



Modified marine predators algorithm for feature selection: case study metabolomics

Mohamed Abd Elaziz^{1,8,9,10} · Ahmed A. Ewees^{2,3} · Dalia Youstri⁴ ·
Laith Abualigah^{5,6} · Mohammed A. A. Al-qaness^{7,11} 

Received: 3 March 2021 / Revised: 6 December 2021 / Accepted: 11 December 2021 /

Published online: 19 January 2022

© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

Abstract

Feature selection (FS) is a necessary process applied to reduce the high dimensionality of the dataset. It is utilized to obtain the most relevant information and reduce the computational efforts of the classification process. Recently, metaheuristics methods have been widely employed for various optimization problems, including FS. In the current study, we present an FS method based on a new modified version of the marine predators algorithm (MPA). In the developed MPASCA model, the sine–cosine algorithm (SCA) is utilized to improve the search ability, which works as a local search of the MPA. To evaluate the performance of the MPASCA algorithm, extensive experiments were carried out using 18 UCI datasets. Moreover, the metabolomics dataset is used to test the proposed method as a real-world application. Furthermore, we implemented extensive comparisons to several state-of-art methods to verify the efficiency of the MPASCA. The evaluation outcomes showed that the MPASCA has significant performance, and it outperforms the compared methods in terms of classification measures.

Keywords Feature selection · Metaheuristics · Marine predators algorithm (MPA) · Sine–cosine algorithm (SCA) · Metabolomics dataset

1 Introduction

Lately, data have been exponentially growing and become an essential source for several science domains, such as data sciences, data mining, and medical sciences [1]. However, the rapid increase in data size causes several problems, such as high dimensionality, irrelevant data, or noise. Therefore, such problems reduce the machine learning analysis accuracy rate and elevation computational costs [2]. Besides, all conventional machine learning classifier approaches are incapable of interacting with features included in used data [3]. Consequently, as mentioned earlier, these problems influence the production of data science and data mining domains because they are mostly utilized machine learning classifier approaches. Therefore,

✉ Mohammed A. A. Al-qaness
alqaness@whu.edu.cn

Extended author information available on the last page of the article

feature selection (FS) is challenged to select a subset of useful features and eliminates unnecessary features [4,5].

Different FS techniques contribute to select the most useful features from the dataset [6]. FS aims to accomplish two basic objectives: decreasing the number of selected features and increasing the accuracy rates of the classifier [7,8]. FS approaches are categorized into filter, embedded, and wrapper, and the main differences among them are in the procedure of selecting the features subset [9]. In wrapper-based methods, a training method or a classification technique is utilized to assess the selected features. Filter-based methods assess the selected features subset based on the properties of the given dataset. In comparison with wrapper-based methods, filter-based methods are faster, as wrapper-based methods have a high consumption time in determining the classifier accuracy values. Furthermore, the wrapper-based methods can find a useful feature subset that satisfies the classifier than the filter-based methods. Fusion methods have the benefit of low computational time but usually are particular to machine learning [10,11].

The FS problem is known as procedures of finding a minimal reduct, which is defined as an NP-hard problem. The standard used solution to find such a minimal reduct is to produce all potential reducts quickly and then select a minimum number of elements in a set [12]. It is an expensive solution, which is only useful in case of a simple data. For most selection purposes, only one smallest reduct is wanted, thus, all considerations included in determining the rest are targetless. Hence, alternative solutions are needed for large data [13]. Moreover, FS methods are widely applied in different intelligent systems, such as in data mining [10,14,15], text clustering [16,17], image processing [18–20], intrusion detection [21–23], data classification [24,25], human activity recognition [26], sentiment analysis [27,28], cancer detection [29,30], disease detection [31,32], power systems [33,34], design problems [35,36], text classification [37,38], and many more other application areas [39–41].

In literature, different optimization approaches have been presented for solving FS problems [8]. However, the optimization methods became essential and useful for the FS community, as they are recognized for their ability to search for solutions. Examples of such methods are whale optimization algorithm [42], artificial fish swarm (AFS) [43], social spider optimization (SSO) [44], moth-flame optimization algorithm (MFO) [8], salp swarm algorithm (SSA) [4], particle swarm optimization (PSO) [45], water wave optimization algorithm (WWO) [46], Harris Hawks optimizer (HHO) [47], fruit fly optimization algorithm (FOA) [48], gray wolf optimizer algorithm (GWO) [6], grasshopper optimization algorithm (GGA) [49], ant colony optimization (ACO) [50], aquila optimizer (AO) [51], arithmetic optimization algorithm (AOA) [52], and bat algorithm (BA) [53].

When comparing these optimization methods, such as PSO, GA, GOA, ALO, PSO, and WWO, several limitations have been found. PSO does not act efficiently for large optimization problems and low population diversity. GA has shortcomings such as the dilemma of crossover operator, which is quickly shifted the candidate solutions through the optimization process. GOA has shortcomings such as slow-motion, poor grasshopper moves, and early convergence. ALO has shortcomings such as the problems of early convergence and local optima. GWO has shortcomings such as strangling in the local area and early convergence. Recently, Faramarzi et al. [54] presented a new nature-inspired optimization algorithm, called marine predators algorithm (MPA) for solving various engineering optimization problems. The primary motivation of MPA is the general foraging approach, which are Lévy and Brownian motions of animals that naturally prey on others in the ocean along with optimal collision rate method in natural interaction between animals and prey. MPA is proceeding ahead of the rules that biological rule in optimal foraging approach and collisions rate plan between animals and prey in the marine environment. The MPA's performance is evaluated on 29

CEC-BC-2017 benchmark functions and five engineering problems. The evaluation of MPA showed that MPA got better results in some cases and comparative results in other cases. These obtained results motivate us to apply MPA to solve the FS problems; it is never applied to solve these kinds of problems since it was introduced. Moreover, MPA has less tuning parameters and is straightforward to implement. It proved its ability to tackle small and large optimization problems and distinguished by its features: flexibility and robust stochastic behavior.

Mirjalili [55] proposed a population-based method called sine–cosine algorithm (SCA) to solve various optimization tasks. It performs multiple initial stochastic solutions and orders them to shift toward the near-optimal solution utilizing a mathematical representation depending on sine and cosine functions. Several adaptive and random variables are blended to the SCA to ensure search methods: exploitation and exploration. It has an excellent ability in exploitation search over the given search space.

In this study, an efficient FS method is proposed based on a new improved version of the marine predators algorithm (MPA) using the operators of the sine–cosine algorithm (SCA), called MPASCA. Generally, MPA suffers from some shortcomings, such as permutate convergence and strangling in the local optimum. So, the proposed MPASCA improves the conventional MPA's performance by added additional components from the SCA algorithm. This proposed procedure is founded to tackle the weaknesses of conventional MPA. Thus, the cooperation between the used algorithms improves the behavior of convergence ability. Several datasets with different characterized are employed to evaluate the performance of the MPASCA. The performance of the MPASCA is compared to the conventional MPA, SCA, and other well-known optimization algorithms. The obtained results proved that the MPASCA achieved better results compared to other similar optimization methods. Moreover, statistical significance tests are conducted to further evaluate the MPASCA method's effectiveness compared to other optimization methods using the same datasets. The results proved that the MPASCA obtained significantly better results.

