mathematics

MDPI

Article

Boosting Atomic Orbit Search Using Dynamic-Based Learning for Feature Selection

Mohamed Abd Elaziz [1,2,3,4], Laith Abualigah [5,6], Dalia Yousri [7], Diego Oliva [8,9,*], Mohammed A. A. Al-Qaness [10], Mohammad H. Nadimi-Shahraki [11,12], Ahmed A. Ewees [13], Songfeng Lu [1,14,*] and Rehab Ali Ibrahim [2]

[1] School of Cyber Science & Engineering, Huazhong University of Science and Technology, Wuhan 430074, China; abd_el_aziz_m@yahoo.com
[2] Department of Mathematics, Faculty of Science, Zagazig University, Zagazig 44519, Egypt; rehab100r@yahoo.com
[3] Artificial Intelligence Research Center (AIRC), Ajman University, Ajman 346, United Arab Emirates
[4] Department of Artificial Intelligence Science & Engineering, Galala University, Galala 44011, Egypt
[5] Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11183, Jordan; aligah.2020@gmail.com
[6] School of Computer Sciences, Universiti Sains Malaysia, George Town 11800, Pulau Pinang, Malaysia
[7] Electrical Engineering Department, Faculty of Engineering, Fayoum University, Faiyum 63514, Egypt; day01@fayoum.edu.eg
[8] Departamento de Ciencias Computacionales, Universidad de Guadalajara, CUCEI, Guadalajara 44430, Mexico
[9] School of Computer Science & Robotics, Tomsk Polytechnic University, 634050 Tomsk, Russia
[10] State Key Laboratory for Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China; alqaness@whu.edu.cn
[11] Big Data Research Center, Najafabad Branch, Islamic Azad University, Najafabad 8514143131, Iran; nadimi@iaun.ac.ir
[12] Faculty of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad 8514143131, Iran
[13] Department of Computer, Damietta University, Damietta 34511, Egypt; ewees@du.edu.eg
[14] Technology Research Institute and Shenzhen Huazhong University of Science, Shenzhen 518057, China
* Correspondence: diego.oliva@cucei.udg.mx (D.O.); lusongfeng@hust.edu.cn (S.L.)

check for updates

Citation: Elaziz, M.A.; Abualigah, L.; Yousri, D.; Oliva, D.; Al-Qaness, M.A.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. https://doi.org/10.3390/math9212786

Academic Editors: Liangxiao Jiang and Hsien-Chung Wu

Received: 24 September 2021
Accepted: 27 October 2021
Published: 3 November 2021

Abstract: Feature selection (FS) is a well-known preprocess step in soft computing and machine learning algorithms. It plays a critical role in different real-world applications since it aims to determine the relevant features and remove other ones. This process (i.e., FS) reduces the time and space complexity of the learning technique used to handle the collected data. The feature selection methods based on metaheuristic (MH) techniques established their performance over all the conventional FS methods. So, in this paper, we presented a modified version of new MH techniques named Atomic Orbital Search (AOS) as FS technique. This is performed using the advances of dynamic opposite-based learning (DOL) strategy that is used to enhance the ability of AOS to explore the search domain. This is performed by increasing the diversity of the solutions during the searching process and updating the search domain. A set of eighteen datasets has been used to evaluate the efficiency of the developed FS approach, named AOSD, and the results of AOSD are compared with other MH methods. From the results, AOSD can reduce the number of features by preserving or increasing the classification accuracy better than other MH techniques.

Keywords: soft computing; machine learning; feature selection (FS); metaheuristic (MH); atomic orbital search (AOS); dynamic opposite-based learning (DOL)

1. Introduction

Data has become the backbones of different fields and domains in recent decades, such as artificial intelligence, data science, data mining, and other related fields. The vast increase of data volumes produced by the web, sensors, and different techniques and systems raised a considerable problem with this excellent data size. The problems of the high dimensionality and big size data have particular impacts on the machine learning classification techniques, represented by the high computational cost and decreasing the classification accuracy [1–3]. To solve such challenges, Dimensionality Reduction (DR) techniques can be employed [4–6]. There are two main types of DR, called feature selection (FS) and feature extraction (FE). FS methods can remove noisy, irrelevant, and redundant data, which also improves the classifier performance. In general, FS techniques select a subset of the data that capture the characteristics of the whole dataset. To do so, two main types of FS, called filter and wrapper, have been widely used. Wrapper methods leverage the learning classifiers to evaluate the chosen features, where filter methods leverage the characteristic of the original data. Filter methods can be considered more efficient than wrapper methods [7]. FS techniques are used in various domains, for example, big data analysis [8], text classification [9], chemical applications [10], speech emotion recognition [11], neuromuscular disorders [12], hand gesture recognition [13], COVID-19 CT images classification [14], and other many other topics [15].

FS is considered as a complex optimization process, which has two objectives. The first one is to minimize the number of features and minimize error rates or maximize the classification accuracy. Therefore, metaheuristics (MH) optimization algorithms have been widely employed for different FS applications, such as differential evolution (DE) [16], genetic algorithm (GA) [17], particle swarm optimization (PSO) [18], Harris Hawks optimization (HHO) algorithm [7], salp swarm algorithm (SSA) [19], grey wolf optimizer [20], butterfly optimization algorithm [21], multi-verse optimizer (MVO) algorithm [22], krill herd algorithm [23], moth-flame optimization (MFO) algorithm [24] Henry gas solubility optimization (HGS) algorithm [25], and many other MH optimization algorithms [26,27].

In the same context, Atomic Orbital Search (AOS) [28] has been proposed as a metaheuristic technique that belongs to physical-based categories. AOS simulates the laws of quantum technicians and the quantum-based atomic design where the typical arrangement of electrons around the nucleus is in attitude. According to the characteristic of AOS, it has been applied to different applications such as global optimization [28]. In [29], AOS has been used to find the optimal solution to various engineering problems. With these advantages of AOS, it suffers from some limitations such as attraction to local optima, leading to degradation of the convergence rate. This motivated us to provide an improved version for AOS.

The enhanced AOS depends on using the dynamic opposite-based learning strategy to improve the exploration and maintain the diversity of solutions during the searching process. DOL is used in this study since it has several properties that will enhance the performance of different MH techniques. For example, it has been applied to improve the performance for antlion optimizer in [30], and this modification is applied to solve CEC 2014 and CEC 2017 benchmark problems. In [31], the SCA has been enhanced using DOL, and the developed method is applied to the problem of designing the plat-fin heat exchangers. In [32], the flexible job scheduling problem has been solved using the modified version of the grasshopper optimization algorithm (GOA) using DOL. Enhanced teaching–learning-based optimization (TLBO) is presented using DOL, and this algorithm is applied to CEC 2014 benchmark functions.

The main contributions of this study are:

1. We propose an alternative feature selection method to improve the behavior of atomic Orbit optimization (AOS).
2. We use the dynamic opposite-based learning to enhance the exploration and maintain the diversity of solutions during the searching process.
3. We compare the performance of the developed AOSD with other MH techniques using different datasets.

The other sections of this study are organized as follows. Section 2 presents the related works and Section 3 introduces the background of AOS and DOL. The developed method is introduced in Section 4. Section 5 introduces the experiment results and the discussion of the experiments using different FS datasets. The conclusion and future works are presented in Section 6.

2. Related Works

In recent years, many MH natural-inspired optimization algorithms have been used in the field of feature selection [33–36]. This section presents a simple review of the latest MH optimization techniques used for FS applications. Hu et al. [37] proposed a modified binary gray wolf optimizer (BGWO) for FS applications. They developed five transfer functions to enhance the BGWO. The authors evaluated the developed approach using different datasets. They concluded that the applications of the extended transfer functions improved the performance of the developed BGWO, and it outperformed the traditional BGWO and GWO. In [38], an FS approach was developed based on the multi-objective Particle Swarm Optimization (PSO) with fuzzy cost. The main idea of this approach is to develop a simple technique, called fuzzy dominance relationship, which is employed to compare the performance of the candidate particles. In addition, it is used to define a fuzzy crowding distance measure to determine the global leader of the particles. This method, called PSOMOFS, was evaluated with UCI datasets and compared to several FS techniques to confirm its competitive performance. Gao et al. [39] developed two variants of the binary equilibrium optimizer (BEO) using two techniques. The first technique is developed by mapping the continuous equilibrium optimizer into discrete types with S and V-shaped transfer functions (BEO-S and BEO-V). The second technique depends on the current target (solution) and the position vector (BEO-T). The two variants of the BEO were evaluated with nineteen UCI datasets, and they obtained good results. Al-tashi et al. [40] proposed a new variant of the GWO for FS applications. The proposed method, called binary multi-objective GWO, is developed using the sigmoid transfer function (BMGWO-S). It was tested with fifteen UCI datasets, and it outperformed the traditional multi-objective GWO (MGWO) and several well-known optimization algorithms. Alazam et al. [41] proposed a wrapper-based FS method using a pigeon-inspired optimizer. The proposed FS method was applied for intrusion detection systems (IDS) in cloud computing environments. It was evaluated using the three well-known IDS dataset, and it improved the classification accuracy of the IDS. Zhang et al. [42] developed the binary version of the differential evolution (BDE) for FS. They used several developed operators to enhance the performance of the BDE, such as the mutation operator and One-bit Purifying Search operator. The evaluation outcomes showed that the application of the developed operators improved the performance of the BDE.

