

Article

# A Cox Proportional-Hazards Model Based on an Improved Aquila Optimizer with Whale Optimization Algorithm Operators

Ahmed A. Ewees <sup>1,2,\*</sup>, Zakariya Yahya Algamal <sup>3</sup>, Laith Abualigah <sup>4,5</sup>, Mohammed A. A. Al-qaness <sup>6</sup>,  
Dalia Yousri <sup>7</sup>, Rania M. Ghoniem <sup>8</sup> and Mohamed Abd Elaziz <sup>9,10,11</sup>

<sup>1</sup> Department of e-Systems, University of Bisha, Bisha 61922, Saudi Arabia

<sup>2</sup> Department of Computer, Damietta University, Damietta 34517, Egypt

<sup>3</sup> Department of Statistics and Informatics, University of Mosul, Mosul 41002, Iraq;  
zakariya.algamal@uomosul.edu.iq

<sup>4</sup> Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan;  
laythdyabat@aaau.edu.jo

<sup>5</sup> School of Computer Sciences, Universiti Sains Malaysia, Gelugor 11800, Pulau Pinang, Malaysia

<sup>6</sup> State Key Laboratory for Information Engineering in Surveying, Mapping and Remote Sensing,  
Wuhan University, Wuhan 430079, China; alqaness@whu.edu.cn

<sup>7</sup> Department of Electrical Engineering, Faculty of Engineering, Fayoum University, Fayoum 63514, Egypt;  
day01@fayoum.edu.eg

<sup>8</sup> Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint  
Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia; rmghoniem@pnu.edu.sa

<sup>9</sup> Faculty of Computer Science & Engineering, Galala University, Suze 435611, Egypt;  
abd\_el\_aziz\_m@yahoo.com

<sup>10</sup> Artificial Intelligence Research Center (AIRC), Ajman University, Ajman 346, United Arab Emirates

<sup>11</sup> Department of Mathematics, Faculty of Science, Zagazig University, Zagazig 44519, Egypt

\* Correspondence: ewees@du.edu.eg



**Citation:** Ewees, A.A.; Algamal, Z.Y.;

Abualigah, L.; Al-qaness, M.A.A.;

Yousri, D.; Ghoniem, R.M.; Abd

Elaziz, M. A Cox

Proportional-Hazards Model Based

on an Improved Aquila Optimizer

with Whale Optimization

Algorithm Operators. *Mathematics*

**2022**, *10*, 1273. [https://doi.org/](https://doi.org/10.3390/math10081273)

10.3390/math10081273

Academic Editor: Ioannis E. Livieris

Received: 4 March 2022

Accepted: 6 April 2022

Published: 12 April 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/>).

**Abstract:** Recently, a new optimizer, called the Aquila Optimizer (AO), was developed to solve different optimization problems. Although the AO has a significant performance in various problems, like other optimization algorithms, the AO suffers from certain limitations in its search mechanism, such as local optima stagnation and convergence speed. This is a general problem that faces almost all optimization problems, which can be solved by enhancing the search process of an optimizer using an assistant search tool, such as using hybridizing with another optimizer or applying other search techniques to boost the search capability of an optimizer. Following this concept to address this critical problem, in this paper, we present an alternative version of the AO to alleviate the shortcomings of the traditional one. The main idea of the improved AO (IAO) is to use the search strategy of the Whale Optimization Algorithm (WOA) to boost the search process of the AO. Thus, the IAO benefits from the advantages of the AO and WOA, and it avoids the limitations of the local search as well as losing solutions diversity through the search process. Moreover, we apply the developed IAO optimization algorithm as a feature selection technique using different benchmark functions. More so, it is tested with extensive experimental comparisons to the traditional AO and WOA algorithms, as well as several well-known optimizers used as feature selection techniques, like the particle swarm optimization (PSO), differential evolution (DE), mouth flame optimizer (MFO), firefly algorithm, and genetic algorithm (GA). The outcomes confirmed that the using of the WOA operators has a significant impact on the AO performance. Thus the combined IAO obtained better results compared to other optimizers.

**Keywords:** feature selection; the Cox proportional-hazards model; Whale Optimization Algorithm; the Aquila Optimizer

**MSC:** 68T01

## 1. Introduction

Data mining is an unavoidable stage in extracting knowledge and data gained from data mining, employed in various sectors, including industrial and medical applications [1]. Recently, there has been a growth in the number of gathered and retained features in databases, though not all of them are valuable for data analysis; therefore, some of them are utterly useless or unnecessary. These traits, therefore, have no utility in the information extraction, but they mainly enhance the complexity and incompleteness of the outcomes. As a result, feature selection aids in reducing the dimensionality of data prior to data processing [2]. There are  $n$  features in a vast database with numerous features to manage. The computation cost to assess all the features is exponential ( $O(2^n)$ ), making it essentially unattainable. As a result, feature selection techniques serve as the foundation for data mining, allowing beneficial characteristics to be retained for further learning tasks while discarding the most irrelevant and less significant ones. In reality, feature selection approaches disregard unimportant features, allowing the learning process to be more successful [3]. It has also been demonstrated that feature selection improves the classification performance of data mining algorithms such as the KNN classifier.

Mainly the three techniques to feature selection are the filter methods, embedded methods, and wrapper approaches [4]. For filter methods, the selected features can be filtered depending on the general properties of the used datasets (i.e., known metrics, for example, correlation). Those methods can be implemented without predictive models. Filter methods are fast, but they face some problems in case of avoiding overfitting, and they may fail in selecting the best features. In contrast, the wrapper methods exist as wrappers around the predictive models, and they employ the predictive models to select the best features. The main drawbacks of those methods are their expensive computation, but they produce better performance. In the case of embedded, the process of selecting features is embedded in the learning model. Embedded models are computationally expensive than wrapper methods, but they can be considered better from the aspects of overfitting.

The Catfish BPSO presented in this work is a wrapper approach. A vast number of chosen features for many pattern classification issues does not necessarily result in an excellent accuracy rate. In some instances, the effectiveness of algorithms solely dedicated to data classification speed and predictive value can reduce even though features may be unimportant or confusing, or are due to positive correlations. During the learning stage, these characteristics might have a detrimental influence on the categorization process. Ideally, the feature selection approach decreases the cost of feature assessment while increasing classifier performance and quality. Several techniques have traditionally been used to choose features from training and testing data, including Arithmetic Optimization Algorithm [5], Binary Butterfly Optimization [6], Aquila Optimizer [7], binary Gradient-based Optimizer [8], Firefly Algorithm [9], Atomic Orbit Search [10], RUN Kutta optimizer (RUN) [11], Colony Predation Algorithm (CPA) [12], Slime Mould Algorithm (SMA) [13], Harris Hawk Optimization (HHO) [14], Hunger Games Search (HGS) [15], and others.

A feature selection approach by ant colony optimization is given in [16]. The approach uses numerous rounds to select the best feature subset without utilizing any learning techniques. Furthermore, the feature importance will be estimated using the correlation among features, resulting in reducing repetition. The experimental findings on numerous commonly used datasets demonstrate the proposed method's efficiency and enhancements over earlier comparable approaches. This paper provides a new machine learning technique for high-dimensional data [17], which uses the Henry gas solubility optimization (HGSO) method to pick key features and enhance classification performance. The suggested technique is assessed against well-established optimization algorithms using multiple datasets with a broad feature size range, from tiny to large. Finally, the empirical research indicates that the suggested method is significantly successful on low or high-dimensional data.

A new mixed ant colony optimization approach is presented in [18] for feature selection utilizing a learning algorithm in this study. Choosing a subset of conspicuous characteristics of decreased size is an essential part of this approach. The suggested method

employs a hybrid search methodology that combines the benefits of the filter and wrapper techniques. The specifics of the comparison demonstrate that the presented process has a surprising capacity to construct reduction subsets of prominent features while still giving high classification performance.

In [19], a new feature selection technique using a mathematical framework of grasshopper interaction in discovering nutrition is suggested. The grasshopper optimization technique was modified to make it acceptable for a feature selection challenge. The proposed strategy is augmented by statistical measures to remove redundant features with the most interesting features during repetitions. Comparative trials show that the suggested approach is more effective than existing classification techniques. This work attempted to increase the effect of text classification using the particle swarm optimizer [20]. Many exploratory search strategies are conducted in this study by examining current accomplishments of enhanced particle swarm optimizers and characteristics of traditional feature selection methods. The basic model is chosen first, followed by two upgraded models based on the structural inertia weight and steady restriction factor to optimize feature selection approaches. The trial findings and significance tests reveal that the dynamically upgraded model outperforms all others in text classification effectiveness and dimension reliability.