The main contributions in this paper have been pointed as follows:

- A new fusion version of the marine predators algorithm (MPA) and sine–cosine algorithm (SCA) is adapted, called MPASCA.
- MPA is combined with SCA to solve its local optima problem and enhance the current best solution.
- MPASCA wrapper feature selection method is proposed in this paper.
- Comprehensive comparisons and statistical tests are performed, such as worst, mean, and best fitness, classification accuracy, standard deviation, and computational time.
- MPASCA method is compared with other optimization algorithms, and the experimental results showed that the proposed MPASCA gave superior results compared to different algorithms.

The rest of this study is presented as follows. A brief literature review is presented in Sect. 2. Section 3 discusses the background of MPA and SCA. Section 4 illustrates the proposed MPASCA method. Section 5 shows the experiments and results. The conclusion and potential future works are presented in Sect. 6.

2 Related works

Feature selection is a necessary pre-processing action in several creative systems such as text clustering, intrusion detection, text categorization, disease prediction, and text summarization

[56]. It also appears as one of the most popular and challenging jobs in machine learning domain [57]. A well-known approach for FS is by using the Monte Carlo and rough set theory (RST), such as kruczyk et al. [58]. Also, Bouzayane and Saad [59] used rough set theory to build a periodic multicriteria classification approach. Kifah et al. [60] developed an FS approach using a double treatment iterative improvement algorithm with the RST. Li et al. [61] developed a new FS method based on Monte Carlo approach for leukemia stem cell expression signatures Identification. Recently, different optimization approaches have been successfully employed to address these problems, as follows.

An improved FS approach is proposed based on applying the SSA to find the optimal features subset in wrapper-method, called ISSA [4]. Two enhancements are incorporated into the conventional SSA to tackle its weaknesses through dealing with FS problems. The first development involves the advantage of Opposition-based Learning at the beginning phase of SSA to increase its population diversity, where the second enhancement involves improving and applying a new local exploration search with SSA to enhance its exploitation ability. Eighteen datasets were used to validate the effectiveness of the improved SSA (ISSA). Moreover, the performance of the ISSA is analyzed with other similar optimization algorithms. The results demonstrated that the ISSA got better results in convergence curves, accuracy, fitness values, and feature reduction.

Agrawal et al. [62] presented a new FS approach based on adding an amalgamation of the Quantum ideas to the conventional WOA, called QWOA. The introduced method improved the search strategies (exploitation and exploratory) of the conventional WOA by utilizing the quantum bit design over the solutions of the WOA population and a quantum rotation as a mutation operator. Furthermore, improved mutation and crossover operators were proposed for the quantum exploration process. The effectiveness of QWOA was tested on fourteen datasets from various domains, and the results were analyzed and compared with the WOA and several similar optimization algorithms. The obtained results proved that the proposed method (QWOA) got a superior performance. Moreover, statistical tests are also employed, and it showed that the QWOA achieved better results compared to several similar metaheuristics.

An improved version of FOA is presented to tackle the implied weaknesses of the original FOA [48]. The proposed version added chaotic exploitation search mechanism and Gaussian mutation operator into the main procedure of the FOA to evade premature convergence problem and enhance the exploitative tendencies, respectively. Comprehensive comparisons had been developed utilizing different characterized datasets to verify the performance of the proposed methods. Experiment results proved that the best version of the FOA is the MCFOA; they were employed to FS. Also, evaluation results demonstrated that MCFOA got optimal classification accuracy.

A wrapper-based method was proposed to solve the FS problems using an improved SSA [57]. The proposed method merged a time-varying mechanism, Random Weight Network (RWN), and binary SSA, called TVBSSA. In the proposed method, TVBSSA was employed as an optimization search mechanism, where RWN was employed as a selection method. The objective function is a combination of three sub-objectives: reducing the reduction degree of the selected features, increasing the classification accuracy rates, and reducing the complexity of produced RWN models. Twenty datasets were used, and several existing methods were employed to test its effectiveness. Experiment results validated the capability of the proposed method was overwhelming other similar optimization algorithms.

More so, an FS method was proposed to find a subset of features that maximize classification accuracy values [13]. The proposed method applies a newly binary WWO algorithm along with a rough set theory (RST), called BWWO. Two experiments were conducted based

on the wrapper-based method, and a rough set-based process as a portion of the used objective function to validate the effectiveness of BWWO. In the first section, the performance of the RST methods was tested on sixteen various datasets. In the second section, the kNN method's performance to select a subset of features that improved the classification accuracy and reduced the number of selected features was evaluated on seventeen various datasets. The obtained results proved the effectiveness of BWWO in obtaining the smallest features subset that maximizes accuracy values. Moreover, the results of Friedman and Wilcoxon's ranking tests showed that the BWWO got significantly better results. An alternative FS method is proposed for finding a subset of features [8]. The proposed method, called OMFODE, improved the search efficiency of the conventional MFO by combining the OBL and differential evolution (DE) algorithm with the MFO. The OBL is employed to produce a better initial population which enhances the convergence ability of MFO; meantime, the DE is utilized to enhance the exploitation search of the MFO. Consequently, the OMFODE can avert from getting trapped in a local optimum, dissimilar to the conventional MFO. A set of experiments was used to evaluate its performance, and results demonstrated that it performed better than other optimization algorithms.

A new version of SSA is proposed for solving the FS, called ISSAFD based on enhancing the followers of the SSA using the SCA and Disrupt Operator (DO) [41]. This improvement supports to enhance the exploration stage and to avert from stuck in a local area. Furthermore, DO is utilized for the given solutions to enhance the population diversity and to keep the equilibrium between the search strategies (exploitation and exploration). Two different modifications of the SSA are produced based on the SCA, namely ISSALD and ISSAF. Experimental results are assessed on 20 datasets. The results showed that ISSAFD had an excellent performance in all of specificity, sensitivity, accuracy, and the number of chosen features compared to other optimization algorithms.

A new GWO algorithm is proposed and combined with two-phase mutation operators to address the FS problem [6]. The sigmoid function was employed to convert into binary search space to meet binary characteristics. The two-phase mutation intensifies the exploitation search ability of the proposed method. The first mutation operator aims to decrease the size of features and keeping a high accuracy rate. While the second mutation operator is applied for adding extra informative features which improve the accuracy values. Comprehensive comparisons on 35 datasets were conducted and compared with other similar optimization approaches. Also, statistical analyses were implemented to verify its performance.

Mafarja and Mirjalili [42] presented two hybridization algorithms based on the WOA for solving FS problems. In the first proposal, the WOA is hybridized with Simulated Annealing (SA) for enhancing its exploitation stage. In the second proposal, the SA is used by MOA for improving the obtained-best solution in each iteration, which is the exploitation search ability. The performance of the hybridization algorithms was evaluated using 18 datasets and compared to other similar FS methods. The results proved the effectiveness of the hybridization algorithms in increasing the accuracy values, which guarantees the ability of the WOA to solve challenging FS problems.

3 Background

In this section, the basics of the marine predators algorithm and sine–cosine algorithm are introduced.