Additionally, different MH optimization algorithms have been developed and utilized for FS applications, such as the binary emperor penguin optimizer, proposed by Dhiman et al. [43]. Three modified binary versions of the dragonfly algorithm (BDA) were presented by [44] for FS, called linear, quadratic, and sinusoidal BDA. The experimental outcomes showed that Sinusoidal-BDA achieved the best performance compared to other modified versions of the BDA. A modified binary Harris hawks optimizer was proposed by Zhang et al. [45] for FS applications. The salp swarm algorithm was used to boost the search process of the original HHO and overcome its shortcomings. Sahlol et al. [46] proposed a modified marine predators algorithm (MPA) using the fractional-order technique. The developed method, called FO-MPA, was applied to enhance the classification accuracy of the COVID-19 CT images. Abdel-Basset et al. [47] proposed an FS approach using four binary versions slime mould algorithm (SMA).

3. Background

3.1. Atomic Orbital Search

The AOS is a newly developed optimization method [28], which is inspired by the laws of quantum technicians where the typical arrangement of electrons around the nucleus is in attitude. The mathematical representation of the AOS is given as follows.

The AOS algorithm uses several solutions (X) as shown in Equation (1), and each solution (X_i) holds several decision variables ($x_{i,j}$).

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^j & \cdots & x_1^D \\ x_2^1 & x_2^2 & \cdots & x_2^j & \cdots & x_2^D \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_i^1 & x_i^2 & \cdots & x_i^j & \cdots & x_i^D \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N^1 & x_N^2 & \cdots & x_N^j & \cdots & x_N^D \end{bmatrix}, i = 1, 2, ..., N, \ j = 1, 2, ..., D \tag{1}$$

where N represents the number of used solutions, and D indicates the dimension length of the tested problem.

The first solutions are randomly initialized using Equation (2).

$$x_i^j = x_{i,min}^j + rand \times (x_{i,max}^j - x_{i,min}^j), \tag{2}$$

where x_i^j the position number i in the solution number j, $x_{i,min}^j$ indicates the lower bound of the ith position, and $x_{i,max}^j$ represents the upper bound of the ith position.

A vector of energy values includes the objective function of different solutions as presented in Equation (3).

$$E = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_i \\ \vdots \\ E_m \end{bmatrix} \tag{3}$$

where E represents a vector of objective values, and E_i refers to the energy level of the solution number i.

The electron likelihood density chart defines solutions positions estimated using the Probability Density Function (PDF). According to the given description of the individuals by PDF, each imaginarily formulated layer includes several solutions. In this respect, the mathematical representation of the K_k positions and the E_k of the used individuals in imaginary courses are given as below:

$$X = \begin{bmatrix} X_1^k \\ X_2^k \\ \vdots \\ X_i^k \\ \vdots \\ X_p^n \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^j & \cdots & x_1^d \\ x_2^1 & x_2^2 & \cdots & x_2^j & \cdots & x_2^d \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_i^1 & x_i^2 & \cdots & x_i^j & \cdots & x_i^d \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_p^1 & x_p^2 & \cdots & x_p^j & \cdots & x_p^d \end{bmatrix}, i = 1, 2, 3, ..., N, \ j = 1, 2, 3, ..., D, \tag{4}$$

$$E^k = \begin{bmatrix} E_1^k \\ E_2^k \\ \vdots \\ E_i^k \\ \vdots \\ E_p^n \end{bmatrix}, k = 1, 2, ..., n \tag{5}$$

where X_i^k is the solution number i in the imaginary layer (IL) number k, and n represents the number of the produced IL. p indicates the number of solutions of IL number k. E_i^k represents the objective value of the solution number i in the IL number k.

In this respect, the required state and energy are defined for the solutions in each supposed IL by analyzing all solutions' average positions and objective values in the felt layer. More so, the mathematical representation for this scheme is given as:

$$BS^k = \frac{\sum_{i=1}^{p} X_i^k}{p} \tag{6}$$

$$BE^k = \frac{\sum_{i=1}^{p} E_i^k}{p} \tag{7}$$

In Equation (7), BS^k and BE^k denote the required state and energy of the layer number k, respectively. X_i^k and E_i^k stand for the position and fitness value of the solution number i in k-th layer.

Depending on the given items, the required energy and state of an atom are defined by estimating the mean positions and objective values of the used solutions as follows:

$$BS = \frac{\sum_{i=1}^{m} X_i}{m} \tag{8}$$

$$BE = \frac{\sum_{i=1}^{m} E_i}{m} \tag{9}$$

where BS and BE are the required state and energy of the atom.

The energy level (E_i^k) of t X_i^k in each IL is associated with the required energy of the layer (BE^k). Suppose the energy ratio of the current solution in a particular layer is larger than the required energy (i.e., $E_i^k \geq BE^k$) so, the photon emission is estimated. In this rule, the individuals are managing to transmit a photon with a cost of energy estimated using γ and β to concurrently give to the required position of the atom (BS) and the position of the electron with the lowest energy ratio (LE) in the atom. The updating process of individuals is formulated as:

$$X_{i+1}^k = X_i^k + \frac{\alpha_i(\beta_i \times LE - \gamma_i \times BS)}{k}, \; k = 1, 2, \ldots, n, \; i = 1, 2, \ldots, p \tag{10}$$

in Equation (10), X_i^k and X_{i+1}^k denote the current and expected values for individual i at kth layer. α_i, β_i, and γ_i refer to random vectors.

Suppose the energy ratio of a solution in a particular layer is smaller than the required energy ($E_i^k < BE^k$); the consumption of photon is examined. The mathematical function for the position updating is presented as follows:

$$X_{i+1}^k = X_i^k + \alpha_i \times (\beta_i \times LE^k - \gamma_i \times BS^k) \tag{11}$$

In the case of generating a random number ($\varnothing$) for each individual and it is valued less than the PR (i.e., $\varnothing < PR$), the number of photons on the solution is not feasible. Therefore, the action of particles between various layers nearby the nucleus is estimated. The position updating is given as follows:

$$X_{i+1}^k = X_i^k + r_i \tag{12}$$

where r_i is a vector of random numbers.

3.2. *Dynamic-Opposite Learning*

The primary steps of the Dynamic-Opposition-Based Learning (DOL) approach are presented. In the beginning, the conventional Opposition-Based Learning (OBL) approach is presented [48]. This approach is used in this paper to enhance the performance of the proposed method. The OBL approach is employed to create a unique opposition solution to the existing solution. It attempts to determine the best solutions that lead to increasing the speed rate of convergence.

The opposite (X^o) of a given real number ($X \in [U, L]$) can be calculated as follows.

$$X^o = U + L - X \tag{13}$$

Opposite point [49]: Suppose that $X = [X_1, X_2, ..., X_{Dim}]$ is a point in a Dim-dimensional search space, and $X_1, X_2, ..., X_{Dim} \in R$ and X_j $[U_j, L_j]$. Thus, the opposite point (X^o) of X is presented as follows:

$$X^o_j = UB_j + L_j - X_j, \qquad where \quad j = 1....D. \tag{14}$$

Moreover, the most useful two points (X^o and X) are chosen according to the fitness function values, and the other is neglected. For the minimization problem, if $f(X) \leq f(X^o)$, X is maintained; oppositely, X^o is maintained.