The work in [21] suggests a non-negative inter feature selection method with variable graph restrictions to overcome the feature selection problem. In the presented model, linear regression is used to design the original data environment into a low-dimensional space to create the label matrix. The results demonstrate the efficiency of the suggested strategy on ten real datasets compared to other comparative approaches. A novel binary version of the grasshopper optimizer is presented and employed in [22] for the feature subset selection challenge in this research. This suggested novel binary grasshopper optimizer is evaluated and analyzed to five optimization algorithms employed in the feature selection issue. These techniques have been developed and tested on different data sets of varying sizes. The findings showed that the suggested strategy outperformed the other approaches examined.

The paper [23] provides a novel feature-selection search strategy for feature selection-based intrusion detection systems (IDS) using the cuttlefish optimization algorithm. Because IDS deal with a vast quantity of data, one of their most important duties is maintaining the highest quality of features that reflect the real data set while removing duplicate and unnecessary characteristics. Compared to the results produced utilizing all features, the feature subset derived via the proposed method provides a greater increasing security and correctness rate with a reduced probability of detection.

Unfortunately, Several issues are not addressed in the research mentioned above [24]. To begin, all features are chosen at random with the same chance. As a result, the principal features cannot be quickly taken for inclusion in the newly generated feature subset. Moreover, the traditional feature selection techniques cannot adequately select the most informative features. Thus, the improved optimization methods are too near to determine the best relative features through an efficient search process. These methods significantly reduce the efficiency of searching for the ideal feature subset.

#### *Motivation and Contribution*

To some extent, population-based optimization algorithms can prevent local optima stagnation. It also has a high capacity to converge to the optima. One of the primary motivations for this research is that there is no suitable optimizer for addressing all kinds of problems, as given in No-Free-Lunch; hence the excellent version of any optimization method on a set of problems does not guarantee a compelling performance on another problems. To the aim to contribute, no one has yet used Aquila Optimizer using the leading search operators of the Whale Optimization Algorithm to tackle feature selection in a systematic manner. The authors selected the Whale Optimization Algorithm because of its proven efficiency and superiority compared with numerous algorithms such as PSO, GA, GWO, etc., in several optimization problems in different fields. Moreover, the logarithmic spiral function of WOA is an attractive operator to enhance the AO phases to cover a major

area in uncertain search space. This was the primary motivator for us to select the Aquila Optimizer as the core of our work. This paper’s overarching focus is on providing new binary variants of the Aquila Optimizer, called IAO, for wrapper feature selection. The IAO enhanced the original search strategies of the Aquila Optimizer by using the main operators of the Whale Optimization Algorithm. This modification enables the IAO to tackle the main weaknesses of using a single search method by avoiding the local search problem and losing the diversity of the solutions in the search stage. The suggested technique identifies the best feature subset, which reduces feature subset size while increasing classification performance. The proposed IAO is evaluated on benchmark problems in terms of fitness values, the selected features number, and classification accuracy. The obtained results showed that the IAO got promising outcomes compared to different feature selection methods. Moreover, the IOA searchability is clearly observed in determining the best relative subset of features.

The following parts of the paper are arranged as follows, Section 2 the Aquila Optimizer and Whale Optimization Algorithm are described. Section 3 introduces the proposed algorithm for feature selection. Section 4 shows experiments, results, and discussions. Section 5, shows the conclusion and future work.

## 2. Background

### 2.1. The Aquila Optimizer (AO)

This subsection contains the basic formulation of the Aquila Optimizer (AO) [7]. To catch its target, the AO algorithm often imitates Aquila’s social behaviour. The AO has been adopted to solve various problems, such as time series forecasting [25], improving intrusion detection system (IDS) [26], task scheduling [27], global optimization [28,29], and others [30].

AO is a population-based optimization technique. It involves the formation of  $X$  with  $N$  agents as in Equation (1).

$$X_{ij} = r_1 \times (UB_j - LB_j) + LB_j, \quad j = [1, 2, 3, \dots, Dim], \quad i = [1, 2, 3, \dots, N] \quad (1)$$

where  $UB_j$  and  $LB_j$  denotes search space.  $r_1 \in [0, 1]$ .  $Dim$  refers to the solution’s dimension.

The next phase in the AO technique is to either explore or exploit until the best solution is identified. According to [7], there are two methods for exploitation and exploration. In the exploration, the best agent  $X_b$  and the  $X_M$  are used as:

$$X_i(t + 1) = X_b(t) \times \left( \frac{1 - t}{T} \right) + (X_M(t) - X_b(t) * rand), \quad (2)$$

$$X_M(t) = \frac{1}{N} \sum_{i=1}^N X(t), \quad \forall j = 1, 2, 3, \dots, Dim \quad (3)$$

In Equation (2), the search is governed by  $\left( \frac{1-t}{T} \right)$ .  $T$  stands for the max generations number.

Meanwhile, the levy flight ( $Levy(D)$ ) is used in exploration phase to improve the ability of the population to find the best solution. This process is formulated as:

$$X_i(t + 1) = Levy(D) \times X_b(t) + X_R(t) + (y - x) * rand, \quad (4)$$

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

where  $\beta$  and  $s$  are set to 0.01 and 1.5, respectively.  $v$  and  $u$  are generated randomly.  $X_R$  represents a solution selected randomly from  $X$ . Furthermore,  $y$  and  $x$  denote two parameters that are utilized to replicate the spiral shape:

$$y = r \times \cos(\theta), x = \sin(\theta) \times r \tag{6}$$

$$r = r_1 + U \times D_1, \theta = -\omega \times D_1 + \theta_1, \theta_1 = \frac{3 \times \pi}{2}, U = 0.005650, \omega = 0.0050 \tag{7}$$

here  $r_1 \in [0, 20]$  stands for random value.

The initial strategy employed in [7] to improve the agents in the exploitation phase, comparable to exploration, is based on both  $X_b$  and  $X_M$ , which is written as:

$$X_i(t + 1) = (X_b(t) - X_M(t)) \times \alpha - rnd + ((UB - LB) \times rnd + LB) \times \delta \tag{8}$$

where  $\delta$  and  $\alpha$  are exploitation parameters.

In the second exploitation approach, the position is updated by *Levy*, the quality function *QF*, or  $X_b$ . The mathematical definition of this technique is as follows:

$$X_i(t + 1) = X_b(t) \times QF - GX - G_2 \times Levy(D) + rnd \times G_1 \tag{9}$$

$$GX = (X(t) \times G_1 \times rnd)$$

$$QF(t) = t^{\frac{2 \times rnd() - 1}{(1 - T)^2}} \tag{10}$$

where  $G_2$  is a parameter which updated using the following formula.

$$G_2 = 2 \times \left(1 - \frac{t}{T}\right) \tag{11}$$

In addition, the parameter  $G_1$  that is used to track the motion of the best solution is updated as:

$$G_1 = 2 \times rnd() - 1 \tag{12}$$

where *rnd* represents a random value. Algorithm 1 lists the steps of the AO.

### 2.2. Whale Optimization Algorithm

WOA [31] is an optimization technique. Its mathematical formulation is based on how well it does in hunting. The WOA uses a unique hunting approach that used by a killer whale species known as the humpback whale, which is known as bubble-net feeding. Each whale’s location indicates a solution that may be updated with respect to its attitude toward attacking the prey; such position is denoted by the symbol  $X_b$ . Whales can use two tactics to attack their prey [32]. The first strategy is known as encircling prey, in which the humpback whale can locate the target and encircle it. The target prey is assumed to be the best answer ( $X_b(t)$ ) by WOA. Once  $X_b(t)$  has been identified (found), the other whales will attempt to update their positions to match  $X_b(t)$ , as shown in Equations (13)–(15):

$$Dis_i = |B \odot X_b(t) - X_i(t)|, B = 2r \tag{13}$$

$$X_i(t + 1) = X_b(t) - A \odot Dis_i \tag{14}$$

In Equation (15),  $Dis_i$  stands for the distance between  $X_i(t)$  and  $X_b(t)$ .  $A$  is a coefficient vector and is calculated by the following equation:

$$A = 2a \odot r - a \tag{15}$$

where  $r \in [0, 1]$ . Whereas the parameter  $a$  denotes a parameter that updated its value as:

$$a = a - t \frac{a}{t_{max}} \tag{16}$$

where  $t_{max}$  stands for the number of generations.