3.1 Marine predators algorithm (MPA)

The MPA is a recently proposed method inspired by the behavior of prey and predator in nature [54]. In MPA, predator and prey are considered as search agents, in which a predator looking for a prey when a prey is looking for its food. Similar to all metaheuristic techniques (MHs), MPA started by a set of random solutions as an initialization. After that, solutions are modified based on the main structure of the algorithm.

Therefore, initial solutions can be obtained depending on the search landscape as follows;

$$Z = LB + r_1 \times (UB - LB) \tag{1}$$

where the LB and UB refer to the lower and upper boundary in the search landscape. $r_1 \in [0, 1]$ is the random number.

In the initialization phase, two basic matrices must be defined, the matrix of the fittest predators (Elite matrix) and the prey matrix. The following formulas are employed for representing the Elite and prey matrices.

$$\text{Elite} = \begin{bmatrix} Z_{11}^1 & Z_{12}^1 & \dots & Z_{1d}^1 \\ Z_{21}^1 & Z_{22}^1 & \dots & Z_{2d}^1 \\ \dots & \dots & \dots & \dots \\ Z_{N1}^1 & Z_{N2}^1 & \dots & Z_{Nd}^1 \end{bmatrix}, \quad Z = \begin{bmatrix} Z_{11} & Z_{12} & \dots & Z_{1d} \\ Z_{21} & Z_{22} & \dots & Z_{2d} \\ \dots & \dots & \dots & \dots \\ Z_{N1} & Z_{N2} & \dots & Z_{Nd} \end{bmatrix}, \tag{2}$$

The principal target of the MHs is reaching for the optimal solutions via modifying the initial random set of solutions based on a list of steps regarding the algorithm structure. MPA follows three phases during updating the solutions that depend on velocity ratio among the predators and prey. The first phase can be considered when velocity ratio among the prey and predators is high while the unit and low-velocity ratios are the observable marks for the second and third phases. The details of each stage are discussed in the following.

3.2 Phase 1: high-velocity ratio

This phase applied for discovering the search landscape (exploration phase). The main feature for this stage is the high-velocity ratio among predators and prey, where the prey moves very fast while the best tactic for the predator is staying without move at all. This phase occurred at the first third of the total number of generations (i.e., $\frac{1}{3}t_{\max}$) where the prey position is updating using the following equations.

$$S_i = R_B \otimes (\text{Elite}_i - R_B \otimes Z_i), \quad i = 1, 2, \dots, n \tag{3}$$

$$Z_i = Z_i + P \cdot R \otimes S_i \tag{4}$$

where $R \in [0, 1]$ represent a vector of uniform random numbers, and $P = 0.5$ is a constant number, where R_B represents a random vector which represents Brownian motion. More so, \otimes is the process of element-wise multiplications.

3.3 Phase 2: unit velocity ratio

The second phase is the transition phase from the exploration to the exploitation stage, in which predators and prey are moving with nearly the same velocity searching for their foods.

This phase performed in the middle stage of the algorithm when $\frac{1}{3}t_{max} < t < \frac{2}{3}t_{max}$. In this case, the best tactic for predator is to follow Brownian, where the prey moves with Lévy flight. Accordingly, the population can be divided into two halves and applying Eqs. (5), (6) to emulate the motion of the first half and Eq. (9), (10) for the second half as defined below.

$$S_i = R_L \otimes (Elite_i - R_L \otimes Z_i), i = 1, 2, \dots, n \tag{5}$$

$$Z_i = Z_i + P.R \otimes S_i \tag{6}$$

where R_L represents random numbers that follow Lévy distributions. Eqs. (5), (6) are employed to the first half of the population that represents the exploitation, where the second half follows the following equations:

$$S_i = R_B \otimes (R_B \otimes Elite_i - Z_i), i = 1, 2, \dots, n \tag{7}$$

$$Z_i = Elite_i + P.CF \otimes S_i, CF = \left(1 - \frac{t}{t_{max}}\right)^{2\frac{t}{t_{max}}} \tag{8}$$

where CF represents a parameter which controls the step size of movements for predators.

3.4 Stage 3: low-velocity ratio

This phase is the last stage of the optimization process. During this phase, predator is moving faster than prey; this is why predator follows Lévy during updating its position. This process is performed at the last third of the iteration numbers ($t > \frac{2}{3}t_{max}$) and this formulated as:

$$S_i = R_L \otimes (R_L \otimes Elite_i - Z_i), i = 1, 2, \dots, n \tag{9}$$

$$Z_i = Elite_i + P.CF \otimes S_i, CF = \left(1 - \frac{t}{t_{max}}\right)^{2\frac{t}{t_{max}}} \tag{10}$$

3.5 Eddy formation and fish aggregating devices' effect (FADS)

The surrounded environment has a significant impact on the behave of the creature. Therefore for MPA, the authors considered the environmental issues such as the eddy formation or Fish Aggregating Devices (FADs) effects that can be mathematically modeled as follows:

$$Z_i = \begin{cases} Z_i + CF[Z_{min} + R \otimes (Z_{max} - Z_{min})] \otimes U & r_5 < FAD \\ Z_i + [FAD(1 - r) + r](Z_{r1} - Z_{r2}) & r_5 > FAD \end{cases} \tag{11}$$

where $FAD = 0.2$, and U represent a binary solution and this preformed by generating random solutions then they are converted into binary using threshold 0.2. $r \in [0, 1]$ represents a random number. r_1 and r_2 represent the indices of the prey.

3.6 Marine memory

Predators have a strong memory of the position where they have been productive in foraging. This feature is implemented by exporting memory to MPA, where the last best solutions of the previous iteration are saved and compared with the current solutions. After that, solutions can be updated depending on the best during the comparison stage.

3.7 Sine-cosine algorithm

In SCA, Mirjalili et al. [55] utilized the characteristics of sine and cosine trigonometric functions to attain qualified solutions for global optimization problems. The SCA structure includes some steps that start with the initialization phase, where a set of random solutions have been generated. The random solutions z have a set of N solutions, and each solution Z_i has dimension D_z as the number of the search agents is N , and the dimension of the studied problem is D . The initial values of the studied function Fit_i as well as computed based on the initial solutions Z_i . Mirjalili et al. [55] sorted the solutions to identify the best solutions Z_B and the corresponding Fit_B value to be based on while modifying the solution values across the number of iterations. The best solution vector and best function value have been modified each iteration to be a guide for the other search agents. Subsequently, the followed solutions update approach in the SCA can be represented as below:

$$Z_i = \begin{cases} Z_i + \sigma_1 \times \sin(\sigma_2) \times |\sigma_3 Z_B - Z_i| & \text{if } \sigma_4 < 0.5 \\ Z_i + \sigma_1 \times \cos(\sigma_2) \times |\sigma_3 Z_B - Z_i| & \text{if } \sigma_4 \geq 0.5 \end{cases} \quad (12)$$

where $|\cdot|$ and σ_i , $i = 1, 2, 3, 4$ represents the absolute value, and random variables produced in the interval of $[0, 1]$, respectively. The σ_2 is applied to assign the next direction of updating the solution (e.g., toward or outward Z_B), whereas σ_3 is employed to test whether the best solution Z_B stochastically maintains, and this occurred if $\sigma_3 > 1$. In addition, σ_1 is used to attain a smooth harmonization between the diversification and the intensification cores of the algorithm to certain the promising region, and this parameter can be modified at each iteration t based on the following equation Eq.(13) [55]:

$$\sigma_1 = \gamma - t \frac{\gamma}{t_{\max}} \quad (13)$$

where t_{\max} and γ represent the total number of iteration and the constant, respectively.

