A Feature Selection Approach Based on Archimedes' Optimization Algorithm for Optimal Data Classification

Lahbib Khrissi[1]*, Nabil El Akkad[2], Hassan Satori[1], Khalid Satori[1]

[1] LISAC, Department of Computer Science, Faculty of Science, Dhar El Mahraz, USMBA, Fez (Morocco)
[2] Laboratory of Engineering, Systems and Applications, ENSA of Fez, USMBA, Fez (Morocco)

* Corresponding author: lahbib.khrissi@usmba.ac.ma

Received 25 February 2022 | Accepted 2 December 2022 | Published 19 January 2023

unir LA UNIVERSIDAD EN INTERNET

Abstract

Feature selection is an active research area in data mining and machine learning, especially with the increase in the amount of numerical data. FS is a search strategy to find the best subset of features among a large number of subsets of features. Thus, FS is applied in most modern applications and in various domains, which requires the search for a powerful FS technique to process and classify high-dimensional data. In this paper, we propose a new technique for dimension reduction in feature selection. This approach is based on a recent metaheuristic called Archimedes' Optimization Algorithm (AOA) to select an optimal subset of features to improve the classification accuracy. The idea of the AOA is based on the steps of Archimedes' principle in physics. It explains the behavior of the force exerted when an object is partially or fully immersed in a fluid. AOA optimization maintains a balance between exploration and exploitation, keeping a population of solutions and studying a large area to find the best overall solution. In this study, AOA is exploited as a search technique to find an optimal feature subset that reduces the number of features to maximize classification accuracy. The K-nearest neighbor (K-NN) classifier was used to evaluate the classification performance of selected feature subsets. To demonstrate the superiority of the proposed method, 16 benchmark datasets from the UCI repository are used and also compared by well-known and recently introduced meta-heuristics in this context, such as: sine-cosine algorithm (SCA), whale optimization algorithm (WOA), butterfly optimization algorithm (BAO), and butterfly flame optimization algorithm (MFO). The results prove the effectiveness of the proposed algorithm over the other algorithms based on several performance measures used in this paper.

Keywords

Archimedes' Optimization Algorithm, Data Classification Feature Selection, KNN Classifier.

DOI: 10.9781/ijimai.2023.01.005

I. Introduction

Over recent years, storage capacity and information acquisition have become cheaper. This allows us to store all possible data related to people's needs. However, there are data that do not necessarily contain useful information to be extracted. With the exceptional increase in the amount of data that can be stored, the exploration of new methods that are able to process data automatically is necessary following a process of knowledge extraction from data. This process allows for the integration and collection of data, selection, cleaning and data processing, data analysis for the extraction of appropriate patterns and models, evaluation and interpretation of constructed models and consolidation of knowledge available for use.

Feature selection (FS) is a research process or technique used to select the most interesting, relevant, or informative features, variables, or measures of a given system in order to accomplish the task for which it was designed [1],[2]. In the field of machine learning and more particularly in classification, some irrelevant and/or redundant features, usually present in the training data, not only make learning more difficult, but also degrade the generalization performance of the training models [3],[4]. FS is a pre-processing step that plays an important role in data mining. It allows to represent a subset of data from a large data set and to eliminate redundant, irrelevant or noisy data. There are several advantages of attribute subset selection: It facilitates data visualization and provides a better understanding [5],[6]. It reduces the complexity of the training data which will lead to the reduction of the time of the learning algorithm. Another important factor is the reduction of the problem dimension, the improvement of the prediction performance and the understanding of the learning model [7],[8]. FS methods are applied to several applications and in various well-known fields, such as computer vision [9],[10], image processing [11]–[13] pattern recognition and machine learning [14]–[16].

Classification algorithms aim to identify the classes to which objects belong based on certain descriptive features. So, the main process of classification in machine learning is to train the classifier to accurately recognize patterns from given training samples and classify the test

Please cite this article as:
L. Khrissi, N. E. Akkad, H. Satori, K. Satori. A Feature Selection Approach Based on Archimedes' Optimization Algorithm for Optimal Data Classification, International Journal of Interactive Multimedia and Artificial Intelligence, vol. 9, no. 1, pp. 28-38, 2024, http://dx.doi.org/10.9781/ijimai.2023.01.005

samples with the trained classifier. The choice of a classification method for the treatment of the FS problem by metaheuristics has a great impact on the quality of the obtained solution. In the literature, there are several types of classifiers that have been used to assist data selection techniques [17] among those commonly used we mention: support vector machine (SVM), K-nearest neighbor (KNN), naive bay (NB), random forest (RF), artificial neural network (ANN) and others.

Metaheuristics (MH) include all algorithms based on the population concept that use selection and recombination to generate new points in the search space. MHs are widely used to solve complex optimization problems. They have been developed and updated for use in various domains, for example task scheduling in cloud computing [18], bioinformatics [19], feature selection [20], image segmentation [21]–[23] and camera self-calibration [24],[25]. However, all MHs need to properly balance the exploration and exploitation phases to achieve good results, otherwise the solutions tend to be trapped in local optima or cannot converge properly.

MHs have been widely adopted to solve complex optimization tasks, including FS. Thus, the FS process for classification can be viewed as a search problem in a state space, where each state can be represented as a vector of size equal to the number of attributes in the problem and each element of the vector can take the value 1 if the corresponding attribute is selected and 0 otherwise. In the literature several metaheuristics have been widely used in recent years to deal with the FS optimization process. Such as, ant optimization algorithm [26], dragonfly optimization algorithm [27], whale optimization algorithm (WOA) [28] marine predators' algorithm (MPA)[29], sine cosine algorithm (SCA) [30], [31] Harris hawk optimization (HHO) algorithm[32] and other. Thus, there are also hybridizations between metaheuristics that has been used to address the FS problem [33]–[35].

Lately (in 2021), a new MH proposed by Hashim et al [36] to solve real-world problems called the Archimedes' optimization algorithm (AOA). The latter is based on Archimedes' principle in the law of physics. It describes the behavior of the force exerted when an object is partially or totally immersed in a fluid. Like most MHs, the AOA proposes solutions in the form of a population. Thus, the search agents are the immersed objects. Their density, volume and acceleration during collisions with other nearby objects are updated to define the updated positions of the objects. The main objective of the AOA is to bring the objects to a state of equilibrium. AOA was tested on 29 reference functions and with four engineering design problems. The results obtained by AOA showed promising results, and shows that AOA balanced between the phases of exploration and exploitation. The search performance of this algorithm was compared to well-known algorithms like GA, PSO, as well as to more recent additions such as WOA, SCA, HHO and EO.

AOA has already started to attract the attention of several researchers for the use of this algorithm in a variety of optimization problems, thanks to their important criteria related to simplicity, efficiency, adaptability and flexibility. In the literature, several works that are used AOA as an optimizer, among them we can cite: in [37] for human facial analysis, in [38] to solve the optimal locations and sizes of solar photovoltaic systems (SPV) in electrical distribution networks, in [39] for wind speed prediction, in [40] to eliminate selective harmonics in a cascaded H-bridge inverter (CHB). In addition, researchers have introduced modifications to the original AOA to increase its effectiveness. In the literature there are several improved versions of the traditional AOA for example: Enhanced Archimedes Optimization Algorithm (EAOA)[41] to improve the balance between exploration and exploitation of AOA and to improve the classification performance, I-AOA [42] an improved version of AOA which is based on combination of two effective strategies, the local escape operator (LEO) and orthogonal learning (OL) to determine the optimal parameters of the polymer electrolyte membrane (PEM) fuel cell (FC) and IAOA [43] allows to increase the population diversity in AOA, to further improve the balance between exploitation and exploration of AOA, and to avoid premature convergence problems whose objective to solve the optimal power flow problem (OPF).