---

**Algorithm 1** Aquila Optimizer (AO)

---

```

1: Input: dimension of each agent  $Dim$ ,
2: Solutions number  $N$ ,
3: Generations number  $T$ .
4: Form the initial population  $X$ .
5:  $t = t + 1$ 
6: while  $t \leq T$  do
7:   Compute the fitness for  $X_i$ .
8:   Select the best  $X_b(t)$ 
9:   for  $(i = 1, 2, \dots, N)$  do
10:    if  $t \leq (\frac{2}{3}) \times T$  then
11:      if  $rand() > 0.50$  then
12:        Update the  $X_i$  by Equation (2).
13:      else
14:        Update the  $X_i$  by Equation (4).
15:      end if
16:    else
17:      if  $rand() \leq 0.50$  then
18:        Update the  $X_i$  by Equation (8).
19:      else
20:        Update  $X_i$  by Equation (9).
21:      end if
22:    end if
23:    if  $Fit(X_3(t + 1)) < Fit(X(t))$  then
24:       $X(t) = (X_3(t + 1))$ 
25:    if  $Fit(X_3(t + 1)) < Fit(X_b(t))$  then
26:       $X_b(t) = X_3(t + 1)$ 
27:    end if
28:    end if
29:  end for
30: end while
31: Output: return  $(X_b)$ .

```

---

The second strategy is bubble-net attacking which represents the exploitation phase and it has two techniques: spiral updating position and shrinking encircling mechanism. Reduce the  $a$  in Equation (15) to satisfy the shrinking encircling process. The distance between  $X_i$  and  $X_b$  is calculated as follows using the spiral updating position mechanism:

$$X(t + 1) = Dis' \odot e^{bl} \odot \cos(2\pi l) + X_b(t) \tag{17}$$

In Equation (17),  $l$  is a value that stands for the shape from the logarithmic spiral. In this regard, the whales can also swim around the  $X_b$  utilizing a spiral-shaped path as well as a diminishing circle at the same time. As a result, the following equation, which is based on combining (13)–(15) and Equation (17), can be used to improve the location.

$$X(t + 1) = \begin{cases} X_b(t) - A \odot Dis & \text{if } p \geq 0.50 \\ Dis' \odot e^{bl} \odot \cos(2\pi l) + X_b(t) & \text{if } p < 0.50 \end{cases} \tag{18}$$

In Equation (18),  $p \in [0, 1]$  stands for probability used to control the updating mechanism.

Furthermore, instead of using  $X_b$ , each whale’s position can be updated through selecting arbitrary search whale,  $X_r$ , as shown in the equation below:

$$X(t + 1) = X_r - A \odot Dis \tag{19}$$

$$Dis = |B \odot X_{rand} - X(t)| \tag{20}$$

Algorithm 2 provides the basic steps of WOA.

---

**Algorithm 2** WOA
 

---

```

1: Input: The max iterations number  $t_{max}$  and the population size  $N$ .
2: Create a set of  $N$  random solutions ( $X$ ).
3: Set  $t = 1$ .
4: Calculate each  $X_i$ 's fitness value ( $F_i$ ).
5: Find the  $X_b$  value that corresponds to the best fitness value  $F_b$ .
6: for  $t = 1 : t_{max}$  do
7:   while  $a > 0$  do
8:     for all  $x_i \in X$  do
9:       Update the value of  $p$  randomly as  $p = rand$ .
10:      if  $p \geq 0.5$  then
11:        To improve  $X_i$ , use Equation (17).
12:      else
13:        if  $|A| \geq 0.5$  then
14:          To improve  $X_i$ , use Equation (19).
15:        else
16:          To improve  $X_i$ , use Equation (13).
17:        end if
18:      end if
19:    end for
20:    Update the value of  $a$ .
21:  end while
22: end for

```

---

### 3. Proposed Method

Herein, this section provides description on the IAO method. It utilizes the benefits of the WOA to enhance the performance of the original version of the AO. In detail, the WOA is used as a local search of the AO to raise its capability in solving different optimization problems which adds more ability and flexibility to the IAO to explore and exploit the search space as well as improve its diversity.

The structure of the IAO is illustrated in Figure 1. The IAO starts by declaring the global parameters and generating the initial population using random distribution methods. This population is evaluated to determine the best solution using the objective function. Throughout the optimization process of the IAO, the expanded exploitation of the original AO is improved using the spiral behavior of the WOA to update the solutions. In this regard, the expanded exploitation equation of the AO is replaced by the spiral equation of the WOA namely Equation (8), as in steps 30 to 36 in Algorithm 1, are updated using the WOA Equation (17). Therefore, the exploitation phase of the IAO benefits from both AO and WOA algorithms. Then, each solution is checked and updated by the objected function then the best one is retained to the subsequent iteration. Such sequence is iterated for all solutions till reaching stop condition, afterwards the best result within the population are selected and saved. Finally, the final results are presented.

Furthermore, the IAO begins by declaring the parameters of both AO and WOA. Then the AO generates a  $X [x_i, i = 1, 2, \dots, x_N]$  random binary population with size  $N$  and dimension  $D$ . The population values are converted to binary values by Equation (22).

$$X = \begin{cases} 1 & \text{if } x_i \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (21)$$

Then, the initial objective function value is computed using the operators of the AO whereas, the remaining values of the objective function are computed using the IAO structure. This sequence is iterated till meeting stop condition. So, in the final step, the

best results are presented as the output of the IAO. The formula Equation (22) is applied to compute the objective function value:

$$f(x_i(t)) = \xi E_{x_i(t)} + (1 - \xi) \left( \frac{|x_i(t)|}{|C|} \right) \tag{22}$$

where  $E_{x_i(t)}$  denotes the error of the fitness function Equation (23),  $\xi \in [0, 1]$  balances the error and the selected features number. The terms  $x_i(t)$  and  $C$  denote the selected features number and total feature numbers, respectively [5].

$$fit = h(X_i, t) = h_0(t) \exp \left[ \sum_{j=1}^p x_{ij} b_j \right] \tag{23}$$

where,  $X_i = (x_{i1}, x_{i2}, \dots)$  denotes the predictor variable.  $h_0(t)$  refers to the baseline hazard rate function.  $h(X_i, t)$  refers to the hazard rate at time  $t$  for  $X_i$ .

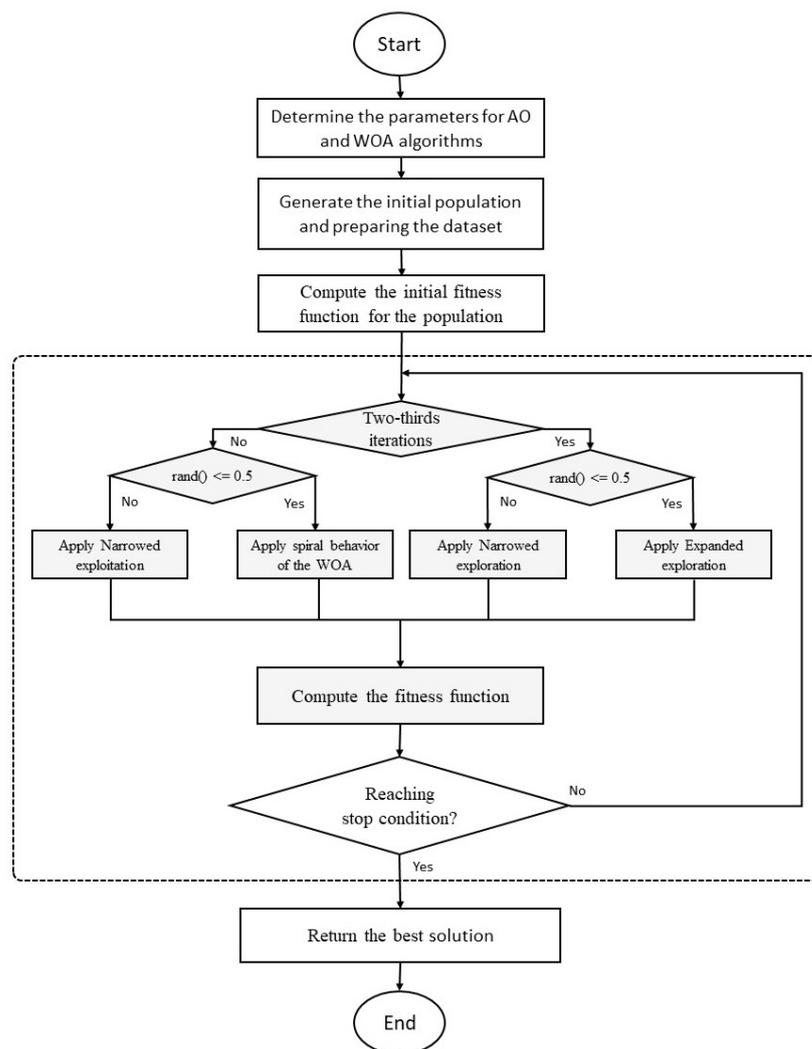


Figure 1. Structure of the IAO.