4 Proposed feature selection method

The structure of the proposed MPASCA is given in Fig. 1. The proposed MPASCA uses SCA operators to enhance the exploitation of MPA, which reflected on the quality of the obtained solution that represents the optimal subset of relevant features. To reach this main objective, the developed MPASCA begins by setting the initial value for a set of N solutions inside the search space that is bound by lower and upper boundaries (here, 0 and 1, respectively). Then compute the fitness value for each solution after converting it to a binary solution, which determines which features are relevant and which must be removed. Thereafter, the best solutions are allocated, and the other solutions can be updated using either the operators of SCA or MPA. The previous steps are implemented until they reached the end of the iterations. The details of each step are discussed with more information below.

4.1 First step: generating solutions

In this step, MPASCA starts by defining the initial value for each parameter and split the data into training and testing set. Followed by using Eq. (14) to set the initial value for each solution

$$Z_{i,j} = LB_j + r_1 \times (UB_j - LB_j), \quad j = 1, 2, \dots, D, \quad i = 1, 2, \dots, N \quad (14)$$

where LB_j and UB_j are the limits of search domain at j th dimension. N is the total number of solutions.

4.2 Second step: updating solutions

This step starts by converting each solution Z_i to binary using the following equation.

$$BZ_{ij} = \begin{cases} 1 & \text{if } Z_{ij} > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

The main aim of using Eq. (15) is to convert the solution from real-value to discrete that suitable for FS problem. Thereafter, the quality of the current solution is evaluated using the kNN classifier that depends on the training set with removing the features that corresponding to zeros in BZ_i . This process (i.e., compute fitness value Fit_i) is defined as:

$$Fit_i = \lambda \times \eta_i + (1 - \lambda) \times \left(\frac{|BZ_i|}{Dim} \right) \tag{16}$$

In Eq. (16), $\lambda \in [0, 1]$ is a uniform random number which is employed to balance between the ratio of selected features ($\frac{|BZ_i|}{Dim}$) and the classification error (η_i). For clarity, if we assume that the number of features of the dataset is 5, so this means that the dimension of the current solution Z_i is 5. In case $Z_i = [0.1 \ 0.6 \ 0.7 \ 0.9 \ 0.2]$, so, by using Eq. (16) the $BZ_i = [0 \ 1 \ 1 \ 1 \ 0]$. These Boolean values lead us to determine the irrelevant features (i.e., first and fifth) that will be removed. As well as the relevant features that corresponding to ones (i.e., second, third, fourth), and they will be used to train the classifier.

The next process is to determine the value of the best solutions and update the solutions using the operators of MPA or SCA. This achieved by computing the probability Pr_i for the current solution Z_i in the exploitation stage as:

$$Pr_i = \frac{Fit_i}{\sum_{i=1}^N Fit_i} \tag{17}$$

Then Z_i is updated using Eq. (18).

$$Z_i = \begin{cases} \text{operators of MPA} & Pr_i \geq rs \\ \text{operators of SCA} & \text{otherwise} \end{cases} \tag{18}$$

where

$$rs = \min(Pr_i) + rand \times (\max(Pr_i) - \min(Pr_i)), \text{ rand} \in [0, 1] \tag{19}$$

In Eq. (18), the value of $r1$ is updated in dynamic environment based on Pr_i and this provides MPASCA by a suitable tool to avoid the problem of fixed the value of $r1$. This leads to enhance solutions and to reduce the time needed to find the value of $r1$ manually.

4.3 Step three: stop conditions

In this step, the terminal conditions are checked, and if they are not satisfied, then the previous step (i.e., updating process) is implemented again. Followed by reducing the testing set depending on the best solution features that correspond to ones and remove other features. Then apply the testing set to the KNN classifier and compute the target of the testing set with computing the performance.

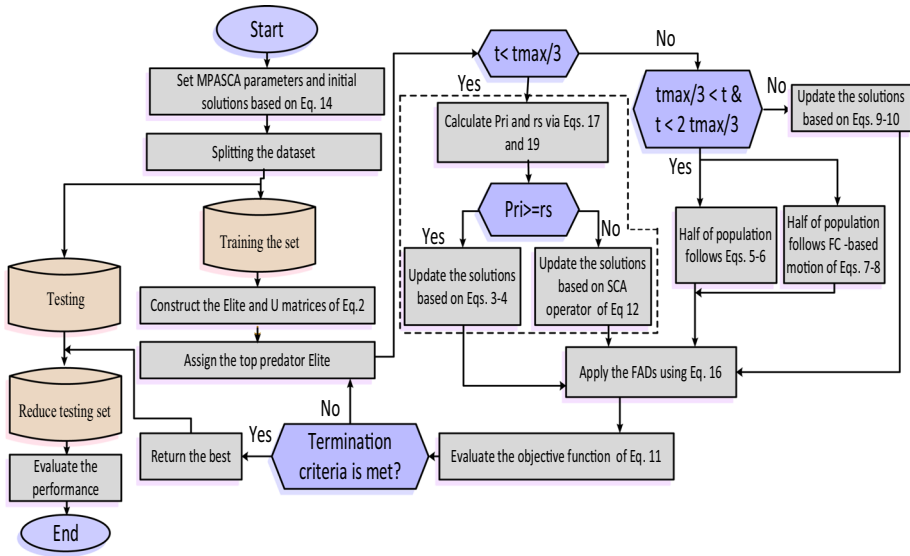


Fig. 1 The structure of MPASCA FS method

4.4 Computational complexity

In this section, the computational complexity of the proposed method is presented. This calculation proved the time needed by the proposed method to solve any problem, which is given as follows.

The real-time complexity of the proposed MPASCA is given. The time complexity of the proposed MPASCA is normally calculated using three main parts: initializing process for the first solutions, determining the fitness functions, and renewing solutions. Suppose that N is the number of the practiced solutions, $O(N)$ is the computational complexity in this part. The computational complexity of the second part for the updating processes is $O(T \times N) + O(T \times N \times \text{Dim})$, which involves studying for the best solutions and renewing them, where the highest number of iterations is named T . This approach is for all of the used search methods; MPA and SCA. The dimension of the provided problem is defined Dim . Thus, the total computational complexity of the proposed method (MPASCA) is $O(N \times (T \times \text{Dim} + 1))$.

5 Results and discussion

5.1 Data description

To validate the developed MPASCA, 18 UCI datasets [63] are used. The descriptions of these datasets are presented in Table 1. It can be seen from this table that those datasets are collected from different fields; also, they have different numbers of features and instances.

