.3559	0.3469	0.3117
S6	0.1731	0.1859	0.2019	0.2481	0.2759	0.2692	0.2090	0.2795	0.2513
S7	0.0489	0.0860	0.1160	0.1197	0.1629	0.0996	0.1389	0.1894	0.1279
S8	0.0894	0.1130	0.3942	0.3658	0.3915	0.1215	0.1340	0.1399	0.1160
S9	0.1254	0.1322	0.2633	0.2583	0.2167	0.2052	0.2278	0.2600	0.2467
S10	0.0705	0.1007	0.2118	0.2855	0.3258	0.1785	0.2106	0.2582	0.1836
S11	0.1348	0.2252	0.3200	0.3667	0.3077	0.0905	0.2065	0.0449	0.0852
S12	0.0862	0.1157	0.2833	0.3333	0.4500	0.1243	0.1081	0.1705	0.0867
S13	0.1939	0.2167	0.1630	0.2556	0.1370	0.2652	0.2227	0.2273	0.2379
S14	0.2214	0.2847	0.2635	0.3010	0.3208	0.2924	0.2793	0.3040	0.2917
S15	0.0738	0.1138	0.0567	0.1017	0.1367	0.0975	0.1438	0.1075	0.0950
S16	0.2741	0.3094	0.3574	0.4094	0.4472	0.3215	0.3215	0.3348	0.3229
S17	0.0538	0.0692	0.1833	0.1722	0.2000	0.0712	0.1192	0.1192	0.0865
S18	0.0250	0.0375	0.3333	0.1000	0.2333	0.0866	0.0688	0.0866	0.0804

Table 6. Selected features numbers for all methods.

	QCham	Cham	LSHADE	SaDE	LSPACMA	GWO	GA	TLBO	WOA
S1	4	4	3	5	6	3	5	3	2
S2	5	9	5	9	6	8	21	7	6
S3	3	3	4	4	11	5	11	5	4
S4	6	8	7	4	10	7	9	6	9
S5	3	6	5	5	4	4	9	4	4
S6	3	6	7	2	3	6	11	6	5
S7	3	9	4	5	4	9	28	9	8
S8	9	21	11	15	14	21	29	15	19
S9	3	11	3	4	3	8	14	9	4
S10	3	8	7	5	4	9	9	6	9
S11	25	59	35	20	25	107	267	58	41
S12	13	30	16	19	20	24	50	29	31
S13	6	9	4	7	5	7	17	8	4
S14	3	5	4	4	7	5	6	6	5
S15	2	8	3	2	2	4	11	5	2
S16	5	21	7	15	9	20	34	19	22
S17	4	6	7	6	5	5	9	5	6
S18	7	4	5	5	4	8	9	3	8

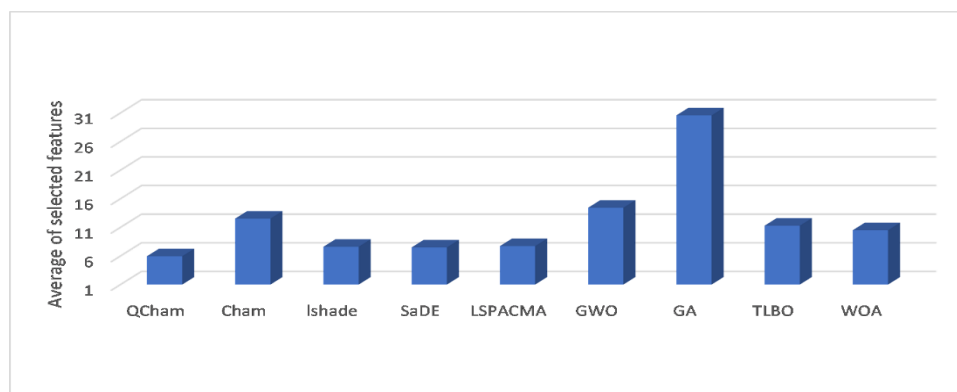


Figure 3. Average of selected features using QCham.