Moreover, the complexity of the presented algorithm depends on the complexity AO and WOA. So, it is  $O(N \times (t_{max} \times D))$ . Since  $O_{IAO} = K1O_{AO} + (N - K1)O_{WOA}$  where  $O_{AO}$  is  $O(N \times (t_{max} \times D))$ .  $O_{WOA}$  is  $O(N \times (t_{max} \times D))$ .

#### 4. Experiment and Results

The use of regression modeling to investigate the effects of many factors on a response is commonplace. In examining time-to-event data, the model of Cox proportional hazards is widely applied. It is a model applied in medical studies to examine the association between a patient's survival time and other predictor variables.

Creating a model of Cox proportional hazards that includes all of the predictors is undesirable when the number of predictors is enormous since it gives low prediction accuracy and is challenging to comprehend [33]. Variable selection has become a significant emphasis on Cox proportional-hazards modeling due to these factors.

Four biological benchmark datasets are used in this study to evaluate the modified variant of AO's (IAO) performance. Diffuse large B-cell lymphoma (DLBC2002) [34] comprises the samples of 240 lymphoma patients, each of one has 7399 gene expression measurements. The Lung cancer (Lung-cancer) [35] is the second dataset. This dataset comprises information on 86 lung cancer patients, each one had their gene expression tested 7129 times. The Dutch Breast Cancer (Duch Breasst) is the third dataset [36] contains 295 breast cancer patients' information. The data for each patient consists of 4919 gene expression measurements. The cytogenetically normal acute myeloid leukaemia dataset (AML-full) is the fourth collection [37]. This dataset contained information from 165 problems. A total of 6283 gene expression measurements are included in the data for each prob. The survival time, whether censored or not, is the response variable in both datasets.

The improved version of AO has been compared with a set of more popular optimizers, including the standard AO algorithm, firefly algorithm (Firefly), genetic algorithm (GA), salp swarm algorithm, indicated as SSA, particle swarm optimizer (PSO), in addition to differential evolution (DE), WOA and finally moth-flame optimizer (MFO). The parameters of those algorithms are listed in Table 1. All optimizers, as mentioned earlier, are implemented over 30 runs with 100 iteration numbers besides to 50 search agents on Matlab "2020a" platform for unbiased comparison. Several statistical metrics have been computed for providing a detailed analysis. The computed metrics of the Tables 2–6 are the average, worst (Max) (Equation (25)) and best (Min) (Equation (24)) fitness function ( $fit$ ) values (Equation (23)), moreover, the standard deviation (Equation (26)) and the selected features number are shown below:

$$Min_{fit} = \min_{1 \leq i \leq r} fit_i \quad (24)$$

$$Max_{fit} = \max_{1 \leq i \leq r} fit_i \quad (25)$$

$$Standard\ deviation = \sqrt{\frac{1}{N} \sum_{i=1}^N |fit_i - \mu|^2} \quad (26)$$

where,  $\mu$  refers to the fitness function ( $fit$ ).  $N$  refers to the sample number.

The average of the fitness function, Max, and Min values of the log-likelihood were reported in Tables 1–5, respectively, to highlight the performance that our IAO and other employed algorithms can achieve on the four datasets (in all Tables the best values are in boldface). According to Tables 1–3, the suggested algorithm, IAO, outperformed the different algorithms for all datasets as it has the least fitness function values. The reported data in Table 4 affirms the highly consistent performance of the proposed optimizer compared with the AO, Firefly, SSA, GA, PSO, DE, and MFO. The WOA can be located at the section rank after the proposed IAO in handling the first and fourth datasets (DLBC2002, AML-full). In contrast, WOA is not efficient for the Lung-cancer dataset. Moreover, WOA has a remarkable deviation from the IAO for the Duch Breast dataset. Accordingly, the IAO can be considered a successful technique across all datasets. For the selected features number of Table 5, the IAO chose fewer genes than the other algorithms. The MFO is the worst technique for handling these datasets as it picked the highest number of genes.

**Table 1.** Parameter settings.

Algorithm	Parameters Values
AO	$\delta = 0.1, \alpha = 0.1$
Firefly	$\beta = 0.2, \alpha = 0.5, \gamma = 1$
SSA	$C_3 \in [0, 1], C_2 \in [0, 1]$
GA	$\gamma = 0.20, pc = 0.80, mu = 0.020, pm = 0.30, \beta = 8$
PSO	$wDamp = 0.990, w = 1, C1 = 1, C2 = 2$
DE	$pCR = 0.20, \beta_{max} = 0.80, \beta_{min} = 0.20$
WOA	$a = [0, 2], l = [-1, 1], b = 1$
MFO	$a \in [-2 - 1], b = 1$
IAO	$\delta = 0.1, \alpha = 0.1, a = [0, 2], b = 1, l = [-1, 1]$

For measuring the accumulated performance of the IOA, the mean values of average, Min, Max, standard deviation, and the features number over the four datasets are depicted in Figures 2–6. The displayed figures are a shred of evidence of the efficiency and superiority of the IAO and its success in handling the four datasets as it has the minor average, Min, Max, standard deviation, and the selected features number with high performance. These observations are primarily due to the created algorithm’s ability to adjust for the limitations of the typical AO algorithm. In addition, Figure 7 illustrates the average of the computation time overall datasets. From this figure the IAO showed accepted time compared to the other methods.

**Table 2.** Numerical results of the fitness functions.

DS	IAO	AO	Firefly	SSA	GA	PSO	DE	WOA	MFO
DLBC2002	<b>−233.043</b>	−232.904	−231.880	−229.445	−230.639	−231.227	−230.936	−231.989	−230.212
Lung-cancer	<b>−62.913</b>	−58.9001	−58.171	−57.963	−58.223	−58.271	−58.451	−61.276	−57.297
Duch Breasst	<b>−307.687</b>	−301.913	−305.818	−303.578	−304.606	−305.465	−305.285	−305.154	−301.364
AML-full	<b>−93.533</b>	−88.8636	−88.822	−88.304	−88.416	−88.759	−88.737	−92.010	−88.075

**Table 3.** Numerical results of the Max.

DS	IAO	AO	Firefly	SSA	GA	PSO	DE	WOA	MFO
DLBC2002	<b>−230.672</b>	−229.409	<b>−230.672</b>	−227.704	−228.858	−229.427	−229.427	−230.671	−227.704
Lung-cancer	<b>−58.087</b>	−56.1811	−56.463	−55.993	−56.463	−56.463	−56.662	−57.290	−55.969
Duch Breasst	<b>−303.420</b>	−298.355	−301.943	−300.563	−299.440	−301.887	−301.979	−300.589	−297.438
AML-full	<b>−91.573</b>	−86.986	−86.808	−87.063	−87.214	−87.655	−87.451	−90.867	−87.107