Related to the opposite point, the dynamic opposite preference (X^{DO}) of the value X is represented as follows:

$$X^{Do} = X + w \times r_8(r_9 \times X^o - X), w > 0 \tag{15}$$

where r_8 and r_9 are random values in the range of [0 1]. w is weighting agent.

Consequently, the dynamic opposite value (X^{DO}_j) of X is equal to $[X_1, X_2, ..., X_{Dim}]$, which is presented as follows:

$$X^{Do}_j = X_j + w \times rand(rand \times X^o_j - X_j), w > 0 \tag{16}$$

Accordingly, DOL optimization begins by creating the first solutions ($X = (X_1, ..., X_{Dim})$ and calculate its dynamic opposite values (X^{Do}) using Equation (16). Next, based on the given fitness value, the best solution from the given (i.e., X^{Do} and X) is used, and another one is excluded.

4. Developed AOSD Feature Selection Algorithm

To improve the performance of the traditional AOS algorithm and use it as an FS method, we use dynamic opposite-based learning. The steps of the developed AOS-based DOL are given in Figure 1. These steps can be classified into two phases; the first one aims to learn the developed method based on the training set. At the same time, the second phase aims to assess the method's performance using the testing set.

4.1. Learning Phase

In this phase, the training set representing 70% from the input is applied to learn the model by selecting the optimal subset of relevant features. The developed AOSD aims at the beginning by constructing initial population, and this is achieved using the following formula:

$$X_i = rand * (U - L) + L, i = 1, 2, ..., N, j = 1, 2, ..., N_F \tag{17}$$

In Equation (17), N_F is the number of features (also, it is used to represents the dimension). U and L are the limits of the search domain. The next process in AOSD is to convert each agent X_i to binary form BX_i, and this is defined in Equation (20).

$$BX_{ij} = \begin{cases} 1 & if\ X_{ij} > 0.5 \\ 0 & otherwise \end{cases} \tag{18}$$

Thereafter, the fitness value of each X_i is computed, and it represents the quality. The following formula represents the fitness value that depends on the selected features from the training set.

$$Fit_i = \lambda \times \gamma_i + (1-\lambda) \times \left(\frac{|BX_i|}{N_F}\right), \tag{19}$$

where $|BX_i|$ is the number of features that correspond to the ones in BX_i. γ_i refers to the classification error obtained from the KNN classifier that learned using the reduced training set using features in BX_i. λ is applied to manage the process of selecting features which simulate reducing the error of classification.

The following process is to apply the DOL as defined in Equation (16) to each X_i to find X_i^{Do}. Then select from $X \cup X_{DO}$ the best N solutions that have the smallest fitness value. In addition, the best solution X_b is determined with best fitness Fit_b.

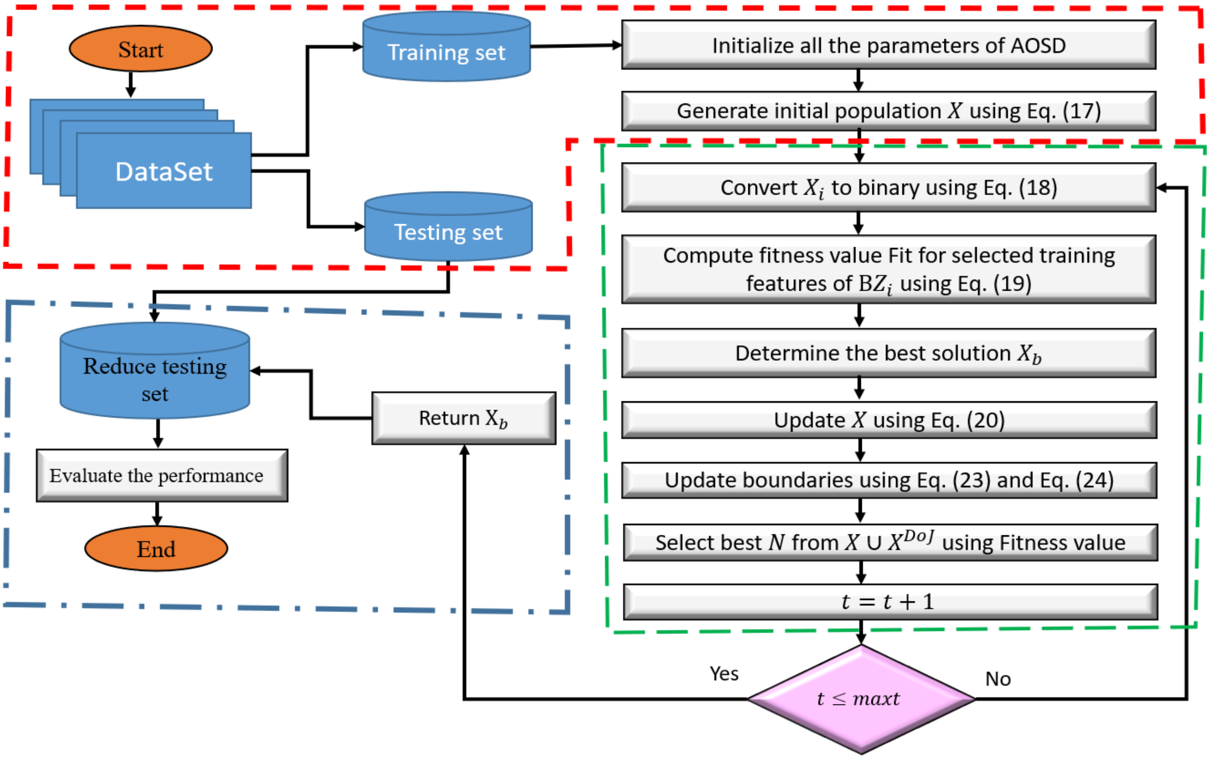
Start
DataSet
Training set
Testing set
Initialize all the parameters of AOSD
Generate initial population X using Eq. (17)
Convert X_i to binary using Eq. (18)
Compute fitness value Fit for selected training features of BZ$_i$ using Eq. (19)
Determine the best solution X_b
Update X using Eq. (20)
Update boundaries using Eq. (23) and Eq. (24)
Select best N from $X \cup X^{DoJ}$ using Fitness value
$t = t + 1$
$t \leq maxt$
Yes
No
Return X$_b$
Reduce testing set
Evaluate the performance
End

Figure 1. Steps of AOSD for FS problem.

After that, AOSD starts to update the solutions X using the operators of AOS as discussed in Section 3.1. To maintain the diversity of the solutions X, their opposite values are computed using the following formula:

$$X = \begin{cases} X & if\ Pr_{DO} > 0.5 \\ X_N & otherwise \end{cases} \tag{20}$$

where Pr_{DO} is random probability used to switch between X and X_N. X_N represents the N solutions chosen from $X \cup X^{DoJ}$ based on their fitness value. Whereas, X_{ij}^{DoJ} for each X_i at dimension j is given as:

$$X_{ij}^{DoJ} = X_{ij} + w \times rand(rand \times X_{ij}^{o} - X_{ij}), w > 0 \tag{21}$$

where X_{ij}^{o} is defined in Equation (16). In the developed AOSD, the limits of search space are updated dynamically using the following formula:

$$L_j = min(X_{ij}) \tag{22}$$

$$U_j = max(X_{ij}) \tag{23}$$

Thereafter, the terminal conditions are checked, and if they are met, then return by X_b. Otherwise, repeat the updating steps of AOSD.

4.2. Evaluation Phase

In this phase, the best solution X_b is employed to reduce the number of features of the testing set representing 30% from given data. This process is performed by selecting only those features corresponding to ones inside its binary version BX_b (computed using Equation (20)). Then, the KNN classifier is applied to the reduced testing set and it predicts the output of the testing set by computing the output's performance using performance measures.

5. Experimental Results

This section introduces the experimental evaluation of the developed AOSD method. Additionally, extensive comparisons to several existing optimization methods are carried out to verify the performance of the developed AOSD method.

5.1. Experimental Datasets and Parameter Settings

We considered comprehensive datasets to evaluate the proposed AOSD method using twenty datasets with different categories, including low and high dimensionality. The low dimensionality datasets are the well-known UCI datasets [50]. The properties of the used datasets are given in Table 1, including the number of classes, number of features, and number of samples. It is worth mentioning that the used datasets covered several domains, such as games, biology, biomedical, and physics.

Table 1. Datasets' characteristics.