Table 7 summarizes the accuracy results of the developed QCham and compared approaches. The results of this test suggest that the proposed QCham is better than the alternatives. In 44% of all datasets, it had the best accuracy; in 11% of all datasets, it had the same accuracy as the other approaches. This result shows that the QCham can select the most relative feature, while maintaining the classification accuracy quality. TLBO came in second, followed by LSPACMA, GA, Cham, and LSHADE. In addition, Figure 4. shows the average of accuracy among the eighteen datasets, and it can be observed that QCham provides results better than others, followed by TLBO, GWO, and WOA, respectively.

Table 7. Accuracy results for all methods.

	QCham	Cham	LSHADE	SaDE	LSPACMA	GWO	GA	TLBO	WOA
S1	0.9857	0.9689	0.9286	0.9643	0.9429	0.9567	0.9462	0.9729	0.9476
S2	0.9688	0.9592	0.8684	0.9123	0.9035	0.9415	0.9351	0.9573	0.9433
S3	0.9540	0.9552	0.9540	0.9195	0.9655	0.9157	0.9609	0.9655	0.9448
S4	0.9743	0.9723	0.7375	0.6400	0.6700	0.8987	0.8667	1.0000	0.8960
S5	0.7358	0.6370	0.7250	0.7450	0.7300	0.7900	0.7130	0.7507	0.7703
S6	0.8944	0.6676	0.7593	0.7500	0.7593	0.8272	0.8753	0.8877	0.7988
S7	0.9834	0.8148	0.9296	0.9789	0.9437	0.9380	0.9577	0.9859	0.9174
S8	0.9627	0.6291	0.5852	0.5250	0.5594	0.9577	0.9616	0.9547	0.9507
S9	0.9666	0.6400	0.8000	0.8073	0.8333	0.8756	0.8844	0.9337	0.8821
S10	0.9935	0.5713	0.7450	0.7325	0.7100	0.9627	0.9477	0.9990	0.9497
S11	0.8567	1.0000	0.7333	0.6667	0.8846	0.9822	0.8667	0.9511	0.9686
S12	0.9714	0.7976	0.6905	0.6905	0.5357	0.9381	0.9937	0.9794	0.9825
S13	0.8833	0.7556	0.8148	0.7500	0.8519	0.7716	0.8580	0.8531	0.7593
S14	0.7938	0.5828	0.7188	0.7630	0.8750	0.7819	0.8250	0.8354	0.7809
S15	0.9825	0.8283	0.9667	0.9500	0.9083	0.9700	0.9600	0.9711	0.9622
S16	0.7732	0.5707	0.5370	0.5580	0.5170	0.7186	0.7533	0.7530	0.7283
S17	1.0000	0.8486	0.8333	0.9167	0.9861	0.9833	0.9833	1.0000	0.9759
S18	1.0000	0.8452	0.6667	1.0000	0.8333	0.9841	1.0000	1.0000	0.9968

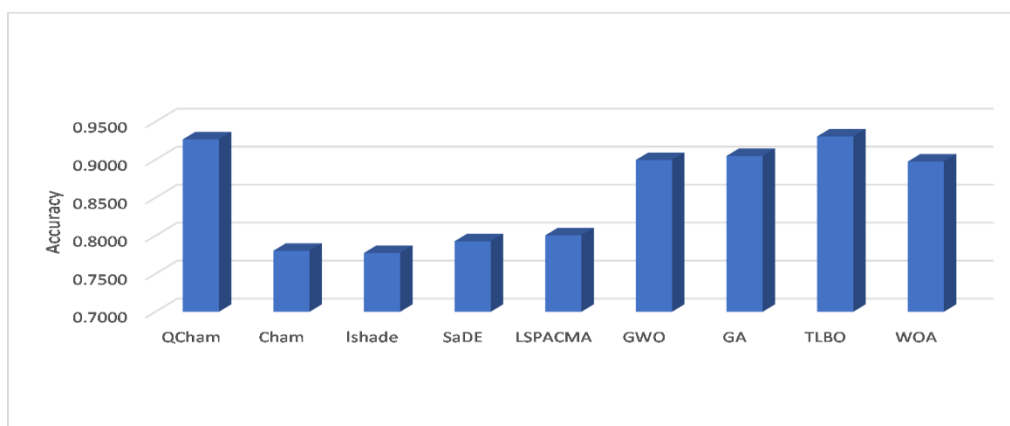


Figure 4. Average Accuracy of the models for all datasets studied.