According to the No-Free-Lunch (NFL) theorem, there is no algorithm that solves all optimization problems [44], i.e., most metaheuristics fail when the problem is modified. Moreover, the results obtained by AOA are promising and the statistical analysis of these results revealed that this algorithm can be considered as a good performance optimizer compared to well-known optimization algorithms such as: Multi-verse Optimizer (MVO) [45], Henry Gas Solubility Optimization (HGSO)[20], Harris Hawks Optimize (HHO) [46], Equilibrium Optimizer (EO) [47], and others. Moreover, AOA is used as an optimizer in several works mentioned above, which shows its optimization efficiency that is ensured by many important criteria that are related to simplicity, efficiency, adaptability and flexibility. This motivated us to try to develop a new FS approach based on AOA. The treatment of the FS problem by this algorithm aims at finding a good classification with high accuracy.

In this work, a new proposed FS approach based on envelopes by AOA is intended for data classification. It should be noted that this algorithm maintains a perfect balance between exploitation and exploration. Due to this feature, AOA is well suited for solving complex problems, especially feature selection. Our method is based on a hybridization between the AOA algorithm and k-NN for feature selection and data classification; the system we propose is defined in three phases: The first is the initialization phase, we generate a number of objects as the initial population and the size of each object in AOA corresponds to the number of attributes; the second is the phase of solution updates, we evaluate the quality of each candidate solution using AOA-assisted enveloping feature selection, a good compromise between accuracy and a reduced number of features should be ensured by a proposed objective function; and the last one is the classification phase, after finishing the process of our method, we return the best solution for classification, in this paper we used the K-NN classifier.

The important contributions of this paper are:

- The introduction of a new algorithm for the FS problem based on AOA.
- The performance of AOA was compared to other well-known metaheuristic algorithms in the literature for FS, such as MFO, SCA, WOA and BOA, all based on accuracy and number of selected features.
- The proposed method is evaluated on sixteen datasets with significantly high dimensions and small instances.
- The classification of the data by the features selected by the different optimizers was performed by the KNN classifier.
- A statistical comparison of the obtained results was performed by the optimizers using different analysis metrics.

The remaining sections of this paper are organized as follows: Section II will present the overview on the algorithms used in this work. The proposed method is described in Section III to solve the classification problem. The experiments and the results obtained are discussed in section V. Finally, the conclusion is presented in section VI.

II. Background Overview

A. Archimedes' Optimization Algorithm (AOA)

The AOA algorithm is a recent population-based metaheuristic introduced by Fatma Hashim in 2020 [36] This algorithm is part of the metaheuristics that are inspired by the rules of physics of the universe, especially Archimedes' law. The principle of Archimedes' phenomenon states that when an object is completely or partially immersed in a fluid, the fluid exerts an upward force on the object equal to the weight of the fluid displaced by the object [48]. The importance of the AOA algorithm lies in the formulation of the solution which is based on three auditory information: Volume (V), Density (D) and Acceleration (A) to the base agents. Thus, initially, the group of agents is randomly generated with dimensions (Dim). The random values V, D and A are provided as they are additive data. Then, the evaluation mechanism is performed for each object to specify the best object (O_{best}).

The operation of this algorithm is realized as follows: At each iteration, the AOA updates the density and volume of each object. The acceleration of the object is updated according to the condition of its collision with any other nearby object. The updated density, volume and acceleration determine the new position of an object in the current solution. The main steps of the AOA are described below:

1. **Initialization:** This step randomly initializes the real population that contains N objects using equation (1). Thus, each object is characterized by its density (D_i), volume (V_i) and acceleration (Γ_i) which are randomly defined using the following equations: Eq. (2), Eq. (3) and Eq. (4):

$$O_i = O_i^{min} + r_1 \times \left(O_i^{max} - O_i^{min}\right); \;\; i = 1, 2, \ldots, N \tag{1}$$

$$D_i = r_2 \tag{2}$$

$$V_i = r_3 \tag{3}$$

$$\Gamma_i = \Gamma_i^{max} + r_4 \times \left(\Gamma_i^{max} - \Gamma_i^{min}\right); \;\; i = 1, 2, \ldots, N \tag{4}$$

Where O_i represents the ith object, O_i^{max} and O_i^{min} are the maximum and minimum boundaries of the search space, respectively.

r_1, r_2, r_3 and r_4 are random vectors that belong to [0, 1].

The population will be evaluated by calculating the score of each object to unearth the best object (O_{best}) by combining their best values of density (D_{best}), volume (V_{best}) and acceleration (Γ_{best}).

2. **Density and volume updates:** In this step, the density and volume values for each object are updated by checking the best density and volume using Eq. (5) and Eq. (6).

$$D_i^{t+1} = D_i^t + s_1 \times (D_{Best} - D_i^t) \tag{5}$$

$$V_i^{t+1} = V_i^t + s_2 \times (V_{Best} - V_i^t) \tag{6}$$

Where s_1, s_2 are random scalars in [0, 1].

3. **Transfer coefficient and density scalar:**

In this step, the collision between the objects occurs until the steady state is reached. The main role of the transfer function ( ) is to switch from exploration mode to exploitation mode, defined by equation (7):

$$T_c = \exp\left(\frac{t-T}{T}\right) \tag{7}$$

T_c increases exponentially with time until it reaches 1. t is the current iteration, while T denotes the maximum number of iterations. Moreover, decreasing the density scalar d_s in AOA allows us to find an optimal solution using Eq. (8):

$$d_s^{t+1} = exp\left(\frac{t-T}{T}\right) - \left(\frac{t}{T}\right) \tag{8}$$

4. **Exploration phase:** In this step, the collision between agents occurs using random material selection (Mr). Thus, the update of acceleration objects is applied using equation (9) when the value of the transfer function is less than or equal to 0.5.

$$\Gamma_i^{t+1} = \frac{D_{Mr} + V_{Mr} + \Gamma_{Mr}}{D_i^{t+1} + V_i^{t+1}} \tag{9}$$

5. **Exploitation phase:** In this step, the collision between the agents is not realized. Thus, the update of acceleration objects is applied using equation (10) when the value of the transfer coefficient is greater than 0.5.

$$\Gamma_i^{t+1} = \frac{D_{Best} + V_{Best} + \Gamma_{Best}}{D_i^{t+1} + V_i^{t+1}} \tag{10}$$

Where Γ_{best} is the acceleration of the optimal object O_{best}.

6. **Acceleration normalization:** In this step, we normalize the acceleration to determine the rate of change using (11):

$$\Gamma_{i-norm}^{t+1} = \alpha \times \frac{\Gamma_i^{t+1} - \Gamma^{min}}{\Gamma^{max} - \Gamma^{min}} + \beta \tag{11}$$

Where α and β are set to 0.9 and 0.1, respectively. The Γ_{i-norm}^{t+1} determines the percentage of steps each agent will change. The highest value of the acceleration means that the object performs the exploration operation; otherwise, the exploitation mode is operational.

7. **The update process:** For the exploration phase ($T_c \leq 0.5$), the position of the ith object at iteration t+1 is changed by equation (12), while the position of the object is updated by equation (13) in the exploitation phase ($T_c > 0.5$).

$$O_i^{t+1} = O_i^t + c_1 \times r_5 \times \Gamma_{i-norm}^{t+1} \times d_s \times (O_{rand} - O_i^t) \tag{12}$$

Where c_1 is equal to 2.

$$O_i^{t+1} = O_{Best}^t + F \times c_2 \times r_6 \times \Gamma_{i-norm}^{t+1} \times d_s \times (\delta \times O_{Best} - O_i^t) \tag{13}$$

Where c_2 is fixed at 6.