Table 1 Datasets description

Datasets	Number of features	Number of instances	Number of classes	Data category
Breastcancer (D1)	9	699	2	Biology
BreastEW(D2)	30	569	2	Biology
CongressEW(D3)	16	435	2	Politics
Exactly(D4)	13	1000	2	Biology
Exactly2(D5)	13	1000	2	Biology
HeartEW(D6)	13	270	2	Biology
IonosphereEW(D7)	34	351	2	Electromagnetic
KrvskpEW(D8)	36	3196	2	Game
Lymphography(D9)	18	148	2	Biology
M-of-n(D10)	13	1000	2	Biology
PenglungEW(D11)	325	73	2	Biology
SonarEW(D12)	60	208	2	Biology
SpectEW(D13)	22	267	2	Biology
Tic-tac-toc(D14)	9	958	2	Game
Vote(D15)	16	300	2	Politics
WaveformEW(D16)	40	5000	3	Physics
WineEW(D17)	13	178	3	Chemistry
Zoo(D18)	16	101	6	Artificial
Orlraws10P(DS_{19})	10304	100	10	Life
SMK-CAN-187(DS_{20})	19993	187	2	Face image
TOX-171(DS_{21})	5748	171	4	Life
warpAR10P(DS_{22})	2420	210	10	Life

5.2 Performance measures

This section presents six performance measures used to evaluate the performance of the proposed method. These measures are the averages of the accuracy, the standard deviation (Std), the fitness value, the minimum of the fitness value (Min), the maximum of the fitness value (Max), the selected features. In order to calculate the average of the performance measures, all algorithms are applied 30 independent runs.

- Accuracy (Acc): This measure calculates the ratio of the corrected classified data. It is calculated as in Eq. (20).

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \quad (20)$$

where TN, TP, FN, and FP denotes the true negative, the true positive, the false negative, and the false positive, respectively.

- Fitness value (FV): This measure evaluates the performance of the methods based on the used fitness function as in Eq. (16).
- Maximum of the fitness value (Max): This measure records the maximum value obtained by the fitness function for each algorithm.

$$\text{Max} = \max_{1 \leq i \leq N} \text{Fit}_b^i \quad (21)$$

Table 2 Results of the fitness means of the proposed method and the compared algorithms

Dataset	MPA	MPASCA	HHO	HGSO	WOA	GWO	GA	SSA	SCA
D1	0.0724	0.0632	0.0782	0.0601	0.0738	0.0679	0.1018	0.0552	0.0555
D2	0.0400	0.0501	0.0836	0.0911	0.0699	0.0809	0.1280	0.0777	0.0638
D3	0.0377	0.0307	0.0364	0.0302	0.0738	0.1075	0.1018	0.0895	0.0373
D4	0.0501	0.0506	0.0607	0.0852	0.1598	0.1415	0.1923	0.1008	0.0467
D5	0.2615	0.2155	0.2155	0.2902	0.2170	0.1998	0.3306	0.2743	0.2557
D6	0.1297	0.1555	0.1555	0.1301	0.2160	0.2038	0.1958	0.1450	0.1957
D7	0.0809	0.0415	0.0415	0.0857	0.0993	0.0817	0.1206	0.0831	0.0352
D8	0.0656	0.0608	0.0739	0.0949	0.0971	0.0955	0.1148	0.0947	0.0984
D9	0.0919	0.0686	0.1058	0.0809	0.1287	0.1564	0.1818	0.0835	0.0938
D10	0.0497	0.0487	0.0540	0.0568	0.1176	0.0998	0.1179	0.0999	0.0491
D11	0.0739	0.0121	0.0121	0.0239	0.0408	0.0489	0.2022	0.0570	0.0546
D12	0.0777	0.0570	0.0690	0.0832	0.0673	0.0965	0.0890	0.1317	0.0571
D13	0.1456	0.1584	0.1584	0.1115	0.2336	0.2353	0.2047	0.2284	0.1568
D14	0.1981	0.2219	0.2360	0.2228	0.2572	0.2548	0.2279	0.2461	0.2148
D15	0.0436	0.0390	0.0390	0.0465	0.0457	0.0533	0.1052	0.0823	0.0568
D16	0.2639	0.2581	0.2662	0.2542	0.2996	0.3026	0.3075	0.3027	0.2568
D17	0.0385	0.0429	0.0429	0.0422	0.0699	0.0571	0.0878	0.0508	0.0267
D18	0.0167	0.0271	0.0271	0.0388	0.0533	0.0660	0.0563	0.0429	0.0196
<i>High-dimensional datasets</i>									
DS19	0.0011	0.0002	0.0005	0.0454	0.1295	0.0475	0.0927	0.0024	0.0214
DS20	0.2434	0.1906	0.2328	0.1763	0.4322	0.2858	0.2234	0.1796	0.2931
DS21	0.0318	0.0240	0.0792	0.1122	0.2052	0.1338	0.1005	0.0242	0.1604
DS22	0.4253	0.2915	0.1138	0.3011	0.5679	0.5905	0.3356	0.1844	0.3395

The significance of bold indicates the best results

- Minimum of the fitness value (Min): This measure records the minimum value obtained by the fitness function for each algorithm.

$$Min = \min_{1 \leq i \leq N} Fit_b^i \tag{22}$$

- Selected features: This measure records the number of selected features obtained by each algorithm.
- Standard deviation (Std): This measure evaluates the stability of an algorithm over different runs. It can be calculated as in Eq. (23).

$$Std = \sqrt{\frac{1}{N} \sum_{i=1}^N (Y_i - Y_{avg})^2} \tag{23}$$

where N is the number of runs. Y_i indicates the given fitness value. Y_{avg} is the mean of all values for the given algorithms. Fit_b^i indicates the best FV at the i th run.

Table 3 Results of the STD of the fitness value of the proposed method and the compared algorithms

Dataset	MPA	MPASCA	HHO	HGSO	WOA	GWO	GA	SSA	SCA
D1	0.0013	0.0053	0.0059	0.0018	0.0113	0.0061	0.0120	0.0053	0.0045
D2	0.0068	0.0087	0.0087	0.0058	0.0115	0.0114	0.0067	0.0081	0.0087
D3	0.0062	0.0057	0.0064	0.0019	0.0153	0.0196	0.0119	0.0087	0.0000
D4	0.0077	0.0057	0.0211	0.0330	0.0963	0.0810	0.0823	0.0418	0.0020
D5	0.0236	0.0022	0.0025	0.0417	0.0262	0.0049	0.0164	0.0164	0.0116
D6	0.0101	0.0175	0.0175	0.0182	0.0225	0.0319	0.0105	0.0240	0.0202
D7	0.0110	0.0128	0.0128	0.0165	0.0177	0.0094	0.0091	0.0174	0.0079
D8	0.0080	0.0097	0.0104	0.0079	0.0150	0.0141	0.0106	0.0092	0.0070
D9	0.0216	0.0256	0.0256	0.0134	0.0597	0.0255	0.0271	0.0132	0.0297
D10	0.0047	0.0042	0.0148	0.0236	0.0410	0.0416	0.0425	0.0384	0.0069
D11	0.0148	0.0044	0.0044	0.0116	0.0314	0.0266	0.0023	0.0025	0.0357
D12	0.0164	0.0164	0.0213	0.0170	0.0128	0.0202	0.0106	0.0168	0.0184
D13	0.0097	0.0213	0.0213	0.0152	0.0072	0.0177	0.0119	0.0158	0.0163
D14	0.0006	0.0059	0.0085	0.0093	0.0180	0.0177	0.0231	0.0188	0.0035
D15	0.0070	0.0043	0.0050	0.0053	0.0168	0.0201	0.0232	0.0173	0.0147
D16	0.0135	0.0111	0.0111	0.0103	0.0174	0.0115	0.0085	0.0102	0.0121
D17	0.0058	0.0068	0.0068	0.0097	0.0139	0.0117	0.0161	0.0091	0.0145
D18	0.0039	0.0022	0.0056	0.0035	0.0119	0.0157	0.0075	0.0052	0.0023
<i>High-dimensional datasets</i>									
DS19	0.0005	0.0000	0.0001	0.0450	0.0002	0.0005	0.0001	0.0041	0.0000
DS20	0.0182	0.0226	0.0402	0.0520	0.0135	0.0006	0.0134	0.0379	0.0120
DS21	0.0112	0.0147	0.0176	0.0196	0.0136	0.0146	0.0259	0.0242	0.0216
DS22	0.0340	0.0275	0.0159	0.0531	0.0216	0.0230	0.0193	0.0341	0.0043