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

Furthermore, we set up essential parameters and strategies to evaluate the proposed AOSD method. For example, we use the Hold-out strategy as a classification strategy, with 80% and 20% for training and testing sets, respectively. More so, we repeat each experiment with 30 independent runs. The K nearest neighbor (KNN) is adopted as the classifier with the Euclidean distance metric (K = 5).

In addition, a number of the well-known optimization algorithms have been considered for the comparison, such as Atomic Orbital Search (AOS), arithmetic optimization algorithm (AOA) [51], Marine Predators Algorithm (MPA) [46], Manta ray foraging optimizer (MRFO) [52], Harris Hawks optimization (HHO), Henry gas solubility optimization

(HGS) algorithm (HGSO), Whale optimization algorithm (WOA), grey wolf optimization (GWO) [53], GA, and BPSO. These methods are uniformly distributed, and the max iteration number is set to 100, where the population size is 10. In addition, the dimensions of these methods are fixed to the feature numbers as in the datasets.

5.2. Performance Measures

We used several evaluation measures to test the proposed AOSD method. The confusion matrix (CM) is described in Table 2. As known, it is used to test the performance of a classifier, including Accuracy, Specificity, and Sensitivity [54].

Table 2. Confusion Matrix.

	Predicted Class	
Actual class	**Positive**	**Negative**
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

- Average accuracy (AVG_{Acc}): This measure is the rate of correctly data classification, and it is computed as [22,55–57]:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{24}$$

Each method is performed 30 times ($N_r = 30$); thus, the AVG_{Acc} is computed as:

$$AVG_{Acc} = \frac{1}{N_r} \sum_{k=1}^{N_r} Acc_{Best}^k \tag{25}$$

- Average fitness value (AVG_{Fit}): it is used to assess the performance of an applied algorithm, and it puts the error rate of classification and reducing the selection ratio as the following equation [22,55–57]:

$$AVG_{Fit} = \frac{1}{N_r} \sum_{k=1}^{N_r} Fit_{Best}^k \tag{26}$$

- Average number of the selected features $(AVG_{|BX_{Best}|})$: This metric is applied to compute the ability of the applied method to reduce the number of features overall number of runs, and it is computed as [22,55–57]:

$$AVG_{|BX_{Best}|} = \frac{1}{N_r} \sum_{k=1}^{N_r} \left|BX_{Best}^k\right| \tag{27}$$

in which $|BX_{Best}^k|$ represents the cardinality of selected features for kth run.

- Average computation time (AVG_{Time}): This measure is used to compute the average of CPU time(s), as in the following equation [22,55–57]:

$$AVG_{Time} = \frac{1}{N_r} \sum_{k=1}^{N_r} Time_{Best}^k \tag{28}$$

- Standard deviation (STD): STD is employed to assess the quality of each applied method and analyze the achieved results in different runs. It is computed as [22,55–57]:

$$STD_Y = \sqrt{\frac{1}{N_r} \sum_{k=1}^{N_r} \left(Y_{Best}^k - AVG_Y\right)^2} \tag{29}$$

(Note: STD_Y is computed for each metric: Accuracy, Fitness, Time, Number of selected features, Sensitivity, and Specificity.

5.3. Comparisons

In this section, the developed AOSD is evaluated over eighteen well-known datasets. The evaluation uses ten algorithms to compare the performance of the developed AOSD, namely AOS, AOA, MPA, MRFO, HHO, HGSO, WOA, bGWO, GA, and BPSO. Six measures are used, called maximum fitness function (MAX), the average of the fitness function, minimum fitness function (MIN), accuracy (Acc), and standard deviation (St). The values obtained by the compared algorithms are recorded in Tables 3–9 where the smaller value in the tables means the better results, except for Table 8, where the higher value is the best; therefore, all best values in the tables are in boldface.

The results of the fitness function values are listed in Table 3 and the smaller fitness value means the better results. This table contains the average of the fitness function for the developed AOSD method and the comparison methods for all datasets. From these results, the AOSD got the best results in 6 out of 18 datasets (i.e., S2, S4, S7, S9, S15, and S16); therefore, it got the first rank. The AOA obtained the best values in three datasets (i.e., S3, S8, and S18), and it was ranked second, followed by MPA, MRFO, BPSO, and HHO, respectively; the GA showed the worst results. With the use of the average, it is possible to analyze the behavior of the results provided by the algorithms in the experiments. In terms of optimization, the fitness standard helps identify a typical value in the experiments for each dataset. Figure 2 shows the performance of the AOSD using the average of the fitness functions.

Table 3. Average of the fitness values for FS approaches.

	AOSD	**AOS**	**AOA**	**MPA**	**MRFO**	**HHO**	**HGSO**	**WOA**	**bGWO**	**GA**	**BPSO**
S1	0.07787	0.07097	0.06313	0.07245	0.06836	**0.05214**	0.06006	0.07381	0.06789	0.10176	0.05905
S2	**0.03779**	0.04860	0.03979	0.04004	0.04558	0.05288	0.09111	0.06994	0.08085	0.12798	0.04738
S3	0.02841	0.03401	**0.02707**	0.03773	0.04760	0.04950	0.03019	0.07382	0.10753	0.10184	0.06191
S4	**0.04258**	0.07248	0.04769	0.05013	0.05393	0.07699	0.08515	0.15975	0.14146	0.19231	0.06569
S5	0.20225	0.29336	0.24958	0.26147	0.21919	0.23719	0.29019	0.21696	**0.19977**	0.33061	0.24169
S6	0.15154	0.19231	0.12897	0.12966	0.16470	**0.11368**	0.13009	0.21598	0.20376	0.19581	0.17427
S7	**0.03450**	0.06409	0.04345	0.08089	0.05246	0.09376	0.10571	0.09927	0.08166	0.12058	0.08644
S8	0.08291	0.07925	**0.06224**	0.06559	0.07450	0.07771	0.09491	0.09712	0.09547	0.11478	0.07716
S9	**0.06864**	0.10067	0.07972	0.09194	0.13178	0.12578	0.10091	0.12868	0.15640	0.18178	0.14111
S10	0.07080	0.06836	0.04769	0.04974	0.05116	0.06224	0.07679	0.11761	0.09975	0.11787	**0.04718**
S11	0.10098	0.05470	0.16158	0.07389	**0.01497**	0.05145	0.02392	0.04076	0.04886	0.20224	0.04142
S12	0.07500	0.08233	0.09371	0.07768	0.08346	0.07990	0.08322	**0.06727**	0.09649	0.08905	0.10349
S13	0.14697	0.12485	0.14667	0.14556	0.15566	**0.10010**	0.11152	0.23364	0.23525	0.20475	0.12909
S14	0.24469	0.22896	0.20556	**0.19815**	0.23154	0.22999	0.22279	0.25719	0.25477	0.22787	0.22899
S15	**0.02050**	0.04950	0.04325	0.06358	0.03783	0.06042	0.04650	0.04567	0.05325	0.10517	0.02217
S16	**0.24680**	0.28778	0.27168	0.26385	0.27501	0.29047	0.29423	0.29956	0.30259	0.30750	0.26581
S17	0.03885	0.04462	0.03385	0.03846	0.03949	0.04936	0.04218	0.06987	0.05705	0.08782	**0.03333**
S18	0.01750	0.04125	**0.01000**	0.01667	0.03917	0.01625	0.03875	0.05327	0.06595	0.05625	0.02333

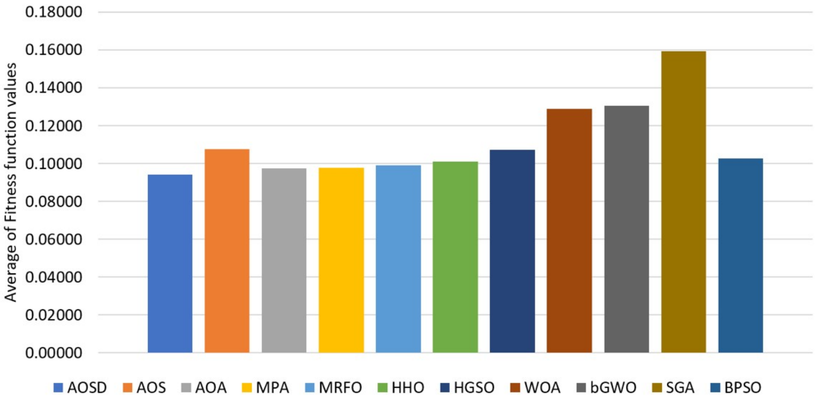

Figure 2. Average of the fitness functions' values.