The parameter δ is positively correlated with time and this parameter is proportionally related to the transfer coefficient T_c, i.e. $\delta = 2 \times T_c$. The main role of this parameter is to ensure a good balance between exploration and exploitation operations. During the early iterations, the margin between the best object and the other object is higher, allowing for a high random walk. However, in the later iterations, the margin will be reduced and provide a low random walk.

F is used for marking which controls the search direction using equation (14):

$$F = \begin{cases} +1 & if \;\; \lambda \leq 0.5 \\ -1 & if \;\; \lambda > 0.5 \end{cases} \tag{14}$$

Where λ = 2 × rand - 0.5.

8. **Evaluation:** In this step, we evaluate the new population using the score index Sc to determine the best object O_{best} and the best additive information, including D_{best}, V_{best} and Γ_{best}.

The pseudo code steps of the AOA algorithm is described in Algorithm 1.

B. K-Nearest Neighbor (K-NN)

The K-Nearest Neighbor (K-NN) is a method based on the notion of proximity (neighborhood) between variables and on reasoning from similar cases to make a decision. It is the training sample, associated with a distance function and a class choice function based on the classes of the nearest neighbors, which constitutes the model. To predict the class of a sample, the algorithm looks for the k nearest neighbors of this new case and predicts the most frequent answer of this k nearest neighbors [49]. Thus, the decision principle is simply to compute the distance of this sample to all the provided samples.

KNN is one of the most commonly used machine learning techniques with different datasets due to its simplicity and easy-to-implement advantages over other supervised machine learning

Algorithm 1: AOA

1. **Initialization**: N population size; T maximum iteration; c_1; c_2
2. Initialize N objects with their densities (D), volumes (V) and accelerations (Γ) using the equations from (1) to (4) respectively.
3. Evaluate the score for each object.
4. Determination of best object (O_{Best}) with (D_{Best}), (V_{Best}) and (Γ_{Best})
5. Set t = 1
6. **while** t ≤ T do
7. **for** each object i **do**
8. Update density and volume using Eq.(5) and Eq.(6)
9. Update transfer coefficient (T_c) and density scalar (d_s) using Eq. (7) and Eq. (8)
10. **if** $T_c \leq 0.5$ **then** **(Exploration operation)**
11. Update acceleration (Γ_i) using Eq. (9)
12. Normalize acceleration (Γ_i) using Eq. (11)
13. Update position using Eq. (12)
14. **else** **(Exploitation operation)**
15. Update acceleration using Eq. (10)
16. Update flagging control F using Eq. (13)
17. Update position using Eq. (14)
18. **end if**
19. Compute the score of each object.
20. **end for**
21. Determine the best object (O_{Best}) with the best value of (D_{Best}), (V_{Best}) and (Γ_{Best}).
22. Set $t = t + 1$
23. **end while**
24. **return** Best object with their quality

techniques. Therefore, KNN classifier is frequently used in several fields such as: healthcare, image and video recognition, finance, etc.

An object is classified according to a majority vote by its neighbors; the object obtains the class which is the most common among its K closest neighbors in the feature space. K must therefore be a positive integer, usually small. An odd k is often chosen to avoid equality in voting. The distance used for the calculation of the proximity of the neighbors is most often the Euclidean distance. The main steps of the k -NN algorithm for ranking the sample are presented in Algorithm 2.

Algorithm 2: k-NN algorithm

1. **Inputs**: Load the training and test data.
2. **Outputs**: Assign a class to the test point based on the majority of classes presented in the chosen points calculate accuracy
3. Choose the value of k for each point in test data
4. **while** stopping condition is not met **do**
5. Find the Euclidean distance to all training data points.
6. Store the Euclidean distances in a list and sort it.
7. Choose the first k points.
8. **Return** Accuracy (Acc)

III. Proposed Method

In this section, the proposed approach has been discussed in detail. In addition, a flowchart and an algorithmic model have also been described to understand the proposed solution. Fig. 1 describes the general flowchart of the proposed method and its implication in the feature selection problem for real-world data sets. The AOA technique used focuses on finding an optimal subset of features from the training set and is tested using the validation set.

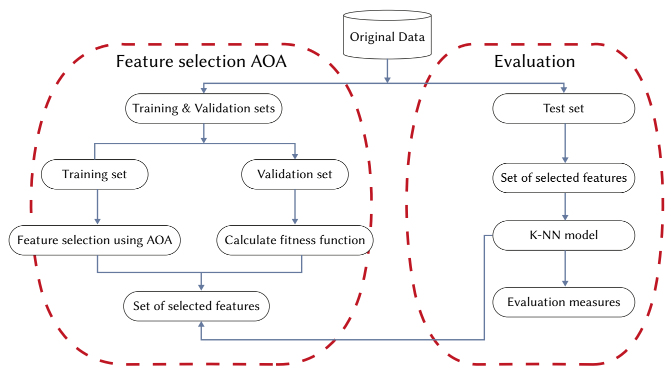

Fig. 1. The general flowchart of the proposed method.

According to the shortcomings of existing FS algorithms, this paper proposes a new method which is based on hybridization between AOA and k-NN algorithm for feature selection and data classification. The system we propose to solve the FS problem is defined in three phases: The first is the initialization phase, the second is the solution updates phase and the last is the classification phase.

In the first step, we define a solution as a numerical vector, we use a vector of (0 and 1) with 1 meaning that the attribute is selected, and 0 otherwise. We generate a number of objects as the initial population and the size of each object in the AOA is the number of attributes. At this point, if the value is greater than or equal to 0.5, then it is rounded to one. In this case, the attribute is considered a relevant feature. Conversely, the attribute is ignored when the value is rounded to zero.

In the second step, we evaluate the quality of each candidate solution using AOA-assisted wraparound feature selection, a good compromise between accuracy and a reduced number of features must be ensured by the objective function proposed in our paper which is described in equation (15) to determine the best $O_{best}O_{best}$ solution. Then, this process is repeated until the termination condition is met to make the necessary updates of the solutions.