The significance of bold indicates the best results

5.3 Computational results

In this section, the MPASCA is evaluated using 18 datasets and it is compared to eight optimization algorithms, including MPA, Henry gas solubility optimization (HGSO) [64], Harris hawks optimization (HHO) [65], WOA [66], GWO [67], genetic algorithm (GA) [68], SSA [24], and SCA. The comparison is based on the average of the fitness function value, STD, Min, Max, and the selected features measures besides the accuracy of the classification phase.

The parameter setting of these algorithms is set as the recommended setting in their original studies. In addition, the common parameters such as $N = 15$, and $t_{max} = 50$. Each method is performed 30 independent runs to fair comparison, and the dataset is split into 80% training, and the rest is considered as a testing set.

The results are tabulated in Tables 2, 3, 4, 5, 7 and in Figs. 2, 3, 4. According to the average of fitness function values, all algorithms tried to minimize the fitness function value; therefore, in this measure, the MPASCA obtained the lowest fitness values in 33% of all datasets, especially in D8, D9, D10, and D12. The SCA was ranked second, followed by MPA with obtaining the lowest values in 22% of all datasets for each one. The HHO and HGSO showed slightly the same behavior, whereas the GWO and GA produced the worst results. In addition, the proposed MPASCA outperformed the other methods in 50% of

Table 4 Results of the Min measure of the proposed method and the compared algorithms

Dataset	MPA	MPASCA	HHO	HGSO	WOA	GWO	GA	SSA	SCA
D1	0.0719	0.0510	0.0692	0.0590	0.0590	0.0573	0.0830	0.0462	0.0526
D2	0.0291	0.0259	0.0437	0.0786	0.0491	0.0607	0.1107	0.0646	0.0458
D3	0.0313	0.0269	0.0291	0.0291	0.0560	0.0664	0.0769	0.0769	0.0373
D4	0.0462	0.0461	0.0462	0.0538	0.0463	0.0462	0.0615	0.0538	0.0462
D5	0.2372	0.2314	0.2147	0.2417	0.2102	0.1967	0.3032	0.2505	0.2327
D6	0.1128	0.1312	0.1128	0.1064	0.1731	0.1628	0.1692	0.1128	0.1205
D7	0.0616	0.0235	0.0147	0.0654	0.0742	0.0674	0.1048	0.0441	0.0576
D8	0.0503	0.0432	0.0585	0.0809	0.0660	0.0683	0.1003	0.0822	0.0645
D9	0.0699	0.0270	0.0689	0.0624	0.0471	0.1065	0.1322	0.0444	0.0556
D10	0.0462	0.0461	0.0462	0.0538	0.0615	0.0538	0.0615	0.0538	0.0462
D11	0.0222	0.0045	0.0068	0.0108	0.0031	0.0203	0.1982	0.0532	0.0091
D12	0.0467	0.0280	0.0433	0.0564	0.0481	0.0648	0.0750	0.1045	0.0300
D13	0.1242	0.1303	0.1348	0.0894	0.2182	0.1939	0.1773	0.2045	0.1394
D14	0.1979	0.2167	0.2219	0.2179	0.2354	0.2307	0.2120	0.2290	0.2120
D15	0.0550	0.0603	0.0338	0.0375	0.0363	0.0338	0.0625	0.0525	0.0275
D16	0.2380	0.2403	0.2507	0.2758	0.2730	0.2847	0.2951	0.2843	0.2354
D17	0.0308	0.0301	0.0328	0.0308	0.0462	0.0385	0.0692	0.0385	0.0154
D18	0.0125	0.0213	0.0288	0.0313	0.0375	0.0438	0.0438	0.0313	0.0488
<i>High-dimensional datasets</i>									
DS19	0.0005	0.0001	0.0004	0.0005	0.1293	0.0470	0.0926	0.0000	0.0214
DS20	0.2274	0.1681	0.1896	0.1188	0.4166	0.2852	0.2155	0.1477	0.2823
DS21	0.0188	0.0070	0.0589	0.0962	0.1895	0.1253	0.0747	0.0074	0.1394
DS22	0.3964	0.2252	0.0962	0.2434	0.5535	0.5768	0.3243	0.1561	0.3345

The significance of bold indicates the best results

the high-dimensional datasets. It obtained the first rank in both DS19 and DS21 followed by HHO and SSA in DS19 and DS21, respectively, and the MPASCA showed competitive results in the rest datasets. The worst results showed by the WOA algorithm.

In terms of STD measure, the MPASCA was the most stable algorithm; it recorded the lowest STD values in four databases as well as in the average of STD in all databases. The second and third ranks were obtained by MPA and SCA, respectively. While the HHO did not get the lowest STD in any dataset, but it was ranked fourth based on the average of STD in all datasets followed by HGSO, SSA, and GA. The largest STD values were obtained by WOA; therefore, it came in the last rank. These results also showed in the high-dimensional datasets.

The Min measure is used to check the behavior of an algorithm in reaching the smallest fitness values in each dataset. In this measure, the MPASCA achieved the smallest and best values in 8 out of 22 datasets, followed by SCA and HGSO, which obtained the best values in 3 datasets for each one, whereas the performance of the MPA and HHO was slightly similar. The GA did not obtain the best values in any datasets.

Based on the Max measure, the worst algorithm was GA. It obtained the worst values in 10 out of 22 datasets, followed by GWO with the worst values in 6 datasets, whereas the WOA obtained the worst fitness values in two datasets. The MPASCA showed good performance in this measure and was ranked first.