Table 4 shows the results of the standard division for all methods. The Std here is used to verify the dispersion of the results along with the experiments with different datasets. A low value in Std represents low dispersion, which means the algorithm is more stable along with the experiments. The AOSD showed good stability compared to the other methods, and it achieved the lowest Std value in 6 out of 18 datasets (i.e., S6, S7, S9, S13, S17, and S18). It was ranked first followed by BPSO and it showed good stability in S14, S8, S10, S11, and S15 datasets. In addition, the AOA, MPA, and MRFO also showed good resilience. The bGWO and WOA showed the worst Std values in this measure.

Table 4. Standard deviation of fitness values for FS approaches.

	AOSD	AOS	AOA	MPA	MRFO	HHO	HGSO	WOA	bGWO	GA	BPSO
S1	0.00981	0.00755	0.00131	0.00130	0.00664	0.00344	0.00181	0.01129	0.00613	0.01204	**0.00000**
S2	0.01058	0.00773	0.01133	0.00678	0.00678	**0.00527**	0.00581	0.01152	0.01138	0.00665	0.00581
S3	0.01863	**0.00183**	0.00293	0.00621	0.00872	0.00510	0.00187	0.01534	0.01959	0.01187	0.00504
S4	0.01611	0.02493	**0.00344**	0.00772	0.00651	0.06475	0.03297	0.09634	0.08102	0.08227	0.06514
S5	0.01149	0.01593	0.01551	0.02358	**0.00000**	0.00000	0.04174	0.02620	0.00487	0.01642	0.00000
S6	**0.00116**	0.02502	0.01463	0.01521	0.02765	0.03781	0.01825	0.02253	0.03192	0.01045	0.00748
S7	**0.00232**	0.01001	0.01858	0.01097	0.01648	0.01745	0.01646	0.01774	0.00944	0.00910	0.00920
S8	0.01807	0.01065	0.00742	0.00796	0.00498	0.01075	0.00795	0.01500	0.01414	0.01060	**0.00374**
S9	**0.00524**	0.03655	0.00977	0.02158	0.00749	0.02783	0.01341	0.05968	0.02549	0.02710	0.03738
S10	0.00350	0.02887	0.00344	0.00397	0.00689	0.01409	0.02357	0.04101	0.04160	0.04253	**0.00271**
S11	0.03558	0.02912	0.05446	0.01485	0.00753	0.04051	0.01160	0.03144	0.02659	0.00234	**0.00153**
S12	0.01437	0.01961	0.01301	0.01644	0.01392	0.01418	0.01696	0.01282	0.02020	**0.01060**	0.01996
S13	**0.00324**	0.02430	0.00761	0.00969	0.02229	0.01815	0.01516	0.00716	0.01772	0.01193	0.00869
S14	0.00204	0.00784	**0.00000**	0.00061	0.00090	0.01094	0.00929	0.01796	0.01766	0.02312	0.00000
S15	0.00892	0.01092	0.00873	0.00704	0.00730	0.00868	0.00529	0.01677	0.02007	0.02316	**0.00326**
S16	0.01690	0.02056	0.01420	0.01350	0.01110	0.00957	0.01028	0.01740	0.01149	**0.00855**	0.00875
S17	**0.00318**	0.00644	0.00421	0.00581	0.01002	0.00895	0.00966	0.01386	0.01171	0.01607	0.00475
S18	**0.00280**	0.00948	0.00839	0.00386	0.01168	0.00939	0.00350	0.01187	0.01567	0.00747	0.00286

In addition, the best fitness values are listed in Table 5. By analyzing the best values obtained by the compared algorithms in all the runs for each dataset, the idea is to see which algorithm can provide the best solution in the best case (or in the best run). This table shows that the AOSD showed the best Min values in 33% of all datasets; it obtained the best Min results in S14, S2, S4, S7, S8, S16, and S18. The HHO and MPA received the best values in this measure in two datasets for each, ranking second and third, respectively. All methods obtained the same results in S4 datasets except for HGSO and GA. The GA recorded the worst performance in this measure.

Table 5. Results of the best fitness function values for FS approaches.

	AOSD	AOS	AOA	MPA	MRFO	HHO	HGSO	WOA	bGWO	GA	BPSO
S1	**0.04016**	0.06548	0.06254	0.07190	0.06548	0.04968	0.05905	0.05905	0.05730	0.08302	0.05905
S2	**0.02702**	0.03912	0.03123	0.02912	0.03246	0.04368	0.07860	0.04912	0.06070	0.11070	0.03789
S3	0.06013	0.03319	**0.02500**	0.03125	0.03944	0.04353	0.02909	0.05603	0.06638	0.07694	0.05819
S4	**0.04615**	**0.04615**	**0.04615**	**0.04615**	**0.04615**	**0.04615**	0.05385	**0.04615**	**0.04615**	0.06154	**0.04615**
S5	0.21154	0.26927	0.22877	0.23719	0.21919	0.23719	0.24169	0.21019	**0.19669**	0.30323	0.24169
S6	0.23590	0.15513	0.12179	0.11282	0.12308	**0.07436**	0.10641	0.17308	0.16282	0.16923	0.16410
S7	**0.01621**	0.05273	0.01765	0.06156	0.02738	0.05953	0.06541	0.07423	0.06744	0.10476	0.07332
S8	**0.04734**	0.06845	0.05292	0.05028	0.06424	0.05717	0.08094	0.06595	0.06832	0.10031	0.07120
S9	0.05889	0.07444	0.06992	0.06992	0.11556	0.09889	0.06239	**0.04711**	0.10651	0.13222	0.06889
S10	0.05385	**0.04615**	**0.04615**	**0.04615**	**0.04615**	**0.04615**	0.05385	0.06154	0.05385	0.06154	**0.04615**
S11	0.06646	**0.00277**	0.09746	0.02215	0.00523	0.00369	0.01077	0.00308	0.02031	0.19815	0.03815
S12	0.06143	0.06643	0.07619	**0.04667**	0.05976	0.06143	0.05643	0.04810	0.06476	0.07500	0.07143
S13	0.11212	0.10455	0.13485	0.12424	0.09848	**0.06515**	0.08939	0.21818	0.19394	0.17727	0.11515
S14	0.21024	0.21493	0.20556	**0.19792**	0.23073	0.22135	0.21788	0.23542	0.23073	0.21198	0.22899
S15	0.05125	0.04000	0.03375	0.05500	0.02500	0.04875	0.03750	0.03625	0.03375	0.06250	**0.01875**
S16	**0.22722**	0.25690	0.25740	0.23800	0.24820	0.27790	0.27580	0.27300	0.28470	0.29510	0.24930
S17	0.04615	0.03846	0.03077	0.03077	**0.02308**	0.03846	0.03077	0.04615	0.03846	0.06923	0.03077
S18	**0.00250**	0.03125	0.00625	0.01250	0.02500	0.00625	0.03125	0.03750	0.04375	0.04375	0.01875

In terms of the worst results of the fitness values, Table 6 shows these results. The study of the worst values in the results of compared algorithms helps to verify that even in the worst case, some algorithms provide reasonable solutions. Besides, it also permits one to see which algorithm is the worst in the worst case. The developed AOSD showed good results compared to other methods and achieved the best results in 7 out of 18 datasets (i.e., S14, S3, S9, S12, S13, S16, and S18). It showed competitive results in the other datasets. The AOA achieved the second rank by obtaining the best results in six datasets (i.e., S4, S6, S7, S8, S10, and S17), followed by AOA and MPA. The other compared methods were ordered as MRFO, BPSO, AOS, HGSO, HHO, bGWO, WOA, and GA in this sequence.

Table 6. Results of the worst fitness values' results for FS approaches.