Table 8 shows the average of CPU time(s) of the developed method and competitive algorithms. It can be observed from these results that the QCham takes small CPU time(s) among the tested datasets, in comparison to other FS methods. However, traditional Cham provides better results than other methods, at S5, S5, S12, S13, S15, and S17. In addition, GA has better CPU time(s) for only one dataset, namely S9.

Table 8. CPU(s) time of the QCham and other methods.

	QCham	Cham	LSHADE	SaDE	LSPACMA	GWO	GA	TLBO	WOA
S1	1.3266	2.5323	3.7597	3.9311	3.7896	4.8045	3.0372	7.1621	3.4947
S2	2.3220	4.4723	3.6052	3.6613	3.6389	3.7984	2.8619	7.1806	3.4250
S3	0.2932	0.4607	3.5721	3.6060	3.6004	3.6473	2.7438	6.4168	3.3492
S4	0.3335	0.5275	3.9773	4.0450	4.0228	4.1293	3.2438	7.9622	3.8586
S5	0.9228	0.4734	3.8583	3.9539	3.9313	4.0432	3.3300	7.2139	3.4962
S6	0.7098	0.4767	3.4766	3.4843	3.4874	3.5319	2.5871	6.7831	3.2491
S7	0.2854	0.4329	3.4925	3.4959	3.4998	3.6391	2.7085	6.7888	3.2704
S8	0.6804	1.2744	8.4271	8.5472	8.4831	14.9635	13.6991	24.1174	12.5058
S9	3.8243	4.4320	3.3085	3.3253	3.3642	3.4318	2.5320	6.5917	3.1419
S10	2.3248	3.5335	4.0457	4.1456	4.0962	4.1284	3.2549	7.8806	3.9806
S11	3.3222	2.4463	3.5204	3.5962	3.5438	6.6027	2.7109	6.5656	3.1659
S12	3.2944	3.0633	3.4390	3.4875	3.4694	3.8382	2.6443	6.4529	3.1784
S13	1.8448	1.4495	3.4457	3.5273	3.5137	3.5143	2.5964	6.7163	3.1486
S14	2.4521	3.5254	4.1032	4.1750	4.1446	4.0571	3.2515	7.9514	3.8750
S15	2.5025	2.4369	3.5085	3.5751	3.5403	3.5511	2.6326	6.7927	3.1668
S16	1.4819	2.4791	21.0411	21.3093	21.1920	30.4215	34.6083	47.0547	28.8056
S17	3.2765	2.4687	4.6537	4.6825	4.6444	3.4650	2.6549	6.4435	3.2474
S18	0.2940	0.4447	0.0196	0.0304	0.9959	0.4348	0.6307	0.4831	0.2730

5.3. Comparison with Other FS Models

The findings of the created QCham are compared to commonly used FS models that rely on the MH techniques in this section. The enhanced GWO (EGWO) [63], binary bat algorithm (BBA) [64], BGOA [65], PSO, two binary GWO algorithms called bGWO1 and bGWO2 [47], AGWO [44], biogeography-based optimization (BBO) [66], enhanced crow search algorithm (ECSA) [67], and satin bird optimizer (SBO) [63] are among the FS techniques. The results of the created QCham, and other approaches' classification accuracies

are shown in Table 9. These results show that the created QCham has a high ability to enhance classification accuracy across all datasets examined.

Table 9. Comparison with FS models.