In the third step, after finishing the process of our method, we

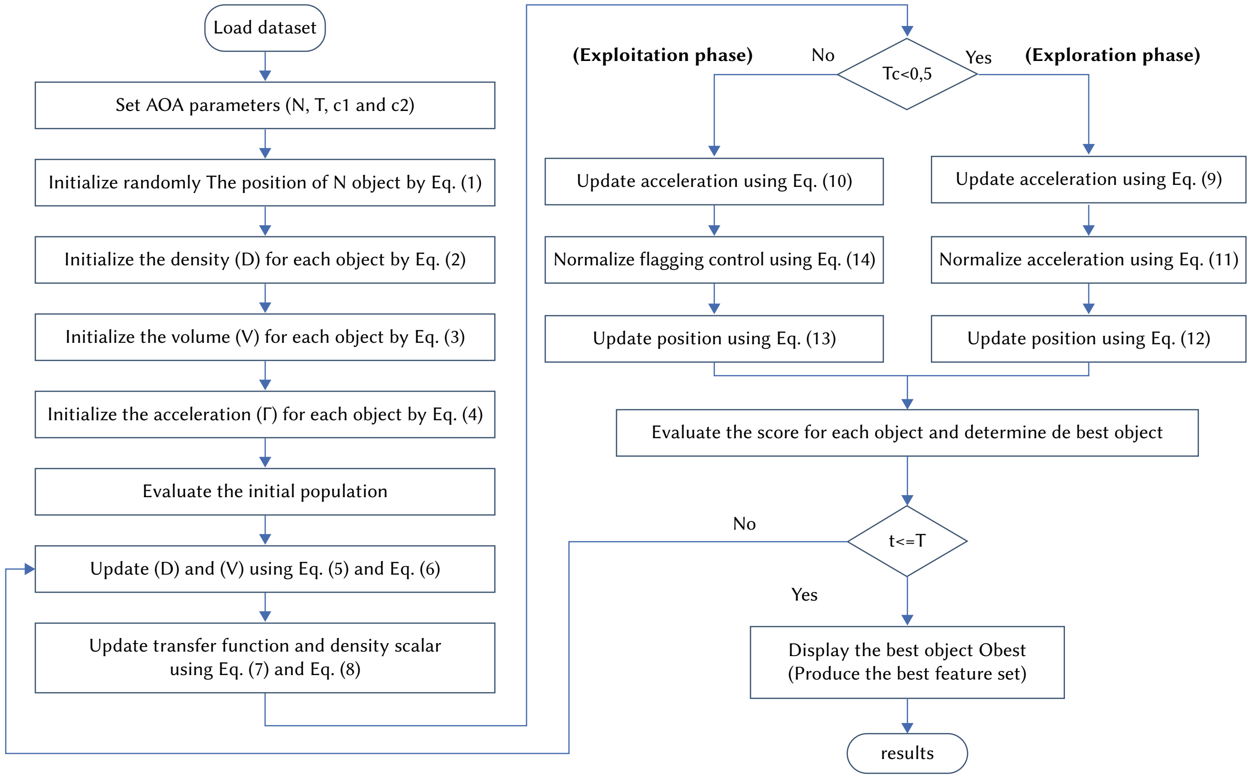

Fig. 2. The flowchart of the proposed FS algorithm.

return the best solution O_{best}. In the original data, we keep only the features with their values corresponding to O_{best}. We used a retention strategy for classification, which implies that we randomly divide the data set into two parts: 80% for the training set and 20% for the test set. In this paper we choose the k -NN algorithm (k = 5) to evaluate the accuracy using a test set. The value of k in k -NN is set to k = 5 based on several papers in the literature to make a fair comparison [50],[51].

The steps for integrating the operators of the AOA algorithm into the feature selection operation are explained in the following flowchart (Fig. 2).

A. Fitness Function

The main objective of the FS problem is to maximize the classification performance and maintain a minimum number of selected features. In wrapper methods, the fitness function is related to the construction of a new classifier based on the features involved in the individual. And as we mentioned before, our method uses a wrapper, so a learning algorithm must be integrated in the evaluation process. In our study we used a classifier well known for its performances, it is the k-NN classifier. To circumvent the cumbersome nature of this approach, we have defined a fitness function that allows us to control the accuracy of the selected features during an iterative process to check the quality for each iteration. Therefore, the overall goal is to find the minimum value of the fitness function given in equation (15):

$$fit_{\vartheta} = \tau * Err + (1-\tau)\frac{\sum_i \vartheta_i}{n} \quad (15)$$

Where:

- *Err* is defined as the classification error rate;
- τ is a constant controlling the classification importance with respect to the number of selected features;
- ϑ is a vector of size n with 0/1 elements representing unselected/selected features;
- n is the total number of features in the data set.

In general, wraparound methods search the space of subsets of variables, guided by the model output. They therefore incorporate the classification algorithm into the attribute selection procedure and use the classification error rate as an evaluation criterion. As mentioned earlier, our attribute selection approach is an envelope approach, which uses the k-NN classifier in the evaluation phase to correctly identify the appropriate features that should be selected. This method achieves good performance and often gets better results; however, it increases the time needed to reach a good solution. In our study the k-NN algorithm used in the trial-and-error experiments where the best choice of K is selected (K = 5) as the best performing on all data sets.

As shown in Fig. 3, our approach contains two search phases: local and global. The transition between these two phases is controlled by the transfer function given in equation (Eq. 7), which means that if the value of T_c is less than 0.5, our algorithm will perform a global search (exploration); otherwise, it will perform a local search (exploitation). Thus, our proposed method has achieved better efficiency when balancing between exploitation and exploration techniques in the search space.

IV. Experimental Results

This section evaluates the effectiveness of our developed method on eighteen benchmark datasets. In addition, we compare it to four other FS algorithms.

A. Dataset Description and Parameter Setting

To verify the performance of our proposed FS Wrapper model which is based on the AOA algorithm, we performed a comparison of our method with the following FS optimization algorithms: SCA, WOA, BAO and MFO. Each solution was evaluated using a fitness function given in equation (Eq. 15), which improves the accuracy of predictions and reduces the number of features. The effectiveness of the approaches proposed in the experiments is measured by several well-known and widely used evaluation metrics in this field [52], [53] which are cited in the next section. All algorithms are hybrid with the standard KNN machine learning classifier (with k=5). All experiments are conducted under the same conditions and the average of each evaluation metric is calculated from 20 independent runs. The proposed algorithms were tested on sixteen benchmark datasets (UCI repository) [54] to discuss the experimental results and comparisons. The details of the datasets are represented in Table I. This dataset is divided into 80% training and 20% testing. All parameters of the algorithms used in this paper are given in Table II with the following common parameters: N=10 (the number of search agents) and T=80 (the maximum number of iterations). In our study we used the MATLAB 2014b platform under the Microsoft Windows 10 professional 64bit operating system with the following hardware configuration: an Intel®Core TM i5 processor (3.20 GHz) with 4 GB RAM.

TABLE I. The Characteristics of the Dataset

No	Dataset	No of features	No of instances
DS1	AA	30	102
DS2	BreastEW	30	596
DS3	CongressEW	16	435
DS4	Exactly	13	1000
DS5	Exactly2	13	1000
DS6	HeartEW	13	270
DS7	IonosphereEW	34	351
DS8	KrvskpEW	36	3196
DS9	M-of-n	13	1000
DS10	penglungEw	325	569
DS11	sonarEW	60	208
DS12	SpecEw	22	267
DS13	Vote	16	300
DS14	WaveformEW	40	5000
DS15	WineEW	13	178
DS16	Zoo	16	101

TABLE II. Parameters of the Algorithms Used in the Experiments

Algorithm	Parameter	Value
SCA	a	2
WOA	$\vec{a}$	decreases linearly from 2 to 0
	$\vec{a}_2$	decreases linearly from −1 to −2
	b	1
BAO	a	0.8
MFO	b	0.75
AOA	c_1	2
	c_2	6
	α	0.9
	β	0.1

B. Performance Evaluation Measures

The effectiveness of the proposed method was evaluated by well-known and widely used evaluation metrics [49] which are:

- Accuracy is the proportion of the number of positive tuples and negative tuples obtained by the classification algorithms in the total number of hits, as shown in Equation 16.

$$Acc = \frac{(tP+tN)}{(tP+tN+fP+fN)} \tag{16}$$

tP, tN, fP and fN denote the numbers of true positives, true negatives, false positives and false negatives, respectively.

- F-score (FScore) - This is an evaluation of the precision of the classifier, statistically, it represents the harmonic mean between recall and accuracy. The formula for F-score is given by the following expression:

$$F_{score} = 2 \times \frac{precision \times recall}{precision + recall} = \frac{tP}{tP+\frac{1}{2}(fP+fN)} \tag{17}$$

- The average (MEAN)- The average of the fitness function values obtained by running an optimization algorithm M several times. Equation 18 displays the mathematical expression of the average of the fitness values:

$$MEAN = \frac{1}{M}\sum_{i=1}^{M} fit_{\vartheta} \tag{18}$$

- The best and worst fitness values are represented in equations 19 and 20 respectively:

$$BEST = Min_{i=1}^{M} fit_{\vartheta} \tag{19}$$

$$WORST = Max_{i=1}^{M} fit_{\vartheta} \tag{20}$$

- Standard Deviation (StD)

  The standard deviation (StD) represents the variance of the best solutions found for each optimization algorithm run for M times. The StD is given in equation 21:

$$StD = \sqrt{\frac{1}{M-1}\sum_{i=1}^{M}(fit_{\vartheta} - MEAN)^2} \tag{21}$$

- Average selection size (AVRSS)

  The average selection size is the average size of the selected features relative to the total number of features. This measure is as in equation (22).

$$AVRSS = \frac{1}{M}\sum_{i=1}^{M}\frac{size(fit_{\vartheta})}{D} \tag{22}$$

  Where size(x) is the number of values of vector x for each run *i*, and *D* is the number of features in the original data set.