	AOSD	AOS	AOA	MPA	MRFO	HHO	HGSO	WOA	bGWO	GA	BPSO
S1	**0.04944**	0.08008	0.06548	0.07659	0.08944	0.05905	0.06373	0.09762	0.08183	0.12278	0.05905
S2	0.08316	0.05702	0.05702	**0.05368**	0.06035	0.06158	0.10228	0.08561	0.10105	0.13649	0.05912
S3	**0.03058**	0.03728	0.03125	0.04784	0.07694	0.06228	0.03319	0.10366	0.14310	0.13297	0.07694
S4	0.07404	0.09754	**0.05385**	0.07504	0.06604	0.30208	0.15473	0.28219	0.23719	0.30962	0.30077
S5	0.28077	0.31092	0.27042	0.29496	0.21919	0.23719	0.33342	0.31165	**0.21208**	0.35592	0.24169
S6	0.26282	0.21282	**0.15513**	**0.15513**	0.21154	0.16282	0.15769	0.25128	0.26923	0.20897	0.18077
S7	0.09097	0.08012	**0.06653**	0.10750	0.08509	0.12697	0.12991	0.12788	0.09959	0.13894	0.10456
S8	0.10474	0.09198	**0.06990**	0.07545	0.08214	0.10333	0.10505	0.11599	0.12151	0.13403	0.08375
S9	**0.06667**	0.16222	0.09215	0.12889	0.14556	0.20889	0.11984	0.24667	0.20516	0.22778	0.18889
S10	0.06423	0.11873	**0.05385**	**0.05385**	0.07054	0.09173	0.12192	0.18362	0.17854	0.21062	**0.05385**
S11	0.14246	0.06952	0.24838	0.08246	**0.03385**	0.13754	0.05415	0.08523	0.09046	0.20646	0.04431
S12	**0.08119**	0.10619	0.11262	0.10286	0.10762	0.11262	0.11095	0.08667	0.12429	0.10810	0.13905
S13	**0.12494**	0.16667	0.15303	0.16515	0.17121	0.12879	0.14394	0.23788	0.26515	0.22273	0.14394
S14	0.26476	0.23247	0.20556	**0.19965**	0.23247	0.26354	0.24184	0.29167	0.29236	0.27934	0.22899
S15	0.06375	0.06750	0.05750	0.07625	0.05000	0.08000	0.05500	0.09500	0.09750	0.14375	**0.02750**
S16	**0.27200**	0.31280	0.29540	0.28820	0.29420	0.31140	0.31520	0.32290	0.32150	0.32150	0.27920
S17	0.07692	0.05385	**0.03846**	0.04615	0.05385	0.07115	0.05385	0.08654	0.07115	0.11923	0.04615
S18	**0.01875**	0.05625	0.02500	0.02500	0.05625	0.03125	0.04375	0.08036	0.08661	0.06875	0.02500

Moreover, the selected features number by each method is recorded in Table 7. In this measure, the best method tries to choose the lowest features and achieve high accuracy results. As shown in Table 7, the AOSD reached the second rank by obtaining the lowest features number in 7 out of 18 datasets, whereas the first rank was received by the WOA method, it selected the lowest number of features in 8 datasets. The third rank was obtained by HGSO followed by MRFO, HHO, AOS, bGWO, MPA, AOA, and BPSO; whereas, the GA showed the worst performance in all datasets.

Table 7. Selected features numbers for FS approaches.

	AOSD	AOS	AOA	MPA	MRFO	HHO	HGSO	WOA	bGWO	GA	BPSO
S1	3	**2**	3	**2**	**2**	4	3	**2**	3	3	3
S2	**2**	3	5	4	4	6	5	4	3	18	7
S3	**2**	**2**	3	4	3	5	**2**	**2**	**2**	9	4
S4	6	6	6	6	6	**3**	7	8	8	8	4
S5	11	4	3	5	6	4	5	5	6	7	**1**
S6	**3**	5	5	5	4	4	7	7	7	9	4
S7	3	4	6	6	**2**	3	4	**2**	6	24	8
S8	9	13	12	10	13	13	11	**4**	17	27	11
S9	4	5	7	7	10	**3**	**3**	**3**	5	13	6
S10	**5**	6	6	6	6	6	7	8	7	8	6
S11	21	**9**	67	24	17	12	35	10	66	254	124
S12	24	27	17	21	13	13	16	**9**	15	45	22
S13	9	8	**4**	6	**4**	5	**4**	5	**4**	14	6
S14	4	5	5	6	6	5	4	**3**	**3**	5	5
S15	**2**	3	4	3	4	**2**	3	3	3	9	5
S16	16	16	9	13	17	11	**7**	12	11	32	15
S17	**3**	5	4	4	**3**	5	**3**	**3**	**3**	8	4
S18	**2**	5	3	3	4	5	5	6	6	7	3

In addition, Table 8 illustrates the results of all compared methods in terms of classification accuracy. The use of accuracy permits the evaluation of the correct predicted data points out of all the data points. By such interpretation, this study permits identifying if an algorithm is outstanding in classification. The accuracy is essential in multiple real applications; for that reason, its use is mandatory. In this measure, the developed AOSD showed good results in 17% of the datasets; therefore, it was able to classify these datasets with high accuracy compared to other methods, and it obtained the same accuracy with the other methods in 22% of the datasets. The MRFO was ranked second, followed by MPA, AOA, BPSO, HHO, AOS, HHO, HGSO, EFO, and bGWO whereas, the lowest accuracy was shown by the WOA method. Figure 3 illustrates the performance of the AOSD based on the average classification accuracy for all datasets.

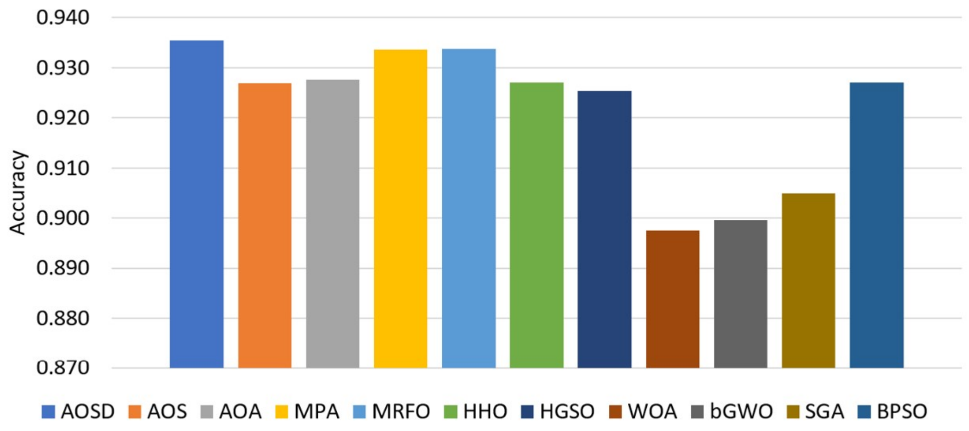

Figure 3. Average of the classification accuracy among tested datasets.

Table 8. Accuracy results for FS approaches.

	AOSD	AOS	AOA	MPA	MRFO	HHO	HGSO	WOA	bGWO	GA	BPSO
S1	0.9714	0.9557	0.9471	0.9557	0.9619	0.9643	**0.9752**	0.9476	0.9567	0.9462	0.9714
S2	0.9737	0.9719	0.9825	0.9807	0.9760	0.9731	0.9333	0.9433	0.9415	0.9351	**0.9854**
S3	0.9655	0.9747	**0.9977**	0.9923	0.9716	0.9640	0.9854	0.9448	0.9157	0.9609	0.9724
S4	**1.0000**	0.9810	**1.0000**	0.9990	0.9993	0.9720	0.9743	0.8960	0.8987	0.8667	0.9800
S5	0.7750	0.7390	0.7620	0.7437	0.7650	0.7450	0.7260	0.7703	**0.7900**	0.7130	0.7400
S6	0.8148	0.8444	0.8926	0.9049	0.8654	**0.9062**	0.8914	0.7988	0.8272	0.8753	0.8531
S7	**0.9859**	0.9549	0.9831	0.9408	0.9681	0.9268	0.9117	0.9174	0.9380	0.9577	0.9399
S8	0.9719	0.9675	0.9734	0.9720	**0.9779**	0.9725	0.9495	0.9507	0.9577	0.9616	0.9643
S9	**0.9667**	0.9400	0.9657	0.9542	0.9289	0.9133	0.9319	0.8821	0.8756	0.8844	0.8889
S10	**1.0000**	0.9890	**1.0000**	**1.0000**	0.9990	0.9947	0.9853	0.9497	0.9627	0.9477	**1.0000**
S11	0.9333	0.9457	0.8454	0.9378	**1.0000**	0.9556	**1.0000**	0.9686	0.9822	0.8667	**1.0000**
S12	0.9762	0.9667	0.9381	0.9683	0.9571	0.9571	0.9556	0.9825	0.9381	**0.9937**	0.9381
S13	0.9259	0.9148	0.8704	0.8790	0.8506	**0.9309**	0.9148	0.7593	0.7716	0.8580	0.8963
S14	0.8281	0.8271	0.8333	**0.8556**	0.8226	0.8177	0.8101	0.7809	0.7819	0.8250	0.8073
S15	**1.0000**	0.9700	0.9700	0.9622	0.9978	0.9667	0.9844	0.9622	0.9700	0.9600	0.9967
S16	0.7500	0.7408	0.7348	0.7589	**0.7665**	0.7287	0.7283	0.7283	0.7186	0.7533	0.7528
S17	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9981	0.9981	0.9759	0.9833	0.9833	**1.0000**
S18	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9968	0.9841	**1.0000**	**1.0000**