Datasets	QCham	WOAT	bGWO2	BBO	ECSA	WOAR	PSO	AGWO	BBA	EGWO	BGOA	SBO
S1	0.9857	0.959	0.975	0.962	0.972	0.957	0.967	0.960	0.937	0.961	0.969	0.967
S2	0.9688	0.949	0.935	0.945	0.958	0.950	0.933	0.934	0.931	0.947	0.96	0.942
S3	0.9540	0.914	0.776	0.936	0.966	0.910	0.688	0.935	0.872	0.943	0.953	0.950
S4	0.9743	0.739	0.75	0.754	1	0.763	0.73	0.757	0.61	0.753	0.946	0.734
S5	0.7358	0.699	0.776	0.692	0.767	0.690	0.787	0.695	0.628	0.698	0.76	0.709
S6	0.8944	0.765	0.7	0.782	0.83	0.763	0.744	0.797	0.754	0.761	0.826	0.792
S7	0.9834	0.884	0.963	0.880	0.931	0.880	0.921	0.893	0.877	0.863	0.883	0.898
S9	0.9627	0.896	0.584	0.80	0.865	0.901	0.584	0.791	0.701	0.766	0.815	0.818
S10	0.9666	0.778	0.729	0.880	1	0.759	0.737	0.878	0.722	0.870	0.979	0.863
S11	0.9935	0.838	0.822	0.816	0.921	0.860	0.822	0.854	0.795	0.756	0.861	0.843
S12	0.8567	0.736	0.938	0.871	0.926	0.712	0.928	0.882	0.844	0.861	0.895	0.894
S13	0.9714	0.861	0.834	0.798	0.847	0.857	0.819	0.813	0.8	0.804	0.803	0.798
S14	0.8833	0.792	0.727	0.768	0.842	0.778	0.735	0.762	0.665	0.771	0.951	0.768
S15	0.7938	0.736	0.92	0.917	0.96	0.739	0.904	0.92	0.851	0.902	0.729	0.934
S17	0.9825	0.935	0.92	0.966	0.985	0.932	0.933	0.957	0.919	0.966	0.979	0.968
S18	0.7732	0.710	0.879	0.937	0.983	0.712	0.861	0.968	0.874	0.968	0.99	0.968

In conclusion, it can be noticed there is no one algorithm can provide better performance among all the tested datasets, and this is concerned with the no free lunch theorem. Whereas the earlier findings imply that applying the proposed QCham approach considerably enhances the capacity to tackle feature selection challenges. The Cham is significantly enhanced when the QBO operators are used in the structure. As a result, the QCham can be regarded as an effective and efficient optimization technique for tackling the feature selection problem, since it has ability to discover the feasible regions that contain feasible solutions, as observed from the quality of the selected features that influence of the classification accuracy.

6. Conclusions and Future Work

This research presented a novel variant for the chameleon swarm algorithm (Cham) by including the operator of quantum-based optimization (QBO) into the Cham's exploration phase to provide an efficient feature selection (FS) optimizer. When treating large-dimensional optimization issues, the Cham has a limitation, for which a hybrid variant called QCham was proposed. To complete the FS optimization objective, the suggested QCham was applied to eighteen different real-world datasets. QCham was compared to the basic Cham, LSHADE, SaDE, LSPACMA, GWO, GA, TLBO, and WOA. In 8 out of 18 datasets, the QCham obtained the best fitness, in 4 out of 18, with the lowermost Std value; in 50% of the datasets, it obtained the lowest number of features, according to the comparisons. This result showed that the QCham has the ability to choose the relevant feature and preserve the classification quality. The proposed QCham obtained the maximum accuracy in the average of all datasets. Additionally, the QBO operators are crucial in enhancing the exploration phase of the original Cham model.

The developed QCham will be tested in subsequent work, with a number of applications, including time-series forecasting, parameter estimation, and picture segmentation.

Author Contributions: M.A.E. and A.H.E.: conceptualization, supervision, methodology, formal analysis, resources, data curation, and writing—original draft preparation. M.A.E. and A.H.E.: conceptualization, supervision, methodology, formal analysis, resources, data curation, and writing—original draft preparation. M.A.E. and A.H.E.: writing—review. M.A., S.A., N.A., and A.F.: writing—review and editing. M.A.E. and A.H.E.: writing—review, editing, and editing, project administration, formal analysis, and funding acquisition. M.A.E. and A.H.E.: supervision, resources, formal analysis, methodology, and writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deputyship for Research and Innovation, Ministry of Education in Saudi Arabia, for funding this research work, through project number IFP-IMSIU202209.

Data Availability Statement: The data are collected from <https://archive.ics.uci.edu/ml/index.php>, accessed on 18 August 2022.