C. Results and Discussion

In all experiments, we use the cross-validation process to evaluate the performance. The k-fold cross validation algorithm consists in splitting the initial set of examples D into k blocks. We then repeat k evaluation learning phases, where a hypothesis h is obtained by learning on (k-1) blocks of data and tested on the remaining block. The error estimator is obtained as the average of the k empirical errors thus obtained.

In this section, as shown in Tables (III, IV, V, VI, VII, VIII) and Figures (3, 4), we evaluate the performance of the proposed method using various feature selection sets and compare it with other algorithms to prove its capability. The bold values in each row indicate the best result among the five algorithms.

Table III shows the best results of fitness values. In this table, we can notice that our proposed method presented competitive results with other comparison algorithms. AOA obtained the best results in nine datasets (i.e., DS2, DS3, DS8, DS10, DS11, DS12, DS13 and DS16), while in the other remaining datasets, the best results of the fitness values are not stable in one algorithm, but overall, there is not a big difference compared to the values obtained by our method in all test of the experiment.

TABLE III. The Best Fitness Values Obtained by the Different Optimizers

dataset	SCA	WOA	BAO	MFO	AOA
DS1	0,01293	0,09771	0,06521	0,08923	**0,06265**
DS2	0,06101	0,07923	0,08452	0,12002	**0,05567**
DS3	0,11321	0,10232	0,11082	0,07965	**0,03512**
DS4	0,05934	0,27624	**0,05385**	0,07212	0,05694
DS5	0,21361	**0,21331**	0,22971	0,24012	0,22968
DS6	0,43210	0,31812	0,28721	**0,24324**	0,26993
DS7	**0,15349**	0,27305	0,17434	0,21871	0,15422
DS8	0,21320	0,11325	0,08786	0,10239	**0,07923**
DS9	**0,06134**	0,24533	0,09742	0,11552	0,07038
DS10	0,06834	0,08972	0,08191	0,08321	**0,05396**
DS11	0,05940	0,09308	0,08229	0,23075	**0,00601**
DS12	0,07941	0,23931	0,20487	0,09501	**0,05729**
DS13	0,21431	0,32945	0,28214	0,20815	**0,20726**
DS14	0,31401	0,09329	0,10051	**0,24883**	0,29627
DS15	**0,02546**	0,08757	0,05384	0,03021	0,05388
DS16	0,02539	0,08611	0,05690	0,02612	**0,02034**

On the other hand, for the worst results of fitness values, as shown in Table IV, our proposed method is superior to other algorithms. AOA achieved the best results in 50% of all datasets used in the experiment (i.e., DS1, D23, DS3, DS4, DS9, DS10, DS13 and DS16). WOA achieved second place in 19% of all datasets (i.e., DS7, DS11, and DS12), followed by BAO and MFO with 13% each; while the SCA algorithm was ranked as the least successful.

The results of feature selection of all methods using the average of fitness values are recorded in Table V. We can observe from this table that our proposed technique got the best average in nine out of 16 datasets (i.e., 56%), so it was ranked first. AOA got the second place in four of the 16 datasets (i.e., DS1, DS8, DS10 and D14). BAO had the worst performance.

Table VI shows the StD values of the compared algorithms. It can be noticed that the proposed method based on the AOA algorithm has an excellent performance. The StD values prove that the proposed AOA is a powerful method for solving feature selection problems. AOA got more than 56% of the best cases according to the StD value, followed by SCA, BAO and MFO which got almost 13% for each and finally WOA algorithm is ranked as the bad method according to the StD values in the dataset compared to other methods. But, in general, the average StD values for all the datasets used in our experiment selected the performance of the algorithms even if they did not get the best StD value, for example in the case of WOA algorithm.

TABLE IV. Worst Fitness Values Obtained by the Different Optimizers

dataset	SCA	WOA	BAO	MFO	AOA
DS1	0,0452	0,0597	0,0464	0,0543	**0,0417**
DS2	0,0638	0,0492	0,0646	0,1107	**0,0562**
DS3	0,1774	0,0563	0,0762	0,0767	**0,0216**
DS4	0,0619	0,0463	0,0538	0,0617	**0,0321**
DS5	**0,2769**	0,2799	0,2778	0,2899	0,2957
DS6	0,4853	0,2101	0,2342	**0,2033**	0,2136
DS7	0,3038	**0,1231**	0,1193	0,1693	0,1345
DS8	0,2538	0,0743	**0,0435**	0,1048	0,0753
DS9	0,1613	0,0959	0,0859	0,1001	**0,0764**
DS10	0,0736	0,0616	0,0513	0,0615	**0,0462**
DS11	0,2007	**0,0030**	0,0525	0,1992	0,0041
DS12	0,1535	**0,0481**	0,1045	0,0750	0,0553
DS13	0,2703	0,2184	0,2182	0,1781	**0,1694**
DS14	0,3756	0,0637	**0,0529**	0,2210	0,2213
DS15	0,1433	0,0692	0,0385	**0,0275**	0,0538
DS16	0,1453	0,0437	0,0319	0,0153	**0,0100**

TABLE V. The Average Fitness Values Obtained by the Different Optimizers

dataset	SCA	WOA	BAO	MFO	AOA
DS1	**0,0291**	0,0787	0,0558	0,0718	0,0522
DS2	0,0624	0,0642	0,0746	0,1154	**0,0559**
DS3	0,1453	0,0793	0,0935	0,0782	**0,0284**
DS4	0,0606	0,1613	0,0538	0,0669	**0,0445**
DS5	**0,2452**	0,2466	0,2537	0,2649	0,2626
DS6	0,4587	**0,2241**	0,2607	0,2233	0,2418
DS7	0,2286	0,2231	0,1468	0,1940	**0,1444**
DS8	0,2335	0,0938	**0,0457**	0,1036	0,0773
DS9	0,1113	0,1556	0,0917	0,1078	**0,0734**
DS10	0,0710	**0,0457**	0,0666	0,0724	0,0501
DS11	0,1300	0,0481	0,0674	0,2150	**0,0051**
DS12	0,1164	0,1437	0,1547	0,0850	**0,0563**
DS13	0,2423	0,2739	0,2502	0,1931	**0,1883**
DS14	0,3448	0,0785	0,0767	**0,2349**	0,2588
DS15	0,0844	0,0784	0,0462	**0,0289**	0,0539
DS16	0,0854	0,0649	0,0444	0,0207	**0,0152**

Fig 3. The percentage of the selected features for the comparative methods.

Fig 4. Comparison between the different proposed methods in terms of classification accuracy on all datasets.