**Table 4.** Numerical results of the Min.

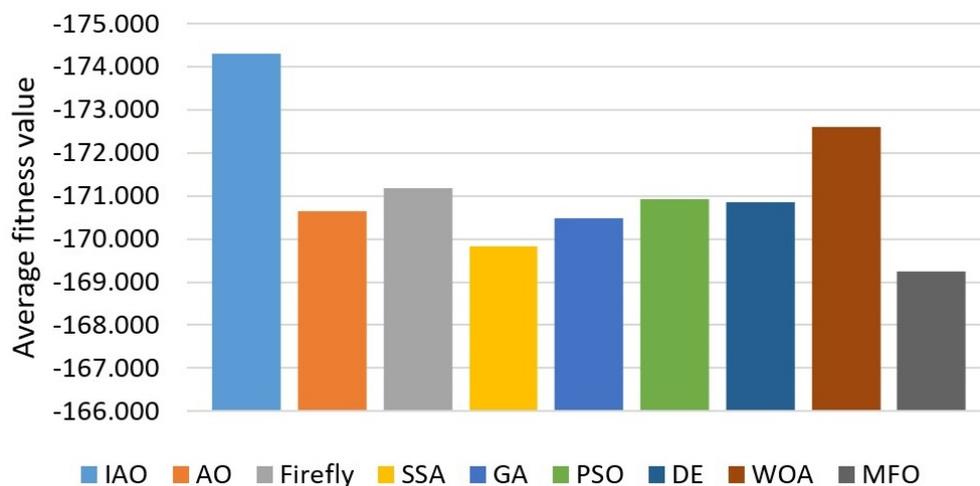
DS	IAO	AO	Firefly	SSA	GA	PSO	DE	WOA	MFO
DLBC2002	<b>−233.409</b>	−233.343	−233.342	−231.866	−233.310	<b>−233.409</b>	<b>−233.409</b>	−232.713	−232.010
Lung-cancer	<b>−63.935</b>	−63.935	<b>−63.935</b>	<b>−63.935</b>	<b>−63.935</b>	<b>−63.935</b>	<b>−63.935</b>	−63.935	−60.872
Duch Breasst	<b>−314.193</b>	−306.826	<b>−314.193</b>	<b>−314.193</b>	−311.276	−314.184	<b>−314.193</b>	−311.264	−311.276
AML-full	<b>−93.838</b>	−90.8613	−91.573	−90.849	−90.849	−90.849	−91.573	−93.109	−89.601

**Table 5.** Numerical results of the standard deviation.

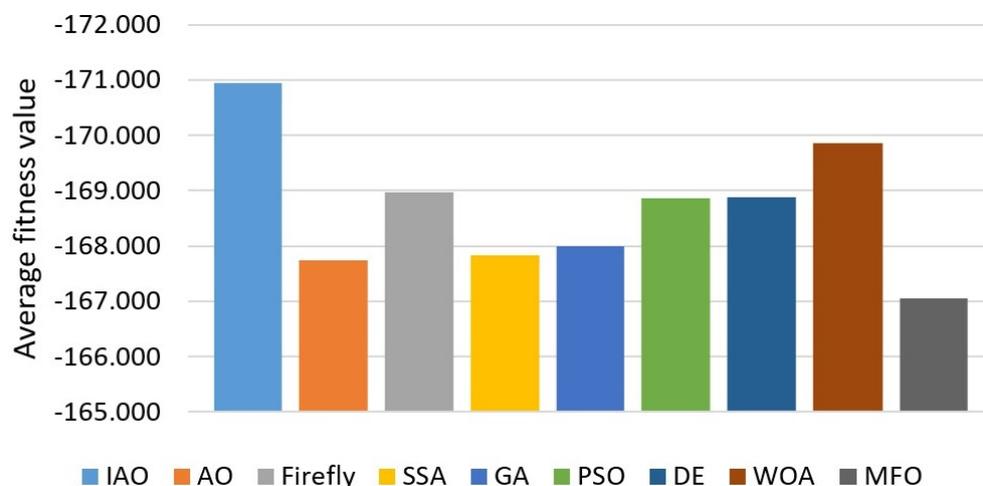
DS	IAO	AO	Firefly	SSA	GA	PSO	DE	WOA	MFO
DLBC2002	<b>0.77164</b>	1.0321	0.84894	1.25801	1.28674	1.21517	1.24301	0.80291	1.25666
Lung-cancer	<b>1.33654</b>	2.14285	2.34839	2.44722	2.32250	2.28789	2.19259	2.49236	1.40572
Duch Breasst	<b>2.46672</b>	1.37495	3.53870	4.31523	3.63534	3.90358	3.94957	3.23130	4.18625
AML-full	<b>0.67068</b>	1.17163	1.69382	1.32353	1.21093	1.21578	1.47812	0.92584	0.95712

**Table 6.** Numerical results of the selected features ratio.

DS	IAO	AO	Firefly	SSA	GA	PSO	DE	WOA	MFO
DLBC2002	<b>0.3560</b>	0.43878	0.4997	0.4996	0.4995	0.4982	0.4996	0.3914	0.5348
Lung-cancer	<b>0.3368</b>	0.43412	0.4980	0.4987	0.4970	0.4962	0.4982	0.3824	0.5288
Duch Breasst	<b>0.4042</b>	0.41995	0.4972	0.5010	0.4953	0.4993	0.5018	0.4660	0.5318
AML-full	<b>0.4152</b>	0.44055	0.5005	0.5009	0.5029	0.5030	0.4984	0.4637	0.5253



**Figure 2.** Average computed from the fitness values over the datasets.



**Figure 3.** Average computed from the MAX fitness values over the datasets.

The Friedman test is applied to check the statistical significance of the experiment’s methods for further analysis. It is one of the most important statistical tests that indicate the significant differences between the compared algorithms [38,39]. Table 7 ranks all methods in all datasets utilizing the Friedman test. From Table 7, we can conclude that the IAO was ranked first in DLBC2002, Lung-cancer, and AML-full datasets whereas, it was ranked second in Duch Breasst after the Firefly method. Moreover, the IAO showed good performance as in Figures 8–11 which illustrate the boxplot for all datasets.



Figure 4. Average computed from the MIN fitness values over the datasets.

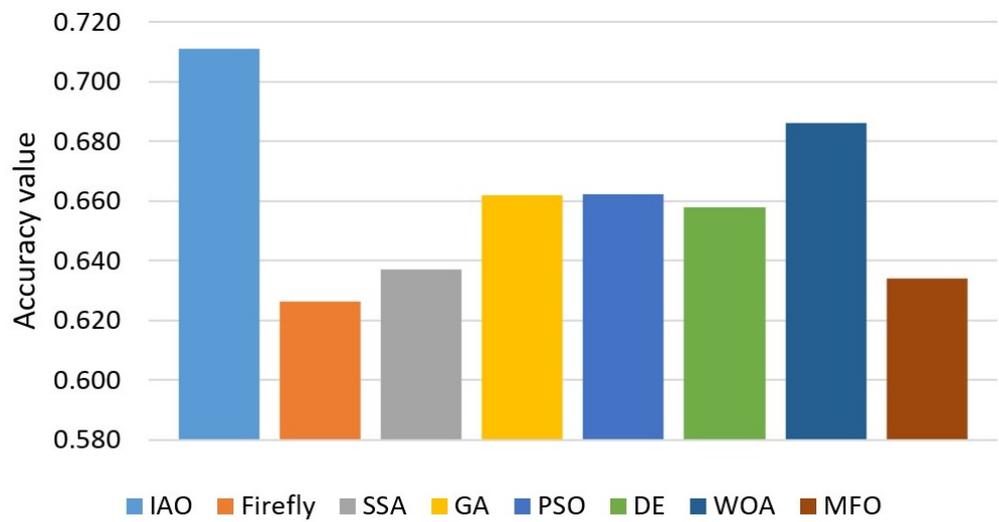


Figure 5. Average accuracy measure over all datasets.