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

References

- Zhao, W.; Wang, C.; Nakahira, Y. Medical Application on Internet of Things. In Proceedings of the IET International Conference on Communication Technology and Application (ICCTA 2011), Beijing, China, 14–16 October 2011; pp. 660–665.
- Javaid, M.; Khan, I.H. Internet of Things (IoT) enabled healthcare helps to take the challenges of COVID-19 Pandemic. *J. Oral Biol. Craniofacial Res.* **2021**, *11*, 209–214. [[CrossRef](#)] [[PubMed](#)]
- Abd Elaziz, M.; Mabrouk, A.; Dahou, A.; Chelloug, S.A. Medical Image Classification Utilizing Ensemble Learning and Levy Flight-Based Honey Badger Algorithm on 6G-Enabled Internet of Things. *Comput. Intell. Neurosci.* **2022**, *2022*, 5830766. [[CrossRef](#)] [[PubMed](#)]
- Boursianis, A.D.; Papadopoulou, M.S.; Diamantoulakis, P.; Liopa-Tsakalidi, A.; Barouchas, P.; Salahas, G.; Karagiannidis, G.; Wan, S.; Goudos, S.K. Internet of Things (IoT) and Agricultural Unmanned Aerial Vehicles (UAVs) in smart farming: A comprehensive review. *Internet Things* **2022**, *18*, 100187. [[CrossRef](#)]
- Salih, K.O.M.; Rashid, T.A.; Radovanovic, D.; Bacanin, N. A Comprehensive Survey on the Internet of Things with the Industrial Marketplace. *Sensors* **2022**, *22*, 730. [[CrossRef](#)]
- Haghnegahdar, L.; Joshi, S.S.; Dahotre, N.B. From IoT-based cloud manufacturing approach to intelligent additive manufacturing: Industrial Internet of Things—an overview. *Int. J. Adv. Manuf. Technol.* **2022**, *119*, 1461–1478. [[CrossRef](#)]
- Abichandani, P.; Sivakumar, V.; Lobo, D.; Iaboni, C.; Shekhar, P. Internet-of-Things Curriculum, Pedagogy, and Assessment for STEM Education: A Review of Literature. *IEEE Access* **2022**, *10*, 38351–38369. [[CrossRef](#)]
- Tubishat, M.; Idris, N.; Shuib, L.; Abushariah, M.A.M.; Mirjalili, S. Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection. *Expert Syst. Appl.* **2020**, *145*, 113122. [[CrossRef](#)]
- Hancer, E.; Xue, B.; Karaboga, D.; Zhang, M. A binary ABC algorithm based on advanced similarity scheme for feature selection. *Appl. Soft Comput. J.* **2015**, *36*, 334–348. [[CrossRef](#)]
- Ewees, A.A.; Abualigah, L.; Yousri, D.; Algamal, Z.Y.; Al-qaness, M.A.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](#)]
- 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](#)]
- Al-qaness, M.A.A. Device-free human micro-activity recognition method using WiFi signals. *Geo Spat. Inf. Sci.* **2019**, *22*, 128–137. [[CrossRef](#)]
- Sahlol, A.T.; Yousri, D.; Ewees, A.A.; Al-qaness, M.A.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](#)]
- Benazzouz, A.; Guilal, R.; Amirouche, F.; Hadj Slimane, Z.E. EMG Feature Selection for Diagnosis of Neuromuscular Disorders. In Proceedings of the 2019 International Conference on Networking and Advanced Systems (ICNAS), Annaba, Algeria, 26–27 June 2019; pp. 1–5.
- 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), Brisbane, Australia, 10–15 June 2018; pp. 1–8.
- 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](#)]
- 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](#)]