TABLE VI. The Average Fitness Values Obtained by the Different Optimizers

dataset	SCA	WOA	BAO	MFO	AOA
DS1	0,0064	0,0038	0,0065	0,0097	**0,0013**
DS2	0,0071	0,0039	0,0037	0,0032	**0,0113**
DS3	**0,0021**	0,0041	0,0078	0,0078	0,0029
DS4	0,0021	0,0063	0,0052	0,0077	**0,0000**
DS5	0,0357	0,0057	0,0042	0,0072	**0,0012**
DS6	0,0056	0,0083	0,0276	**0,0102**	0,0155
DS7	0,0131	0,0074	0,0081	0,0331	**0,0126**
DS8	0,0036	**0,0031**	0,0098	0,0393	0,0085
DS9	0,0106	0,0031	**0,0000**	0,0141	0,0074
DS10	**0,0000**	0,0106	0,0022	0,0133	0,0034
DS11	0,0058	0,0064	0,0134	**0,0000**	0,0054
DS12	0,0027	0,0081	0,0052	0,0134	**0,0000**
DS13	0,0098	0,0092	0,0093	0,0112	**0,0087**
DS14	0,0089	0,0087	**0,0035**	0,0052	0,0142
DS15	0,0036	0,0042	0,0034	0,0091	**0,0000**
DS16	0,0000	0,0002	0,0031	0,0063	**0,0000**

Moreover, the percentage of the best number of selected features is presented in Fig. 3. The performance of our proposed method is clearly visible in this figure. To obtain higher accuracy values, it is necessary to obtain a small number of selected features. It can be seen that WOA obtained the smallest number of features in 68% of the data sets. WOA obtained the second largest and smallest number of features in 22% of the datasets, followed by SCA; while MFO recorded the largest number of features among all algorithms.

From Fig. 4, we notice that the results of the envelope attribute selection method based on the AOA algorithm combined with the KNN classifier are better than the results obtained by the other methods with the same classifier in terms of accuracy. The results reach a classification accuracy of 97% for the average of all datasets by our approach, followed by the optimization by WOA which has a classification accuracy of 91%, while MFO showed the lowest accuracy in the average of all datasets. We also note that the results reach 100% classification rate for four datasets used in the experiments, namely DS4, DS10, DS15 and DS16.

To statistically validate our study and to give it more value, we applied the Wilcoxon statistical test [55]. This test (Wilcoxon Rank-sum) is a non-parametric statistical test that tests the hypothesis that the medians of each of two groups of data are close. As in all statistical tests, it allows to accept and also to reject the NULL hypothesis. The latter considers that the median of two real data vectors X and Y is fair. The p-value was compared at a significance level of 0.05. For ease of understanding, the symbols "w/t/l" indicate that the AOA is superior (win), equal (tie) and inferior (lose) to the other algorithms. So as shown by the p-values presented in Table VII, we can see that the method proposed by AOA brings a significant improvement over the algorithms: SCA and WOA in most of the collections used. However, this superiority is statistically weak for the other algorithms such as BAO and MFO. Therefore, AOA shows a good performance in terms of Wilcoxon test and it can be chosen as a reference algorithm as it offers significant classification results in this work. Also, in terms of F-score values obtained, the proposed method outperforms the other algorithms on thirteen datasets as shown in Table VIII, on the other hand the SCA and WOA techniques are only effective on two datasets.

TABLE VII. P-Values of the Wilcoxon Rank Sum Test of the AOA Accuracy Results Versus Other Algorithms

dataset	p-values			
	SCA	WOA	BAO	MFO
DS1	0,0064304	0,003800	<u>0,066050</u>	<u>0,078097</u>
DS2	0,0072970	0,003399	0,003027	0,003002
DS3	0,0021002	<u>0,104111</u>	<u>0,200778</u>	<u>0,120078</u>
DS4	0,0036721	0,006333	0,000102	0,008077
DS5	0,0354567	0,005713	0,010142	0,080172
DS6	0,0056000	0,008314	<u>0,027036</u>	0,109102
DS7	0,0105631	<u>0,067401</u>	<u>0,080081</u>	<u>0,321031</u>
DS8	0,0030116	<u>0,003120</u>	0,002098	0,030093
DS9	0,0141206	0,003100	0,000000	0,000000
DS10	0,0000000	<u>0,010612</u>	<u>0,070022</u>	0,013073
DS11	0,0059018	0,006224	0,007134	0,000056
DS12	0,0028900	<u>0,208100</u>	0,005332	<u>0,098034</u>
DS13	0,0049098	0,002340	<u>0,070293</u>	0,011002
DS14	0,0030989	<u>0,078712</u>	0,003505	<u>0,108052</u>
DS15	0,0036003	0,064233	<u>0,201034</u>	<u>0,300091</u>
DS16	0,0000000	0,000400	0,000305	0,003005
w|t|l	16|0|0	11|5|0	9|7|0	8|8|0

TABLE VIII. The F-Score Values Obtained by the Different Optimizers

dataset	SCA	WOA	BAO	MFO	AOA
DS1	0,9201	0,9555	0,9202	0,7365	**0,9578**
DS2	0,9145	0,9541	0,9661	0,9221	**0,9896**
DS3	0,9565	0,9478	0,9093	0,5393	**0,9979**
DS4	0,9773	0,9160	0,9398	0,4571	**1,0000**
DS5	0,9731	0,9321	0,9422	0,8987	**0,9833**
DS6	0,6653	**0,7921**	0,5342	0,7201	0,7788
DS7	0,7531	0,8021	0,7696	0,5261	**0,9038**
DS8	0,9679	0,9285	0,7337	0,7228	**0,9902**
DS9	0,8435	0,9555	**1,0000**	0,4101	0,9995
DS10	**1,0000**	0,9517	0,9899	**1,0000**	**1,0000**
DS11	0,8621	**0,9722**	0,7156	0,5210	0,8621
DS12	0,9775	0,9866	0,9663	0,6896	**0,9989**
DS13	0,8101	0,7623	0,7899	0,7034	**0,8811**
DS14	0,8651	0,7933	0,8132	0,7901	**0,9543**
DS15	0,9102	0,9798	0,9067	0,8991	**1,0000**
DS16	**1,0000**	0,9998	0,8446	0,8436	**1,0000**

Looking at the tables from III to VIII, figures from 3 to 4 and the in-depth analysis of the results of the developed method against the other comparison methods, we can see that the AOA algorithm achieved the best results in terms of accuracy, number of selected features and fitness value on the majority of the datasets. The results that are displayed in the previous tables show that the effectiveness of each algorithm depends on the dataset used, but overall and according to the results of the average values of the evaluation criteria used in this paper on the dataset, we can therefore conclude that our approach often has superiority over the other methods based on WOA, SCA, MFO and BAO algorithms that are performed under the same conditions and on the same datasets that we have mentioned previously.

In general, the previous results show that there is a significant improvement in solving feature selection problems using the operators of the AOA algorithm. Therefore, it can be said that AOA can be considered as an effective optimization algorithm, especially for solving feature selection problems.

V. Conclusion

In this work, we have addressed the feature selection problem by using a recent optimizer called the Archimedes' optimization algorithm (AOA). Experiments are applied on sixteen different datasets (UCI) to handle the FS optimization task and to study the effectiveness of the proposed method. The latter has been compared by four FS methods based on WOA, SCA, FMO and BAO algorithms with the same classifier which is KNN. We also used several evaluation criteria to properly assess different aspects of the performance of the compared algorithms. The comparisons revealed that our technique achieved the best average fitness in nine of the 16 datasets, the lowest StD value in 56% of the datasets, and the lowest feature count in 68% of the datasets. These results indicate that AOA is able to select a small number of features and achieve very high classification accuracy. Therefore, we conclude that the operators of the AOA algorithm improve the mining and exploration phases well which increases the efficiency of this algorithm to solve classification problems.

References

[1] M. A. Khan et al., "Cucumber leaf diseases recognition using multi level deep entropy-ELM feature selection," Applied Sciences, vol. 12, no. 2, p. 593, 2022.