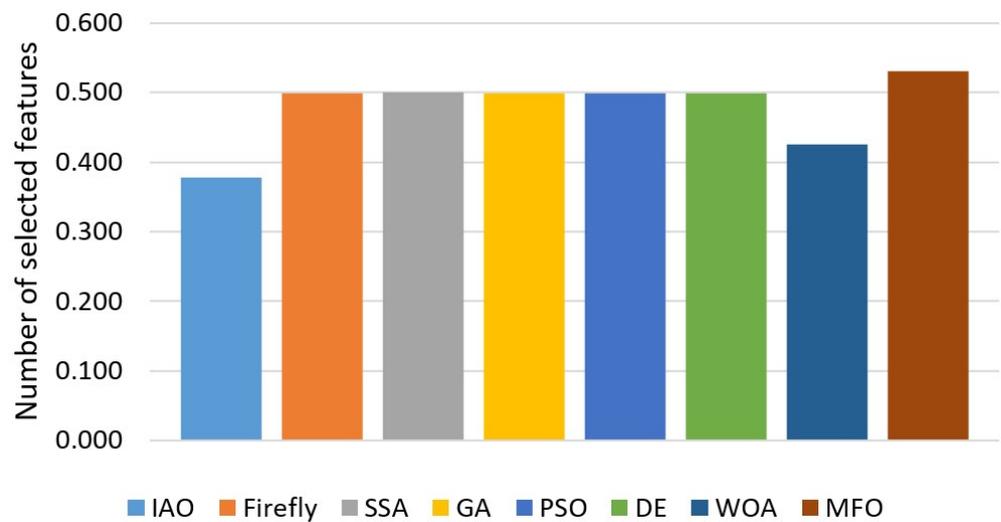


Figure 6. Average of the selected features for all datasets.

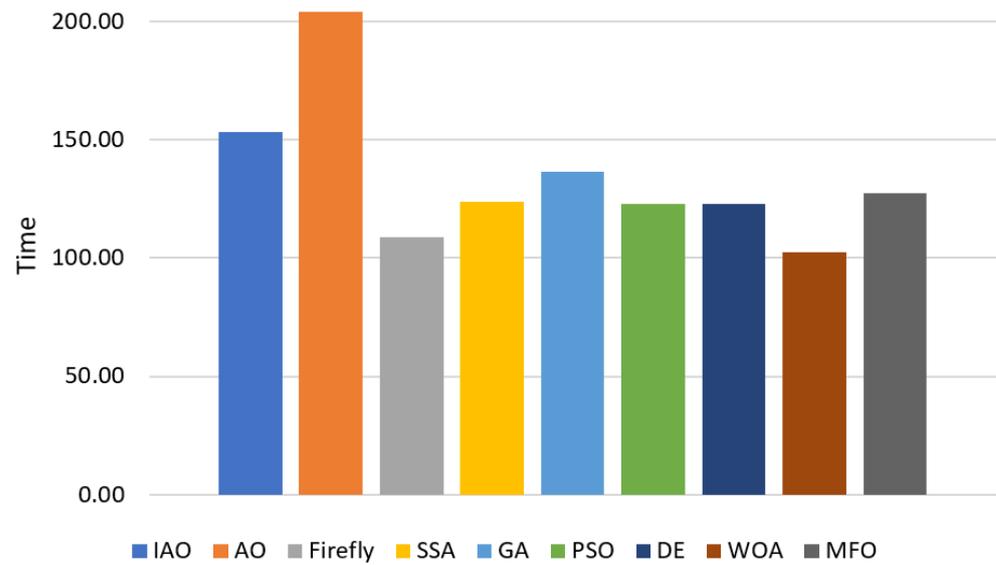


Figure 7. Average computation time.

Table 7. Results of the Friedman for all datasets.

	IAO	AO	Firefly	SSA	GA	PSO	DE	WOA	MFO
DLBC2002	<b>2.438</b>	3.750	3.438	8.563	6.375	4.500	5.563	3.438	6.938
Lung-cancer	<b>2.250</b>	5.750	5.375	6.563	5.000	5.125	4.313	2.375	8.250
Duch Breasst	3.188	7.000	<b>3.125</b>	6.125	4.813	3.938	3.563	4.938	8.313
AML-full	<b>1.000</b>	4.500	5.125	6.813	6.500	5.375	5.375	2.375	7.938

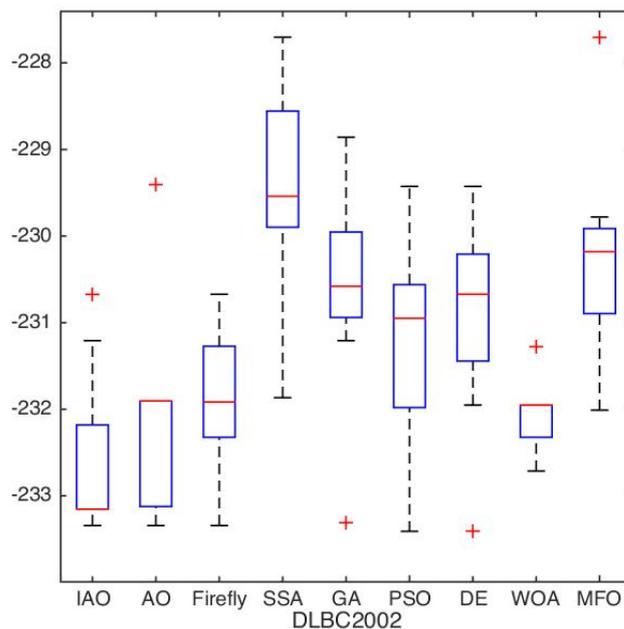


Figure 8. Boxplot for DLBC2002 dataset.

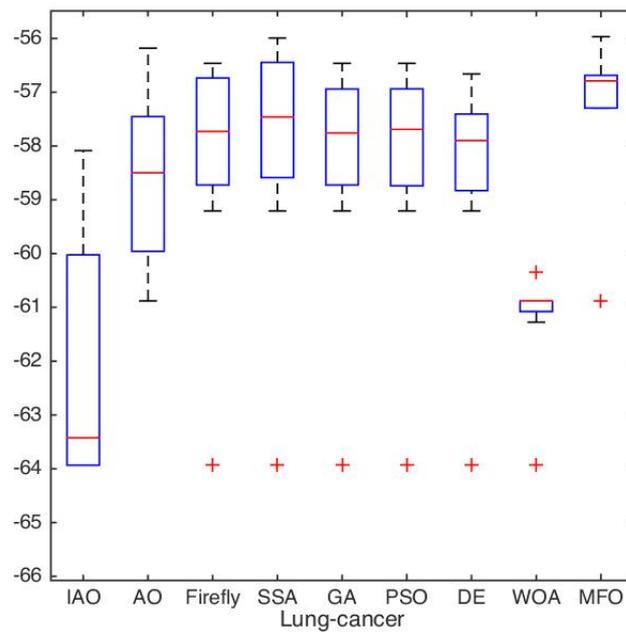


Figure 9. Boxplot for Lung-cancer dataset.

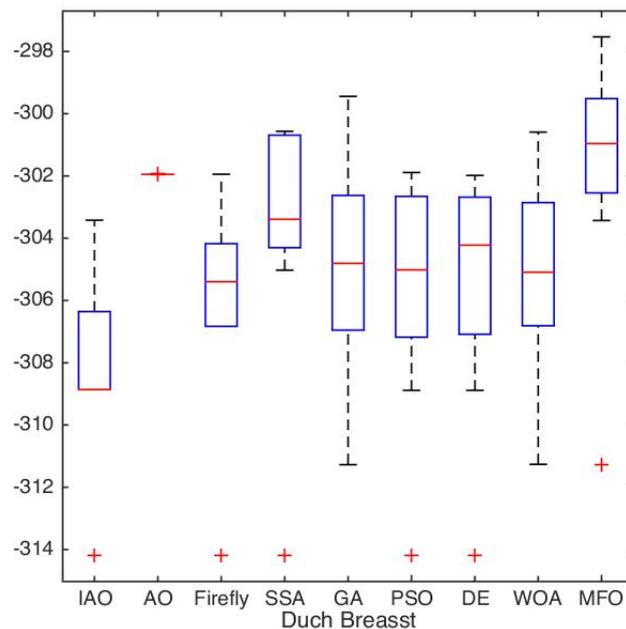


Figure 10. Boxplot for Duch Breasst dataset.

To analyse the exploration and exploitation of the IAO and the original version of AO, their behaviours with the studied datasets are illustrated in Figure 12. The curves of the Figure 12 illustrate the ratios of exploitation and exploration throughout the search stages at the studied datasets for the IOA and standard AO. From these curves, it can be observed that there is a balance between both curves of IAO throughout the search process. The exploration stage raises in the first part of the optimization stage. The exploitation is started after 10% of this process and working together with exploration with a nearly equal ratio as indicated in the cases of DLBC2002, Duch Breast, and AML-full, while the AO is still searching for adequate solutions. Hence the IAO achieves a successful trade-off between the two phases.