Moreover, Table 9 records the statistical results of the Friedman rank test to rank all methods using both the classification accuracy and the fitness function values. This test studies the statistical differences between the algorithms considering the results obtained for the 30 independent runs in all datasets. From Table 9, we can see that developed AOSD achieved the first rank in classification accuracy, followed by MRFO, MPA, AOA, BPSO, AOS, and HHO. The WOA was ranked last. Whereas, in the fitness function, the AOSD showed an excellent rank and was came second after the AOA with slight deference, followed by MPA, BPSO, MRFO, HHO, and HGSO. The GA was ranked last. From these results, we can notice that the AOSD showed the best results in accuracy, whereas it showed the second-best in the fitness function. These results indicate the superiority of the AOSD due to the fact that the classification accuracy measure can be more important than the fitness function value in solving classification problems.

Table 9. Friedman rank test results for all methods.

	AOSD	AOS	AOA	MPA	MRFO	HHO	HGSO	WOA	bGWO	GA	BPSO
Fitness	4.11	6.06	**3.50**	4.33	5.11	5.33	5.94	8.00	8.50	10.11	5.00
Accuracy	**8.17**	6.31	7.42	7.56	7.67	6.06	5.42	3.11	3.47	4.14	6.69

In general, the aforementioned results show that the developed AOSD method showed a noticeable enhancement in solving classification problems by selecting the essential features. The DOL approach improves the performance of the AOS by increasing the ability of the AOS to discover the search domain and save it from getting stuck in a local point.

Furthermore, the results of the AOSD showed its advantages over the compared algorithms by achieving the best fitness functions values in 33% of all datasets, whereas the second-rank HHO method achieved the best values in 16% of the datasets. This result was also observed in the rest of the measures. In addition, if we compare the differences between the proposed method AOSD and its original version AOS, in the accuracy measure, we can see that the proposed method outperformed the original version in 16 out of 18 datasets and showed similar accuracies in the other two cases. Besides, the proposed method is ranked first according to the statistical test (i.e., Friedman test) for accuracy measure, which indicates a significant difference between the AOSD and the compared method at p-value equals 0.05. Based on the results, we will work in the future to increase the performance of the proposed method by improving its exploitation phase and applying it in different optimization problems.

6. Conclusions

This paper developed a modified Atomic Orbit Search (AOS) and used it as a feature selection (FS) approach. The modification has been performed using dynamic opposite-based learning (DOL) to enhance the exploration and diversity of solutions. This leads to improving the convergence rate to explore the feasible region that contains the optima solution (relevant features). To justify the performance of the AOSD as an FS approach, a set of twenty datasets collected from different real-life applications has been used. In addition, the results of AOSD have been compared with other well-known FS approaches based on MH techniques such as AOS, APA, MPA, MRFO, HHO, HGSO, WOA, GWO, GA, and PSO. The obtained results concluded that the developed AOSD provided higher efficiency than other FS approaches.

Besides the obtained results, the developed AOSD can be extended to other real-life applications, including medical images, superpixel-Based clustering, Internet of things (IoT), security, and other fields.

Author Contributions: Conceptualization, D.Y.; Data curation, M.A.E., L.A., A.A.E. and S.L.; Formal analysis, R.A.I.; Funding acquisition, A.A.E.; Investigation, M.A.E., L.A., D.Y., M.A.A.A.-Q. and M.H.N.-S.; Methodology, D.O., S.L. and R.A.I.; Software, M.A.E., A.A.E. and R.A.I.; Supervision, M.H.N.-S.; Validation, M.A.A.A.-Q. and M.H.N.-S.; Visualization, D.Y.; Writing, D.O., M.A.A.A.-Q., A.A.E. and S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: This work is supported by the Hubei Provincial Science and Technology Major Project of China under Grant No. 2020AEA011 and the Key Research & Development Plan of Hubei Province of China under Grant No. 2020BAB100 and the project of Science,Technology and Innovation Commission of Shenzhen Municipality of China under Grant No. JCYJ20210324120002006.

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

References

1. Tubishat, M.; Idris, N.; Shuib, L.; Abushariah, M.A.; Mirjalili, S. Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection. *Expert Syst. Appl.* **2020**, *145*, 113122. [CrossRef]
2. Shao, Z.; Wu, W.; Li, D. Spatio-temporal-spectral observation model for urban remote sensing. *Geo-Spat. Inf. Sci.* **2021**, *17*, 372–386. [CrossRef]
3. Ibrahim, R.A.; Ewees, A.A.; Oliva, D.; Abd Elaziz, M.; Lu, S. Improved salp swarm algorithm based on particle swarm optimization for feature selection. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 3155–3169. [CrossRef]
4. Zebari, R.; Abdulazeez, A.; Zeebaree, D.; Zebari, D.; Saeed, J. A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. *J. Appl. Sci. Technol. Trends* **2020**, *1*, 56–70. [CrossRef]
5. Venkatesh, B.; Anuradha, J. A review of feature selection and its methods. *Cybern. Inf. Technol.* **2019**, *19*, 3–26. [CrossRef]
6. Shao, Z.; Sumari, N.S.; Portnov, A.; Ujoh, F.; Musakwa, W.; Mandela, P.J. Urban sprawl and its impact on sustainable urban development: A combination of remote sensing and social media data. *Geo-Spat. Inf. Sci.* **2021**, *24*, 241–255. [CrossRef]
7. Abdel-Basset, M.; Ding, W.; El-Shahat, D. A hybrid Harris Hawks optimization algorithm with simulated annealing for feature selection. *Artif. Intell. Rev.* **2021**, *54*, 593–637. [CrossRef]
8. El-Hasnony, I.M.; Barakat, S.I.; Elhoseny, M.; Mostafa, R.R. Improved feature selection model for big data analytics. *IEEE Access* **2020**, *8*, 66989–67004. [CrossRef]
9. Deng, X.; Li, Y.; Weng, J.; Zhang, J. Feature selection for text classification: A review. *Multimed. Tools Appl.* **2019**, *78*, 3. [CrossRef]
10. Ewees, A.A.; Abualigah, L.; Yousri, D.; Algamal, 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**, 31, 1–15.
11. Alex, S.B.; Mary, L.; Babu, B.P. Attention and Feature Selection for Automatic Speech Emotion Recognition Using Utterance and Syllable-Level Prosodic Features. *Circuits, Syst. Signal Process.* **2020**, *39*, 11. [CrossRef]