18. Ibrahim, R.A.; Ewees, A.A.; Oliva, D.; Elaziz, M.A.; Lu, S. Improved salp swarm algorithm based on particle swarm optimization for feature selection. *J. Ambient Intell. Humaniz. Comput.* **2018**, *10*, 1–15. [[CrossRef](#)]
19. Ma, J.; Bi, Z.; Ting, T.O.; Hao, S.; Hao, W. Comparative performance on photovoltaic model parameter identification via bio-inspired algorithms. *Sol. Energy* **2016**, *132*, 606–616. [[CrossRef](#)]
20. Chang, F.J.; Chang, Y.T. Adaptive neuro-fuzzy inference system for prediction of water level in reservoir. *Adv. Water Resour.* **2006**, *29*, 1–10. [[CrossRef](#)]
21. Khanduzi, R.; Maleki, H.R.; Akbari, R. Two novel combined approaches based on TLBO and PSO for a partial interdiction/fortification problem using capacitated facilities and budget constraint. *Soft Comput.* **2018**, *22*, 5901–5919. [[CrossRef](#)]
22. Elsheikh, A.H.; Muthuramalingam, T.; Abd Elaziz, M.; Ibrahim, A.M.M.; Showaib, E.A. Minimization of fume emissions in laser cutting of polyvinyl chloride sheets using genetic algorithm. *Int. J. Environ. Sci. Technol.* **2022**, *19*, 6331–6344. [[CrossRef](#)]
23. Babikir, H.A.; Elaziz, M.A.; Elsheikh, A.H.; Showaib, E.A.; Elhadary, M.; Wu, D.; Liu, Y. Noise prediction of axial piston pump based on different valve materials using a modified artificial neural network model. *Alexandria Eng. J.* **2019**, *58*, 1077–1087. [[CrossRef](#)]
24. Elmaadawy, K.; Elaziz, M.A.; Elsheikh, A.H.; Moawad, A.; Liu, B.; Lu, S. Utilization of random vector functional link integrated with manta ray foraging optimization for effluent prediction of wastewater treatment plant. *J. Environ. Manag.* **2021**, *298*, 113520. [[CrossRef](#)] [[PubMed](#)]
25. Khoshaim, A.B.; Moustafa, E.B.; Bafakeeh, O.T.; Elsheikh, A.H. An Optimized Multilayer Perceptrons Model Using Grey Wolf Optimizer to Predict Mechanical and Microstructural Properties of Friction Stir Processed Aluminum Alloy Reinforced by Nanoparticles. *Coatings* **2021**, *11*, 1476. [[CrossRef](#)]
26. Abd Elaziz, M.; Elsheikh, A.H.; Oliva, D.; Abualigah, L.; Lu, S.; Ewees, A.A. Advanced Metaheuristic Techniques for Mechanical Design Problems: Review. *Arch. Comput. Methods Eng.* **2022**, *29*, 695–716. [[CrossRef](#)]
27. Shehabeldeen, T.A.; Elaziz, M.A.; Elsheikh, A.H.; Zhou, J. Modeling of friction stir welding process using adaptive neuro-fuzzy inference system integrated with harris hawks optimizer. *J. Mater. Res. Technol.* **2019**, *8*, 5882–5892. [[CrossRef](#)]
28. Ekinici, S.; Hekimoğlu, B.; Izci, D. Opposition based Henry gas solubility optimization as a novel algorithm for PID control of DC motor. *Eng. Sci. Technol. Int. J.* **2021**, *24*, 331–342. [[CrossRef](#)]
29. Shehabeldeen, T.A.; Elaziz, M.A.; Elsheikh, A.H.; Hassan, O.F.; Yin, Y.; Ji, X.; Shen, X.; Zhou, J. A Novel Method for Predicting Tensile Strength of Friction Stir Welded AA6061 Aluminium Alloy Joints based on Hybrid Random Vector Functional Link and Henry Gas Solubility Optimization. *IEEE Access* **2020**, *30*, 188–193. [[CrossRef](#)]