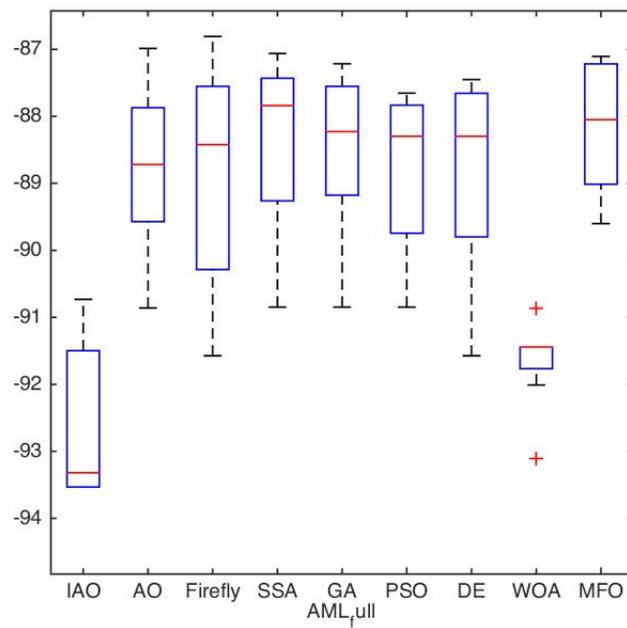


Figure 11. Boxplot for AML-full dataset.

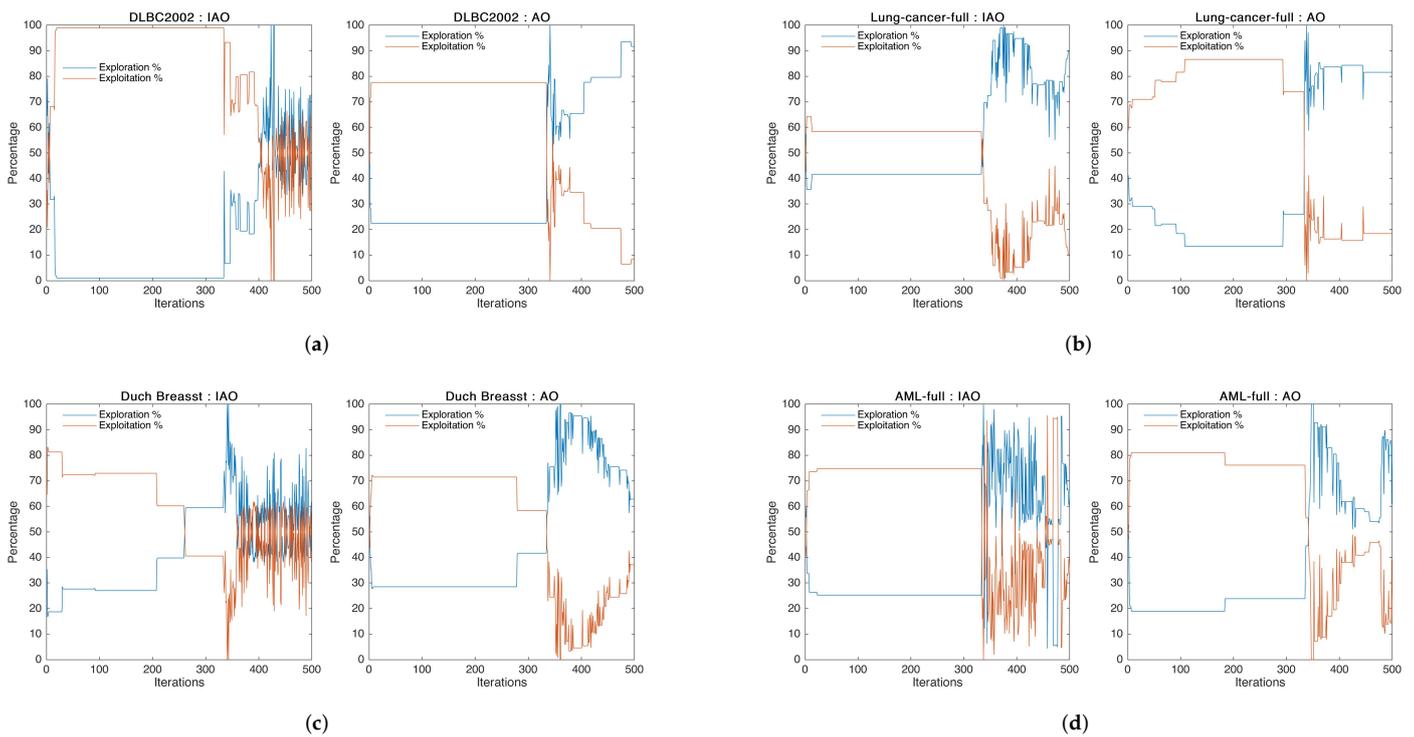


Figure 12. Exploration and exploitation curves of the IAO and AO for all datasets. (a) DLBC2002. (b) Lung-cancer. (c) Duch Breasst. (d) AML-full.

### 5. Advantages and Drawbacks

In this section, the attractive features and limitations of the proposed IAO can be summed up in the following bullet points:

- The advantages of the proposed IAO are achieving a high balance between the exploration and exploitation stages. The consistency and efficiency of the IAO are more remarkable than the standard AO. Moreover, the IAO’s ability to reach the optimal

solutions compared with the other competitors (Firefly, GA, SSA, PSO, DE, WOA, and MFO).

- The main limitation of the proposed IAO is increasing the adjustable parameters of the algorithm via using the operators of WOA. For solving this issue, the authors will modify these parameters adaptively in future work.

## 6. Conclusions

The advances in optimization methods have been adopted to address different problems, including feature selection (FS). Therefore, this study proposed a novel version of the Aquila optimizer (AO) to solve FS applications. The main idea of this improved, called IAO is to boost the search performance of the original AO using the operators of the Whale Optimization Algorithm (WOA). Thus, this new combination provides the proposed IAO with a strong search ability to avoid local optima stagnation. We considered extensive evaluation and comparisons to verify the quality of the suggested IAO. We used four medical datasets to test the IAO. The results reflected that the IAO performed better than the versions of AO and WOA, as well as several well-known optimization methods, such as a particle swarm optimizer, differential evolution, mouth flame optimizer, firefly algorithm, and genetic algorithm. In future work, the IAO will be evaluated in different fields for example multi-objective optimization and parameter estimation.

**Author Contributions:** Conceptualization, A.A.E. and M.A.A.A.-q.; Data curation, Z.Y.A.; Formal analysis, A.A.E., Z.Y.A., L.A. and D.Y.; Investigation, A.A.E., L.A., M.A.A.A.-q., R.M.G. and M.A.E.; Methodology, A.A.E., Z.Y.A. and M.A.A.A.-q.; Resources, Z.Y.A., R.M.G. and M.A.E.; Software, A.A.E.; Validation, A.A.E., Z.Y.A., D.Y., R.M.G. and M.A.E.; Visualization, A.A.E., L.A., M.A.A.A.-q., D.Y., R.M.G. and M.A.E.; Writing—original draft, A.A.E., Z.Y.A., L.A., M.A.A.A.-q., D.Y., R.M.G. and M.A.E.; Writing—review & editing, A.A.E., Z.Y.A., L.A., M.A.A.A.-q., D.Y. and M.A.E. All authors have read and agreed to the published version of the manuscript.