12. Benazzouz, A.; Guilal, R.; Amirouche, F.; Slimane, Z.E.H. 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.
13. Al-qaness, M.A. Device-free human micro-activity recognition method using WiFi signals. *Geo-Spat. Inf. Sci.* **2019**, *22*, 128–137. [CrossRef]
14. Yousri, D.; Abd Elaziz, M.; Abualigah, L.; Oliva, D.; Al-Qaness, M.A.; Ewees, A.A. COVID-19 X-ray images classification based on enhanced fractional-order cuckoo search optimizer using heavy-tailed distributions. *Appl. Soft Comput.* **2021**, *101*, 107052. [CrossRef]
15. Nadimi-Shahraki, M.H.; Banaie-Dezfouli, M.; Zamani, H.; Taghian, S.; Mirjalili, S. B-MFO: A Binary Moth-Flame Optimization for Feature Selection from Medical Datasets. *Computers* **2021**, *10*, 136. [CrossRef]
16. Hancer, E. A new multi-objective differential evolution approach for simultaneous clustering and feature selection. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103307. [CrossRef]
17. Amini, F.; Hu, G. A two-layer feature selection method using genetic algorithm and elastic net. *Expert Syst. Appl.* **2021**, *166*, 114072. [CrossRef]
18. Song, X.f.; Zhang, Y.; Gong, D.w.; Sun, X.y. Feature selection using bare-bones particle swarm optimization with mutual information. *Pattern Recognit.* **2021**, *112*, 107804. [CrossRef]
19. Tubishat, M.; Ja'afar, S.; Alswaitti, M.; Mirjalili, S.; Idris, N.; Ismail, M.A.; Omar, M.S. Dynamic salp swarm algorithm for feature selection. *Expert Syst. Appl.* **2021**, *164*, 113873. [CrossRef]
20. Sathiyabhama, B.; Kumar, S.U.; Jayanthi, J.; Sathiya, T.; Ilavarasi, A.; 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]
21. 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]
22. Ewees, A.A.; Abd El Aziz, M.; Hassanien, A.E. Chaotic multi-verse optimizer-based feature selection. *Neural Comput. Appl.* **2019**, *31*, 991–1006. [CrossRef]
23. Abualigah, L.M.Q. *Feature Selection and Enhanced Krill Herd Algorithm for Text Document Clustering*; Springer: Berlin/Heidelberg, Germany, 2019.
24. Abd Elaziz, M.; Ewees, A.A.; Ibrahim, R.A.; Lu, S. Opposition-based moth-flame optimization improved by differential evolution for feature selection. *Math. Comput. Simul.* **2020**, *168*, 48–75. [CrossRef]
25. Neggaz, N.; Houssein, E.H.; Hussain, K. An efficient henry gas solubility optimization for feature selection. *Expert Syst. Appl.* **2020**, *152*, 113364. [CrossRef]
26. Helmi, A.M.; Al-qaness, M.A.; Dahou, A.; Damaševičius, R.; Krilavičius, T.; Elaziz, M.A. A Novel Hybrid Gradient-Based Optimizer and Grey Wolf Optimizer Feature Selection Method for Human Activity Recognition Using Smartphone Sensors. *Entropy* **2021**, *23*, 1065. [CrossRef]
27. 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]
28. Azizi, M. Atomic orbital search: A novel metaheuristic algorithm. *Appl. Math. Model.* **2021**, *93*, 657–683. [CrossRef]
29. Azizi, M.; Talatahari, S.; Giaralis, A. Optimization of Engineering Design Problems Using Atomic Orbital Search Algorithm. *IEEE Access* **2021**, *9*, 102497–102519. [CrossRef]
30. Dong, H.; Xu, Y.; Li, X.; Yang, Z.; Zou, C. An improved antlion optimizer with dynamic random walk and dynamic opposite learning. *Knowl.-Based Syst.* **2021**, *216*, 106752. [CrossRef]
31. Zhang, L.; Hu, T.; Yang, Z.; Yang, D.; Zhang, J. Elite and dynamic opposite learning enhanced sine cosine algorithm for application to plat-fin heat exchangers design problem. *Neural Comput. Appl.* **2021**, 1–14. [CrossRef]
32. Feng, Y.; Liu, M.; Zhang, Y.; Wang, J. A Dynamic Opposite Learning Assisted Grasshopper Optimization Algorithm for the Flexible JobScheduling Problem. *Complexity* **2020**, *2020*, 1–19.
33. 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]
34. Sharma, M.; Kaur, P. A Comprehensive Analysis of Nature-Inspired Meta-Heuristic Techniques for Feature Selection Problem. *Arch. Comput. Methods Eng.* **2021**, *28*, 1103–1127. [CrossRef]
35. Rostami, M.; Berahmand, K.; Nasiri, E.; Forouzande, S. Review of swarm intelligence-based feature selection methods. *Eng. Appl. Artif. Intell.* **2021**, *100*, 104210. [CrossRef]
36. Nguyen, B.H.; Xue, B.; Zhang, M. A survey on swarm intelligence approaches to feature selection in data mining. *Swarm Evol. Comput.* **2020**, *54*, 100663. [CrossRef]
37. Hu, P.; Pan, J.S.; Chu, S.C. Improved binary grey wolf optimizer and its application for feature selection. *Knowl.-Based Syst.* **2020**, *195*, 105746. [CrossRef]
38. Hu, Y.; Zhang, Y.; Gong, D. Multiobjective particle swarm optimization for feature selection with fuzzy cost. *IEEE Trans. Cybern.* **2020**, *51*, 874–888. [CrossRef]
39. Gao, Y.; Zhou, Y.; Luo, Q. An efficient binary equilibrium optimizer algorithm for feature selection. *IEEE Access* **2020**, *8*, 140936–140963. [CrossRef]

40. Al-Tashi, Q.; Abdulkadir, S.J.; Rais, H.M.; Mirjalili, S.; Alhussian, H.; Ragab, M.G.; Alqushaibi, A. Binary multi-objective grey wolf optimizer for feature selection in classification. *IEEE Access* **2020**, *8*, 106247–106263. [CrossRef]
41. Alazzam, H.; Sharieh, A.; Sabri, K.E. A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer. *Expert Syst. Appl.* **2020**, *148*, 113249. [CrossRef]
42. Zhang, Y.; Gong, D.W.; Gao, X.z.; Tian, T.; Sun, X.Y. Binary differential evolution with self-learning for multi-objective feature selection. *Inf. Sci.* **2020**, *507*, 67–85. [CrossRef]
43. 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]
44. Hammouri, A.I.; Mafarja, M.; Al-Betar, M.A.; Awadallah, M.A.; Abu-Doush, I. An improved dragonfly algorithm for feature selection. *Knowl.-Based Syst.* **2020**, *203*, 106131. [CrossRef]
45. Zhang, Y.; Liu, R.; Wang, X.; Chen, H.; Li, C. Boosted binary Harris hawks optimizer and feature selection. *Eng. Comput.* **2020**, *37*, 3741–3770. [CrossRef]
46. Sahlol, A.T.; Yousri, D.; Ewees, A.A.; Al-Qaness, M.A.; Damasevicius, R.; Abd Elaziz, M. COVID-19 image classification using deep features and fractional-order marine predators algorithm. *Sci. Rep.* **2020**, *10*, 15364. [CrossRef] [PubMed]
47. Abdel-Basset, M.; Mohamed, R.; Chakrabortty, R.K.; Ryan, M.J.; Mirjalili, S. An efficient binary slime mould algorithm integrated with a novel attacking-feeding strategy for feature selection. *Comput. Ind. Eng.* **2021**, *153*, 107078. [CrossRef]
48. Tizhoosh, H.R. Opposition-based learning: A new scheme for machine intelligence. In Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), Vienna, Austria, 28–30 November 2005; Volume 1, pp. 695–701.
49. Houssein, E.H.; Hussain, K.; Abualigah, L.; Abd Elaziz, M.; Alomoush, W.; Dhiman, G.; Djenouri, Y.; Cuevas, E. An improved opposition-based marine predators algorithm for global optimization and multilevel thresholding image segmentation. *Knowl.-Based Syst.* **2021**, *229*, 107348. [CrossRef]
50. Frank, A. UCI Machine Learning Repository. Available online: http://archive.ics.uci.edu/ml (accessed on 1 August 2020).
51. Abualigah, L.; Diabat, A.; Mirjalili, S.; Abd Elaziz, M.; Gandomi, A.H. The arithmetic optimization algorithm. *Comput. Methods Appl. Mech. Eng.* **2021**, *376*, 113609. [CrossRef]
52. Abd Elaziz, M.; Yousri, D.; Al-qaness, M.A.; AbdelAty, A.M.; Radwan, A.G.; Ewees, A.A. A Grunwald–Letnikov based Manta ray foraging optimizer for global optimization and image segmentation. *Eng. Appl. Artif. Intell.* **2021**, *98*, 104105. [CrossRef]
53. Ibrahim, R.A.; Abd Elaziz, M.; Lu, S. Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization. *Expert Syst. Appl.* **2018**, *108*, 1–27. [CrossRef]
54. Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. [CrossRef]
55. Ferri, C.; Hernández-Orallo, J.; Modroiu, R. An experimental comparison of performance measures for classification. *Pattern Recognit. Lett.* **2009**, *30*, 27–38. [CrossRef]
56. Elaziz, M.A.; Hosny, K.M.; Salah, A.; Darwish, M.M.; Lu, S.; Sahlol, A.T. New machine learning method for image-based diagnosis of COVID-19. *PLoS ONE* **2020**, *15*, e0235187. [CrossRef]
57. Neggaz, N.; Ewees, A.A.; Abd Elaziz, M.; Mafarja, M. Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection. *Expert Syst. Appl.* **2020**, *145*, 113103. [CrossRef]