30. Askari, Q.; Younas, I.; Saeed, M. Political Optimizer: A novel socio-inspired meta-heuristic for global optimization. *Knowl. Based Syst.* **2020**, *195*, 105709. [[CrossRef](#)]
31. Too, J.; Abdullah, A.R.; Saad, N.M. Mohd Saad A New Quadratic Binary Harris Hawk Optimization for Feature Selection. *Electronics* **2019**, *8*, 1130. [[CrossRef](#)]
32. Abd Elaziz, M.; Dahou, A.; Alsaleh, N.A.; Elsheikh, A.H.; Saba, A.I.; Ahmadein, M. Boosting COVID-19 Image Classification Using MobileNetV3 and Aquila Optimizer Algorithm. *Entropy* **2021**, *23*, 1383. [[CrossRef](#)]
33. Askarzadeh, A. A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Comput. Struct.* **2016**, *169*, 1–12. [[CrossRef](#)]
34. Songyang, L.; Haipeng, Y.; Miao, W. Cat swarm optimization algorithm based on the information interaction of subgroup and the top-N learning strategy. *J. Intell. Syst.* **2022**, *31*, 489–500. [[CrossRef](#)]
35. Zhao, W.; Wang, L.; Zhang, Z. Artificial ecosystem-based optimization: A novel nature-inspired meta-heuristic algorithm. *Neural Comput. Appl.* **2020**, *32*, 9383–9425. [[CrossRef](#)]
36. Said, M.; El-Rifaie, A.M.; Tolba, M.A.; Houssein, E.H.; Deb, S. An Efficient Chameleon Swarm Algorithm for Economic Load Dispatch Problem. *Mathematics* **2021**, *9*, 2770. [[CrossRef](#)]
37. Izci, D.; Ekinici, S.; Kayri, M.; Eker, E. A novel improved arithmetic optimization algorithm for optimal design of PID controlled and Bode's ideal transfer function based automobile cruise control system. *Evol. Syst.* **2022**, *13*, 453–468. [[CrossRef](#)]
38. Rizk-Allah, R.M.; El-Hameed, M.A.; El-Fergany, A.A. Model parameters extraction of solid oxide fuel cells based on semi-empirical and memory-based chameleon swarm algorithm. *Int. J. Energy Res.* **2021**, *45*, 21435–21450. [[CrossRef](#)]
39. Mostafa, R.R.; Ewees, A.A.; Ghoniem, R.M.; Abualigah, L.; Hashim, F.A. Boosting chameleon swarm algorithm with consumption AEO operator for global optimization and feature selection. *Knowl. Based Syst.* **2022**, *246*, 108743. [[CrossRef](#)]
40. Umamageswari, A.; Bharathiraja, N.; Irene, D.S. A Novel Fuzzy C-Means based Chameleon Swarm Algorithm for Segmentation and Progressive Neural Architecture Search for Plant Disease Classification. *ICT Express* **2021**, *in press*. [[CrossRef](#)]
41. Al-Othman, A.K.; Ahmed, N.A.; Al-Fares, F.S.; AlSharidah, M.E. Parameter Identification of PEM Fuel Cell Using Quantum-Based Optimization Method. *Arab. J. Sci. Eng.* **2015**, *40*, 2619–2628. [[CrossRef](#)]
42. Agrawal, R.K.; Kaur, B.; Sharma, S. Quantum based Whale Optimization Algorithm for wrapper feature selection. *Appl. Soft Comput.* **2020**, *89*, 106092. [[CrossRef](#)]
43. Ho, S.L.; Yang, S.; Ni, G.; Huang, J. A Quantum-Based Particle Swarm Optimization Algorithm Applied to Inverse Problems. *IEEE Trans. Magn.* **2013**, *49*, 2069–2072. [[CrossRef](#)]
44. Chaudhuri, A.; Sahu, T.P. Feature selection using Binary Crow Search Algorithm with time varying flight length. *Expert Syst. Appl.* **2021**, *168*, 114288. [[CrossRef](#)]