Table 5 Results of the Max measure of the proposed method and the compared algorithms

Dataset	MPA	MPASCA	HHO	HGSO	WOA	GWO	GA	SSA	SCA
D1	0.0719	0.0510	0.0692	0.0590	0.0590	0.0573	0.0830	0.0462	0.0526
D2	0.0291	0.0259	0.0537	0.0786	0.0491	0.0607	0.1107	0.0646	0.0458
D3	0.0313	0.0269	0.0291	0.0291	0.0560	0.0664	0.0769	0.0769	0.0373
D4	0.0462	0.0461	0.0462	0.0538	0.0463	0.0462	0.0615	0.0538	0.0462
D5	0.2372	0.2314	0.2147	0.2417	0.2102	0.1967	0.3032	0.2505	0.2327
D6	0.1128	0.1312	0.1128	0.1064	0.1731	0.1628	0.1692	0.1128	0.1205
D7	0.0616	0.0235	0.0147	0.0654	0.0742	0.0674	0.1048	0.0441	0.0576
D8	0.0503	0.0432	0.0585	0.0809	0.0660	0.0683	0.1003	0.0822	0.0545
D9	0.0699	0.0270	0.0689	0.0624	0.0471	0.1065	0.1322	0.0444	0.0556
D10	0.0462	0.0461	0.0462	0.0538	0.0615	0.0538	0.0615	0.0538	0.0462
D11	0.0222	0.0045	0.0068	0.0108	0.0031	0.0203	0.1982	0.0532	0.0091
D12	0.0467	0.0280	0.0433	0.0564	0.0481	0.0648	0.0750	0.1045	0.0300
D13	0.1242	0.1303	0.1348	0.0894	0.2182	0.1939	0.1773	0.2045	0.1394
D14	0.1979	0.2167	0.2219	0.2179	0.2354	0.2307	0.2120	0.2290	0.2120
D15	0.0550	0.0603	0.0438	0.0375	0.0363	0.0338	0.0625	0.0525	0.0255
D16	0.2380	0.2403	0.2507	0.2758	0.2730	0.2847	0.2951	0.2843	0.2304
D17	0.0308	0.0301	0.0328	0.0308	0.0462	0.0385	0.0692	0.0385	0.0153
D18	0.0125	0.0213	0.0288	0.0313	0.0375	0.0438	0.0438	0.0313	0.0488
<i>High-dimensional datasets</i>									
DS19	0.0014	0.0002	0.0006	0.0905	0.1296	0.0480	0.0928	0.0071	0.0214
DS20	0.2632	0.2134	0.2691	0.2199	0.4401	0.2864	0.2388	0.2215	0.3060
DS21	0.0386	0.0332	0.0893	0.1341	0.2133	0.1507	0.1266	0.0519	0.1825
DS22	0.4628	0.3255	0.1272	0.3480	0.5928	0.6171	0.3578	0.2222	0.3420

The significance of bold indicates the best results

In feature selection problems, the goal is to find the smallest number of features with saving the quality of these features. Therefore, the feature numbers are considered in this experiment. In this context, the smallest number is the best. As shown in Table 6, the SCA yielded the smallest feature numbers in 8 out of 22 datasets, followed by the MPASCA with the smallest feature numbers in 7 datasets, whereas the WOA, MPA, and HHO showed similar selection ratios to some extent.

The selected features by all algorithms are evaluated using KNN classifier to test their quality, whereas the smallest feature number is not always the best, but in some cases it includes low qualities features. Therefore, these features should be evaluated by a classifier.

In this part, the accuracy is used as an output of this phase. Table 7 shows the accuracy of each algorithm, overall datasets. The proposed method, MPASCA, obtained the highest accuracy in 78% of all datasets. The SSA came in the second rank and achieved the best accuracy in 27% of all datasets, followed by HHO with the highest accuracy in 23%. Both SCA and HGSO showed similar results to some extent and were ranked third and fourth, respectively. The MPA showed similar performance to GA in most datasets, whereas the WOA obtained the last rank and produced the lowest accuracy in all datasets.

In general, the MPASCA showed a good ability to select the most significant features in the selection phase and achieve the highest accuracy in the classification phase.

Table 6 Results of the selected features of the proposed method and the compared algorithms

Dataset	MPA	MPASCA	HHO	HGSO	WOA	GWO	GA	SSA	SCA
D1	2.9	2.2	1.8	3.4	2.4	2.6	4.8	3.3	3.1
D2	6.8	6.6	8.6	9.3	5.7	8.5	20.9	15.7	5.9
D3	4.9	3.1	2.7	2.7	3.9	5.1	10.7	8.5	1.2
D4	6.4	7.1	6.7	8.1	8.6	6.5	9.4	8.3	6.1
D5	4.0	1.3	1.0	5.7	1.3	1.4	9.4	8.2	1.0
D6	5.7	3.8	7.1	4.2	4.5	6.3	10.9	6.9	5.8
D7	9.4	6.9	9.5	8.9	8.5	8.8	28.1	19.1	7.7
D8	14.5	13.9	19.1	17.8	19.0	20.7	28.9	22.9	15.4
D9	9.1	4.4	8.6	7.1	4.1	8.0	14.0	9.9	5.5
D10	6.5	6.7	7.5	8.3	9.4	8.6	9.2	8.7	6.2
D11	58.1	39.5	37.2	77.7	40.5	106.8	267.3	185.3	65.1
D12	29.5	21.7	24.8	25.9	30.9	24.5	50.0	35.3	21.1
D13	8.1	6.0	8.3	7.7	3.7	6.5	16.9	12.6	1.9
D14	6.1	5.8	5.9	4.7	5.4	5.3	6.3	5.9	5.2
D15	4.7	3.2	4.9	5.2	1.9	4.2	11.1	6.9	3.9
D16	18.7	16.0	18.5	19.9	22.0	19.7	34.2	26.8	12.5
D17	5.0	4.4	6.2	5.3	6.3	5.5	9.5	6.6	4.7
D18	2.7	2.6	4.3	6.2	8.1	8.3	9.0	6.9	4.4
<i>High-dimensional datasets</i>									
DS19	1112.7	509.7	525.3	4448.3	636.3	2203.3	8702.7	5987.3	1317.0
DS20	4460.7	647.0	5119.7	2356.7	522.3	4926.0	16959.3	11709.0	2690.7
DS21	841.0	781.3	997.3	1102.7	2016.3	2322.7	4896.0	3453.3	906.7
DS22	376.7	386.0	800.0	520.0	426.3	670.0	2047.7	1575.7	221.3

The significance of bold indicates the best results

5.4 Statistical results

In this section, the Friedman test is used to rank the used methods in the previous experiments. Table 8 contains the Friedman results of the fitness values (FV), Min, and Max measures. From the table, we can conclude that the MPASCA algorithm obtained the best rank than other methods in all measures followed by SCA, MPA, HHO, and HGSO, respectively, whereas GA came in the last rank. In addition, Figs. 5, 6, 7 illustrate the post hoc test; the algorithms appear in the figure with this order: MPASCA, MPA, HHO, HGSO, WOA, GWO, GA, SSA. There are statistically significant differences between the proposed method and 50% of the compared algorithms in all measures.

5.5 Real application

In this section, we test the applicability of the proposed MPASCA when applied to the real metabolomics dataset. The metabolomics studies have a great role in early diagnosis of diseases where metabolomics can be defined as a branch of science which deals with the analysis of metabolites in the tissues of organisms in order to observe physiological functions and responses of the organism to different stimuli, such as infection, illness or drug usage.