[2] M. A. Khan et al., "A fused heterogeneous deep neural network and robust feature selection framework for human actions recognition," Arabian Journal for Science and Engineering, pp. 1–16, 2021.

[3] A. Mehmood, U. Tariq, C. W. Jeong, Y. Nam, R. R. Mostafa, and A. Elaeiny, "Human Gait Recognition: A Deep Learning and Best Feature Selection Framework," Computers, Materials & Continua, vol. 70, pp. 343–360, 2022.

[4] N. Hussain et al., "Multiclass cucumber leaf diseases recognition using best feature selection," Computers, Materials & Continua, vol. 70, pp. 3281–3294, 2022.

[5] F. Zia et al., "A multilevel deep feature selection framework for diabetic retinopathy image classification," 2022.

[6] A. Rehman, M. A. Khan, T. Saba, Z. Mehmood, U. Tariq, and N. Ayesha, "Microscopic brain tumor detection and classification using 3D CNN and feature selection architecture," Microscopy Research and Technique, vol. 84, no. 1, pp. 133–149, 2021.

[7] M. U. Khan et al., "Expert hypertension detection system featuring pulse plethysmograph signals and hybrid feature selection and reduction scheme," Sensors, vol. 21, no. 1, p. 247, 2021.

[8] M. A. Khan et al., "Multimodal brain tumor classification using deep learning and robust feature selection: A machine learning application for radiologists," Diagnostics, vol. 10, no. 8, p. 565, 2020.

[9] L. Khrissi, H. Satori, K. Satori, and N. el Akkad, "An Efficient Image Clustering Technique based on Fuzzy C-means and Cuckoo Search Algorithm," International Journal of Advanced Computer Science and Applications, vol. 12, no. 6, pp. 423–432, 2021, doi: 10.14569/IJACSA.2021.0120647.

[10] L. Khrissi, N. E. Akkad, H. Satori, and K. Satori, "Color image segmentation based on hybridization between Canny and k-means," in 7th Mediterranean Congress of Telecommunications 2019, CMT 2019, 2019. doi: 10.1109/CMT.2019.8931358.

[11] D. Yousri, M. Abd Elaziz, L. Abualigah, D. Oliva, M. A. A. Al-Qaness, and A. A. Ewees, "COVID-19 X-ray images classification based on enhanced fractional-order cuckoo search optimizer using heavy-tailed distributions," Applied Software Computing, vol. 101, p. 107052, 2021.

[12] Z. Faska, L. Khrissi, K. Haddouch, and N. el Akkad, "A Powerful and Efficient Method of Image Segmentation Based on Random Forest Algorithm," in Digital Technologies and Applications, 2021, pp. 893–903.

[13] L. Khrissi, N. El Akkad, H. Satori, and K. Satori, "Simple and Efficient Clustering Approach Based on Cuckoo Search Algorithm," 2020 Fourth International Conference On Intelligent Computing in Data Sciences (ICDS), pp. 1–6, Oct. 2020, doi: 10.1109/ICDS50568.2020.9268754.

[14] S. Cheng, L. Ma, H. Lu, X. Lei, and Y. Shi, "Evolutionary computation for solving search-based data analytics problems," Artificial Intelligence Review, vol. 54, no. 2, pp. 1321–1348, 2021, doi: 10.1007/s10462-020-09882-x.

[15] D. S. A. Elminaam, S. A. Ibrahim, E. H. Houssein, and S. M. Elsayed, "An Efficient Chaotic Gradient-Based Optimizer for Feature Selection," IEEE Access, vol. 10, pp. 9271–9286, 2022, doi: 10.1109/ACCESS.2022.3143802.

[16] H. Moussaoui, N. el Akkad, and M. Benslimane, "Moroccan Carpets Classification Based on SVM Classifier and ORB Features," in International Conference on Digital Technologies and Applications, 2022, pp. 446–455.

[17] C. C. Aggarwal, X. Kong, Q. Gu, J. Han, and P. S. Yu, "Active learning: A survey," Data Classification: Algorithms and Applications, pp. 571–605, 2014, doi: 10.1201/b17320.

[18] E. H. Houssein, A. G. Gad, Y. M. Wazery, and P. N. Suganthan, "Task scheduling in cloud computing based on meta-heuristics: Review, taxonomy, open challenges, and future trends," Swarm and Evolutionary Computation, vol. 62, p. 100841, 2021.

[19] F. A. Hashim, E. H. Houssein, K. Hussain, M. S. Mabrouk, and W. Al-Atabany, "A modified Henry gas solubility optimization for solving motif discovery problem," Neural Computing & Applicationsvol. 32, no. 14, pp. 10759–10771, 2020.

[20] N. Neggaz, E. H. Houssein, and K. Hussain, "An efficient henry gas solubility optimization for feature selection," Expert Systems With Applications, vol. 152, p. 113364, 2020.

[21] L. Khrissi, N. el Akkad, H. Satori, and K. Satori, "Image Segmentation Based on K-means and Genetic Algorithms," in Advances in Intelligent Systems and Computing, 2020, vol. 1076, pp. 489–497. doi: 10.1007/978-981-15-0947-6_46.

[22] L. Khrissi, N. el Akkad, H. Satori, and K. Satori, "Clustering method and sine cosine algorithm for image segmentation," Evolutionary Intelligence, Jan. 2021, doi: 10.1007/s12065-020-00544-z.

[23] L. Khrissi, N. el Akkad, H. Satori, and K. Satori, "A Performant Clustering Approach Based on An Improved Sine Cosine Algorithm," International Journal of Computing, pp. 159–168, Jun. 2022, doi: 10.47839/ijc.21.2.2584.

[24] M. Merras, N. el Akkad, A. Saaidi, A. G. Nazih, and K. Satori, "Camera calibration with varying parameters based on improved genetic algorithm," WSEAS Transactions on Computersvol. 13, pp. 129–137, 2014.

[25] N. el Akkad, M. Merras, A. Saaidi, and K. Satori, "Robust method for self-calibration of cameras having the varying intrinsic parameters," Journal of Theoretical and Applied Information Technology, vol. 50, no. 1, pp. 57 – 67. 2013.

[26] H. M. Zawbaa, E. Emary, and B. Parv, "Feature selection based on antlion optimization algorithm," Proceedings of 2015 IEEE World Conference on Complex Systems, WCCS 2015, 2016, doi: 10.1109/ICoCS.2015.7483317.

[27] M. Mafarja et al., "Binary dragonfly optimization for feature selection using time-varying transfer functions," 2018, doi: 10.1016/j.knosys.2018.08.003.

[28] A. G. Hussien, A. E. Hassanien, E. H. Houssein, S. Bhattacharyya, and M. Amin, "S-shaped binary whale optimization algorithm for feature selection," vol. 727. Springer Singapore, 2019. doi: 10.1007/978-981-10-8863-6_9.

[29] A. T. Sahlol, D. Yousri, A. A. Ewees, M. A. A. Al-Qaness, R. Damasevicius, and M. A. Elaziz, "COVID-19 image classification using deep features and fractional-order marine predators algorithm," Scientific Reports , vol. 10, no. 1, pp. 1–15, 2020.

[30] A. I. Hafez, H. M. Zawbaa, E. Emary, and A. E. Hassanien, "Sine cosine optimization algorithm for feature selection," in 2016 international symposium on innovations in intelligent systems and applications (INISTA), 2016, pp. 1–5.

[31] P. C. Chiu, A. Selamat, O. Krejcar, K. K. Kuok, E. Herrera-Viedma, and G. Fenza, "Imputation of Rainfall Data Using the Sine Cosine Function Fitting Neural Network.," International Journal of Interactive Multimedia & Artificial Intelligence, vol. 6, no. 7, 2021.