45. Sadeghian, Z.; Akbari, E.; Nematzadeh, H. A hybrid feature selection method based on information theory and binary butterfly optimization algorithm. *Eng. Appl. Artif. Intell.* **2021**, *97*, 104079. [[CrossRef](#)]
46. Maleki, N.; Zeinali, Y.; Niaki, S.T.A. A k-NN method for lung cancer prognosis with the use of a genetic algorithm for feature selection. *Expert Syst. Appl.* **2021**, *164*, 113981. [[CrossRef](#)]
47. Song, X.; Zhang, Y.; Gong, D.; Sun, X. Feature selection using bare-bones particle swarm optimization with mutual information. *Pattern Recognit.* **2021**, *112*, 107804. [[CrossRef](#)]
48. Sathiyabhama, B.; Kumar, S.U.; Jayanthi, J.; Sathiya, T.; Ilavarasi, A.K.; Yuvarajan, V.; Gopikrishna, K. A novel feature selection framework based on grey wolf optimizer for mammogram image analysis. *Neural Comput. Appl.* **2021**, *33*, 14583–14602. [[CrossRef](#)]
49. Aljarah, I.; Habib, M.; Faris, H.; Al-Madi, N.; Heidari, A.A.; Mafarja, M.; Elaziz, M.A.; Mirjalili, S. A dynamic locality multi-objective salp swarm algorithm for feature selection. *Comput. Ind. Eng.* **2020**, *147*, 106628. [[CrossRef](#)]
50. Dhiman, G.; Oliva, D.; Kaur, A.; Singh, K.K.; Vimal, S.; Sharma, A.; Cengiz, K. BEPO: A novel binary emperor penguin optimizer for automatic feature selection. *Knowl. Based Syst.* **2021**, *211*, 106560. [[CrossRef](#)]
51. Amini, F.; Hu, G. A two-layer feature selection method using Genetic Algorithm and Elastic Net. *Expert Syst. Appl.* **2021**, *166*, 114072. [[CrossRef](#)]
52. Neggaz, N.; Houssein, E.H.; Hussain, K. An efficient henry gas solubility optimization for feature selection. *Expert Syst. Appl.* **2020**, *152*, 113364. [[CrossRef](#)]
53. Rostami, M.; Berahmand, K.; Nasiri, E.; Forouzandeh, S. Review of swarm intelligence-based feature selection methods. *Eng. Appl. Artif. Intell.* **2021**, *100*, 104210. [[CrossRef](#)]
54. Agrawal, P.; Abutarboush, H.F.; Ganesh, T.; Mohamed, A.W. Metaheuristic Algorithms on Feature Selection: A Survey of One Decade of Research (2009–2019). *IEEE Access* **2021**, *9*, 26766–26791. [[CrossRef](#)]
55. Braik, M.S. Chameleon Swarm Algorithm: A bio-inspired optimizer for solving engineering design problems. *Expert Syst. Appl.* **2021**, *174*, 114685. [[CrossRef](#)]
56. Srikanth, K.; Panwar, L.K.; Panigrahi, B.; Herrera-Viedma, E.; Sangaiah, A.K.; Wang, G.-G. Meta-heuristic framework: Quantum inspired binary grey wolf optimizer for unit commitment problem. *Comput. Electr. Eng.* **2018**, *70*, 243–260. [[CrossRef](#)]
57. Tanabe, R.; Fukunaga, A.S. Improving the Search Performance of SHADE Using Linear Population Size Reduction. In Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 6–11 July 2014; pp. 1658–1665.
58. Qin, A.K.; Suganthan, P.N. Self-Adaptive Differential Evolution Algorithm for Numerical Optimization. In Proceedings of the 2005 IEEE Congress on Evolutionary Computation, Edinburgh, UK, 2–4 September 2005; Volume 2, pp. 1785–1791.
59. Gill, H.S.; Khehra, B.S.; Singh, A.; Kaur, L. Teaching-learning based optimization algorithm to minimize cross entropy for selecting multilevel threshold values. *Egypt. Inform. J.* **2018**, *20*, 11–25. [[CrossRef](#)]
60. 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 Sebastian, Spain, 5–8 June 2017; pp. 145–152.
61. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
62. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
63. Arora, S.; Singh, H.; Sharma, M.; Sharma, S.; Anand, P. A New Hybrid Algorithm Based on Grey Wolf Optimization and Crow Search Algorithm for Unconstrained Function Optimization and Feature Selection. *IEEE Access* **2019**, *7*, 26343–26361. [[CrossRef](#)]
64. Nakamura, R.Y.M.; Pereira, L.A.M.; Costa, K.A.; Rodrigues, D.; Papa, J.P.; Yang, X.-S. BBA: A Binary Bat Algorithm for Feature Selection. In Proceedings of the 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images, Washington, DC, USA, 22–25 August 2012; pp. 291–297.
65. Mafarja, M.; Aljarah, I.; Heidari, A.A.; Hammouri, A.I.; Faris, H.; Al-Zoubi, A.M.; Mirjalili, S. Evolutionary Population Dynamics and Grasshopper Optimization approaches for feature selection problems. *Knowl. Based Syst.* **2018**, *145*, 25–45. [[CrossRef](#)]
66. Saremi, S.; Mirjalili, S.; Lewis, A. Biogeography-based optimisation with chaos. *Neural Comput. Appl.* **2014**, *25*, 1077–1097. [[CrossRef](#)]
67. Ouadfel, S.; Abd Elaziz, M. Enhanced Crow Search Algorithm for Feature Selection. *Expert Syst. Appl.* **2020**, *159*, 113572. [[CrossRef](#)]