**Funding:** Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R138), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

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

## References

1. Ghaemi, M.; Feizi-Derakhshi, M.R. Feature selection using forest optimization algorithm. *Pattern Recognit.* **2016**, *60*, 121–129. [[CrossRef](#)]
2. Agrawal, R.; Kaur, B.; Sharma, S. Quantum based whale optimization algorithm for wrapper feature selection. *Appl. Soft Comput.* **2020**, *89*, 106092. [[CrossRef](#)]
3. Gasca, E.; Sánchez, J.S.; Alonso, R. Eliminating redundancy and irrelevance using a new MLP-based feature selection method. *Pattern Recognit.* **2006**, *39*, 313–315. [[CrossRef](#)]
4. Chuang, L.Y.; Tsai, S.W.; Yang, C.H. Improved binary particle swarm optimization using catfish effect for feature selection. *Expert Syst. Appl.* **2011**, *38*, 12699–12707. [[CrossRef](#)]
5. 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](#)]
6. Arora, S.; Anand, P. Binary butterfly optimization approaches for feature selection. *Expert Syst. Appl.* **2019**, *116*, 147–160. [[CrossRef](#)]
7. 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](#)]
8. Jiang, Y.; Luo, Q.; Wei, Y.; Abualigah, L.; Zhou, Y. An efficient binary Gradient-based optimizer for feature selection. *Math. Biosci. Eng.* **2021**, *18*, 3813–3854. [[CrossRef](#)]
9. 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**, 1–15. [[CrossRef](#)]
10. Abd Elaziz, M.; Abualigah, L.; Yousri, D.; Oliva, D.; Al-qaness, M.A.; Nadimi-Shahraki, M.H.; Ewees, A.A.; Lu, S.; Ali Ibrahim, R. Boosting Atomic Orbit Search Using Dynamic-Based Learning for Feature Selection. *Mathematics* **2021**, *9*, 2786. [[CrossRef](#)]
11. Ahmadianfar, I.; Heidari, A.A.; Gandomi, A.H.; Chu, X.; Chen, H. RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method. *Expert Syst. Appl.* **2021**, *181*, 115079. [[CrossRef](#)]

12. Tu, J.; Chen, H.; Wang, M.; Gandomi, A.H. The colony predation algorithm. *J. Bionic Eng.* **2021**, *18*, 674–710. [[CrossRef](#)]
13. Li, S.; Chen, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* **2020**, *111*, 300–323. [[CrossRef](#)]
14. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [[CrossRef](#)]
15. Yang, Y.; Chen, H.; Heidari, A.A.; Gandomi, A.H. Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. *Expert Syst. Appl.* **2021**, *177*, 114864. [[CrossRef](#)]
16. Tabakhi, S.; Moradi, P.; Akhlaghian, F. An unsupervised feature selection algorithm based on ant colony optimization. *Eng. Appl. Artif. Intell.* **2014**, *32*, 112–123. [[CrossRef](#)]
17. Neggaz, N.; Houssein, E.H.; Hussain, K. An efficient henry gas solubility optimization for feature selection. *Expert Syst. Appl.* **2020**, *152*, 113364. [[CrossRef](#)]
18. Kabir, M.M.; Shahjahan, M.; Murase, K. A new hybrid ant colony optimization algorithm for feature selection. *Expert Syst. Appl.* **2012**, *39*, 3747–3763. [[CrossRef](#)]
19. Zakeri, A.; Hokmabadi, A. Efficient feature selection method using real-valued grasshopper optimization algorithm. *Expert Syst. Appl.* **2019**, *119*, 61–72. [[CrossRef](#)]
20. Lu, Y.; Liang, M.; Ye, Z.; Cao, L. Improved particle swarm optimization algorithm and its application in text feature selection. *Appl. Soft Comput.* **2015**, *35*, 629–636. [[CrossRef](#)]
21. Zhang, Y.; Ma, Y. Non-negative multi-label feature selection with dynamic graph constraints. *Knowl.-Based Syst.* **2021**, *238*, 107924. [[CrossRef](#)]
22. Hichem, H.; Elkamel, M.; Rafik, M.; Mesaaoud, M.T.; Ouahiba, C. A new binary grasshopper optimization algorithm for feature selection problem. *J. King Saud- Univ.-Comput. Inf. Sci.* **2019**, *34*, 316–328. [[CrossRef](#)]
23. Eesa, A.S.; Orman, Z.; Brifcani, A.M.A. A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems. *Expert Syst. Appl.* **2015**, *42*, 2670–2679. [[CrossRef](#)]
24. Chen, Y.P.; Li, Y.; Wang, G.; Zheng, Y.F.; Xu, Q.; Fan, J.H.; Cui, X.T. A novel bacterial foraging optimization algorithm for feature selection. *Expert Syst. Appl.* **2017**, *83*, 1–17. [[CrossRef](#)]
25. AlRassas, A.M.; Al-qaness, M.A.; Ewees, A.A.; Ren, S.; Abd Elaziz, M.; Damaševičius, R.; Krilavičius, T. Optimized ANFIS model using Aquila Optimizer for oil production forecasting. *Processes* **2021**, *9*, 1194. [[CrossRef](#)]
26. Fatani, A.; Dahou, A.; Al-qaness, M.A.; Lu, S.; Abd Elaziz, M. Advanced Feature Extraction and Selection Approach Using Deep Learning and Aquila Optimizer for IoT Intrusion Detection System. *Sensors* **2021**, *22*, 140. [[CrossRef](#)]
27. Kandan, M.; Krishnamurthy, A.; Selvi, S.; Sikkandar, M.Y.; Aboamer, M.A.; Tamilvizhi, T. Quasi oppositional Aquila optimizer-based task scheduling approach in an IoT enabled cloud environment. *J. Supercomput.* **2022**, 1–15. [[CrossRef](#)]
28. Wang, S.; Jia, H.; Liu, Q.; Zheng, R. An improved hybrid Aquila Optimizer and Harris Hawks Optimization for global optimization. *Math. Biosci. Eng.* **2021**, *18*, 7076–7109. [[CrossRef](#)]
29. Zhang, Y.J.; Yan, Y.X.; Zhao, J.; Gao, Z.M. AOAAO: The Hybrid algorithm of Arithmetic Optimization algorithm with Aquila Optimizer. *IEEE Access* **2022**, *10*, 10907–10933. [[CrossRef](#)]
30. Vashishtha, G.; Kumar, R. Autocorrelation energy and aquila optimizer for MED filtering of sound signal to detect bearing defect in Francis turbine. *Meas. Sci. Technol.* **2021**, *33*, 015006. [[CrossRef](#)]
31. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
32. Al-qaness, M.A.; Ewees, A.A.; Abd Elaziz, M. Modified whale optimization algorithm for solving unrelated parallel machine scheduling problems. *Soft Comput.* **2021**, *25*, 9545–9557. [[CrossRef](#)]
33. Leng, C.; Helen Zhang, H. Model selection in nonparametric hazard regression. *Nonparametr. Stat.* **2006**, *18*, 417–429. [[CrossRef](#)]
34. Rosenwald, A.; Wright, G.; Chan, W.C.; Connors, J.M.; Campo, E.; Fisher, R.I.; Gascoyne, R.D.; Muller-Hermelink, H.K.; Smeland, E.B.; Giltman, J.M.; et al. The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma. *N. Engl. J. Med.* **2002**, *346*, 1937–1947. [[CrossRef](#)] [[PubMed](#)]
35. Beer, D.G.; Kardia, S.L.; Huang, C.C.; Giordano, T.J.; Levin, A.M.; Misek, D.E.; Lin, L.; Chen, G.; Gharib, T.G.; Thomas, D.G.; et al. Gene-expression profiles predict survival of patients with lung adenocarcinoma. *Nat. Med.* **2002**, *8*, 816–824. [[CrossRef](#)]
36. Van Houwelingen, H.C.; Bruinsma, T.; Hart, A.A.; Van't Veer, L.J.; Wessels, L.F. Cross-validated Cox regression on microarray gene expression data. *Stat. Med.* **2006**, *25*, 3201–3216. [[CrossRef](#)]
37. Metzeler, K.H.; Hummel, M.; Bloomfield, C.D.; Spiekermann, K.; Braess, J.; Sauerland, M.C.; Heinecke, A.; Radmacher, M.; Marcucci, G.; Whitman, S.P.; et al. An 86-probe-set gene-expression signature predicts survival in cytogenetically normal acute myeloid leukemia. *Blood J. Am. Soc. Hematol.* **2008**, *112*, 4193–4201. [[CrossRef](#)]
38. Al-qaness, M.A.; Ewees, A.A.; Fan, H.; Abualigah, L.; Abd Elaziz, M. Boosted ANFIS model using augmented marine predator algorithm with mutation operators for wind power forecasting. *Appl. Energy* **2022**, *314*, 118851. [[CrossRef](#)]
39. Yousri, D.; AbdelAty, A.M.; Al-qaness, M.A.; Ewees, A.A.; Radwan, A.G.; Abd Elaziz, M. Discrete fractional-order Caputo method to overcome trapping in local optima: Manta Ray Foraging Optimizer as a case study. *Expert Syst. Appl.* **2022**, *192*, 116355. [[CrossRef](#)]