[32] Y. Zhang, R. Liu, X. Wang, H. Chen, and C. Li, "Boosted binary Harris hawks optimizer and feature selection," Engineering with Computers, vol. 37, no. 4, pp. 3741–3770, 2021.

[33] N. Neggaz, A. A. Ewees, M. Abd Elaziz, and M. Mafarja, "Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection," Expert Systems With Applications, vol. 145, p. 113103, 2020.

[34] M. M. Mafarja and S. Mirjalili, "Hybrid Whale Optimization Algorithm with simulated annealing for feature selection," Neurocomputing, vol. 260, pp. 302–312, Oct. 2017, doi: 10.1016/j.neucom.2017.04.053.

[35] M. Abdel-Basset, W. Ding, and D. El-Shahat, "A hybrid Harris Hawks optimization algorithm with simulated annealing for feature selection," Artificial Intelligence Review, vol. 54, no. 1, pp. 593–637, 2021.

[36] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, "Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems," Applied Intelligence, vol. 51, no. 3, pp. 1531–1551, 2021, doi: 10.1007/s10489-020-01893-z.

[37] I. Neggaz and H. Fizazi, "An Intelligent handcrafted feature selection using Archimedes optimization algorithm for facial analysis," Soft Computing, vol. 26, pp. 10435–10464, 2022.

[38] V. Janamala and K. Radha Rani, "Optimal allocation of solar photovoltaic distributed generation in electrical distribution networks using Archimedes optimization algorithm," Clean Energy, vol. 6, no. 2, pp. 271–287, 2022.

[39] L. Zhang, J. Wang, X. Niu, and Z. Liu, "Ensemble wind speed forecasting with multi-objective Archimedes optimization algorithm and sub-model selection," Applied Energy, vol. 301, p. 117449, 2021.

[40] R. A. Khan et al., "Archimedes Optimization Algorithm Based Selective Harmonic Elimination in a Cascaded H-Bridge Multilevel Inverter," Sustainability, vol. 14, no. 1, p. 310, 2021.

[41] A. S. Desuky, S. Hussain, S. Kausar, M. A. Islam, and L. M. el Bakrawy, "EAOA: An Enhanced Archimedes Optimization Algorithm for Feature Selection in Classification," IEEE Access, vol. 9, pp. 120795–120814, 2021.

[42] E. H. Houssein, B. E. Helmy, H. Rezk, and A. M. Nassef, "An enhanced Archimedes optimization algorithm based on Local escaping operator and Orthogonal learning for PEM fuel cell parameter identification," Engineering Applications of Artificial Intelligence, vol. 103, p. 104309, 2021, doi: https://doi.org/10.1016/j.engappai.2021.104309.

[43] O. Akdag, "A Improved Archimedes Optimization Algorithm for multi/single-objective Optimal Power Flow," Electric Power Systems Research, vol. 206, p. 107796, 2022.

[44] D. Izzo, M. Märtens, and B. Pan, "A survey on artificial intelligence trends in spacecraft guidance dynamics and control," Astrodynamics, vol. 3, no. 4, pp. 287–299, 2019, doi: 10.1007/s42064-018-0053-6.

[45] A. A. Ewees, M. A. el Aziz, and A. E. Hassanien, "Chaotic multi-verse optimizer-based feature selection," Neural Computing and Applications, vol. 31, no. 4, pp. 991–1006, 2019, doi: 10.1007/s00521-017-3131-4.

[46] Y. Zhang, R. Liu, X. Wang, H. Chen, and C. Li, "Boosted binary Harris hawks optimizer and feature selection," Engineering with Computers, vol. 37, no. 4, pp. 3741–3770, 2021.

[47] G. Tikhe, T. Joshi, A. Lahorkar, A. Sane, and J. Valadi, "Feature selection using equilibrium optimizer," in Data Engineering and Intelligent Computing, Springer, 2021, pp. 307–315.

[48] C. Rorres, "Completing book II of Archimedes's on floating bodies," The mathematical intelligencer, vol. 26, no. 3, pp. 32–42, 2004.

[49] Y. Y. Yiming, "An Evaluation of Statistical Approaches to Text Categorization," Journal of Information Retrieval, vol. 1, pp. 67–88, 1999.

[50] M. Mafarja, I. Aljarah, H. Faris, A. I. Hammouri, A. M. Al-Zoubi, and S. Mirjalili, "Binary grasshopper optimisation algorithm approaches for feature selection problems," Expert Systems with Applications, vol. 117, pp. 267–286, 2019, doi: 10.1016/j.eswa.2018.09.015.

[51] T. Thaher, A. A. Heidari, M. Mafarja, J. S. Dong, and S. Mirjalili, "Binary Harris Hawks Optimizer for High-Dimensional, Low Sample Size Feature Selection," in Evolutionary Machine Learning Techniques: Algorithms and Applications, S. Mirjalili, H. Faris, and I. Aljarah, Eds. Singapore: Springer Singapore, 2020, pp. 251–272. doi: 10.1007/978-981-32-9990-0_12.

[52] A. S. Desuky and L. M. el Bakrawy, "Improved prediction of post-operative life expectancy after thoracic surgery," Advances in Systems Science and Application, vol. 16, no. 2, pp. 70–80, 2016.

[53] M. Gong, "A Novel Performance Measure for Machine Learning Classification," International Journal of Managing Information Technology, vol. 13, no. 1, pp. 11–19, 2021, doi: 10.5121/ijmit.2021.13101.

[54] D. Dua and C. Graff, "UCI machine learning repository," 2017.

[55] D. Rey and M. Neuhäuser, "Wilcoxon-signed-rank test," in International encyclopedia of statistical science, Springer, 2011, pp. 1658–1659.

Lahbib Khrissi

Lahbib Khrissi received the bachelor's and master's degrees from SMBA-Fez University in 2000 and 2007 respectively. He is currently working toward the PhD degree in the LISAC Laboratory at faculty of Sciences, Sidi Mohammed Ben Abdellah University, Fez, Morocco. His current research interests include Data mining, Image processing and Artificial intelligence.

Nabil El Akkad

Nabil El Akkad received the PhD degree from SMBA-Fez University in 2014. He is currently a professor of computer science at National School of Applied Sciences (ENSA) of Fez, Sidi Mohammed Ben Abdellah University, Fez, Morocco. He is a member of the LISA and LIIAN Laboratories. His research interests include Artificial intelligence, Image processing, Data mining, Camera self calibration, 3D reconstruction, Machine learning, Real-time rendering and Cryptography.

Hassan Satori

Hassan Satori is Associate Professor, Department of Computer Science, Faculty of Sciences, Dhar El Mahraz, Sidi Mohammed Ben Abdellah University -Fez- Morocco. He is Ph.D in Speech recognition and Signal Processing in 2009. He received a MSc in Physics from the Mohamed Premier University, and a Ph.D in Nanotechnology and computer simulation, also from the same University, in 2001. From 2001-2002 he was a postdoctoral fellow in Physics, mathematical modeling and computer simulation at the Chemnitz University of Technology, Germany, and was then a postdoctoral fellow in computer simulation at Department of Interface Chemistry and Surface Engineering, Max-Planck-Institut, Düsseldorf, Germany, (2003-2005). Dr. Hassan SATORI has large academic teaching and research experience in the fields.

Khalid Satori

Khalid Satori received the PhD degree from the National Institute for the Applied Sciences INSA at Lyon in 1993. He is currently a professor of computer science at SMBA-Fez University. He is the director of the LISAC Laboratory. His research interests include real-time rendering, Image-based rendering, virtual reality, biomedical signal, camera self calibration, genetic algorithms and 3D reconstruction.