Table 7 Accuracy measure results of the proposed method and the compared algorithms

Dataset	MPA	MPASCA	HHO	HGSO	WOA	GWO	GA	SSA	SCA
D1	0.956	0.959	0.964	0.975	0.948	0.957	0.946	0.979	0.979
D2	0.980	0.988	0.973	0.933	0.943	0.942	0.935	0.972	0.981
D3	0.977	0.992	0.964	0.985	0.945	0.916	0.961	0.959	0.956
D4	0.999	1.000	0.972	0.974	0.896	0.899	0.867	0.959	0.999
D5	0.744	0.759	0.745	0.726	0.770	0.790	0.713	0.765	0.785
D6	0.905	0.824	0.906	0.891	0.799	0.827	0.875	0.898	0.893
D7	0.941	0.977	0.927	0.912	0.917	0.938	0.958	0.970	0.968
D8	0.972	0.982	0.973	0.950	0.951	0.958	0.962	0.966	0.975
D9	0.954	0.973	0.913	0.932	0.882	0.876	0.884	0.969	0.889
D10	1.000	1.000	0.995	0.985	0.950	0.963	0.948	0.964	1.000
D11	0.938	0.980	0.956	1.000	0.969	0.982	0.867	1.000	0.973
D12	0.968	0.992	0.957	0.956	0.983	0.938	0.994	0.919	0.978
D13	0.879	0.851	0.931	0.915	0.759	0.772	0.858	0.810	0.856
D14	0.829	0.856	0.818	0.810	0.781	0.782	0.825	0.799	0.851
D15	0.962	0.960	0.967	0.984	0.962	0.970	0.960	0.957	0.971
D16	0.759	0.781	0.729	0.728	0.728	0.719	0.753	0.738	0.740
D17	0.983	1.000	0.998	0.998	0.976	0.983	0.983	1.000	0.996
D18	1.000	1.000	1.000	1.000	0.997	0.984	1.000	1.000	1.000
<i>High-dimensional datasets</i>									
DS19	1.000	1.000	1.000	0.950	1.000	1.000	0.950	1.000	1.000
DS20	0.754	0.789	0.754	0.807	0.737	0.737	0.807	0.807	0.702
DS21	0.981	1.000	0.933	0.914	0.981	0.905	0.943	0.990	0.867
DS22	0.545	0.885	0.876	0.667	0.636	0.394	0.679	0.800	0.654

The significance of bold indicates the best results

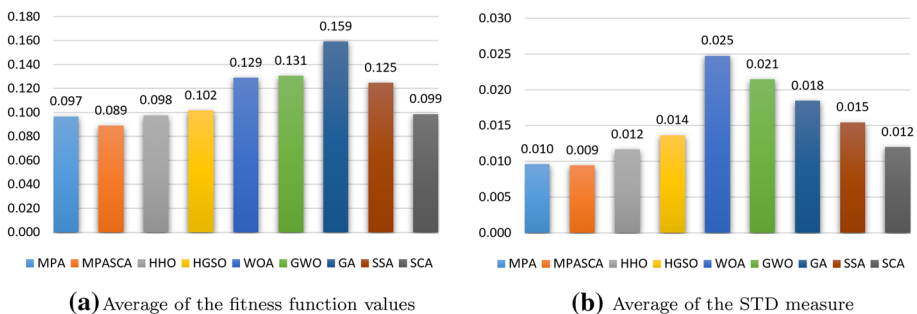


Fig. 2 Averages of the fitness function values and the STD measure in the experiment

Accordingly, the feature selection process of bioinformatics is an essential target. Identifying the genomic biomarkers between thousands or even millions of genes tested can help to define the exact mechanism behind a specific disease or can participate in the drug design process [69,70]. The considered three metabolomics datasets descriptions in this study can be given as below:

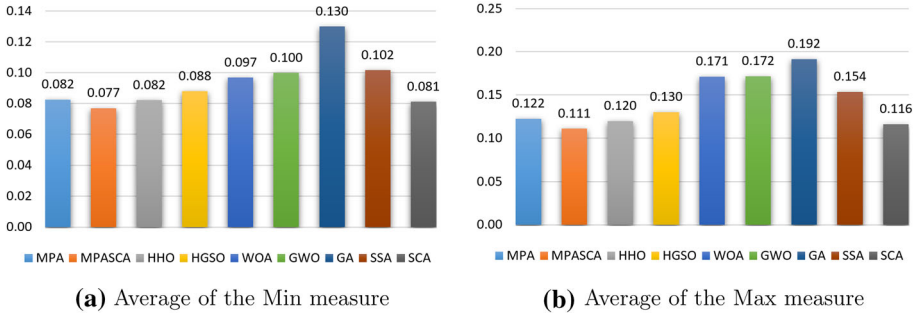


Fig. 3 Averages of the Min and Max measures in the experiment

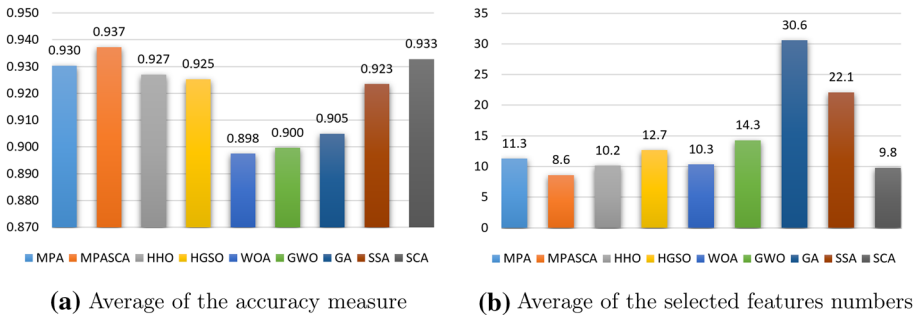


Fig. 4 Averages of the accuracy measure and the number of selected features in the experiment

1. *TBI dataset* In this traumatic brain injury (TBI) dataset, the features refer to the profiling of serum metabolic with/without cognitive impairment (CI). In general, there are 31 and 73 TBI patients with and without CI, respectively. There are 42 metabolites and 104 samples in this dataset, as given in Table 9.
2. *CHD dataset* The coronary heart disease (CHD) [71,72] contains 88 patients with 78 metabolites computed in the patients and healthy controls with UPLC-HRMS (this term refers to the “ultra-performance liquid chromatography-high resolution mass spectrometry”). The collected sample is from patients of the Hospital of Yunnan Province, China, where the 88 samples can be categorized into 21 and 16 patients with type 2 diabetes mellitus and without CHD, respectively. Moreover, there are 51 healthy cases without blood relationships, so there are three classes in the CHD dataset.
3. *ATR dataset* This dataset refers to the Acori Tatarinowii Rhizoma (ATR) collected from two provinces in China (Anhui and Sichuan). There are 21 samples from Sichuan and eight from Anhui. Additionally, the ATR contains 104 features (i.e., volatile compounds) and 29 samples. For more details see [70].

5.5.1 Results and discussion

Table 10 records the classification accuracy results (in this measure, the best accuracy (BestAcc) and the mean accuracy (MeanAcc) are presented) as well as the number of the selected features by each algorithm. The MeanAcc was calculated using 30 runs.

In the TBI dataset, the BestAcc shows that the performance of the MPASCA was similar to the MPA, HGSO, WOA, GWO, and SSA, whereas all these algorithms reached 100%

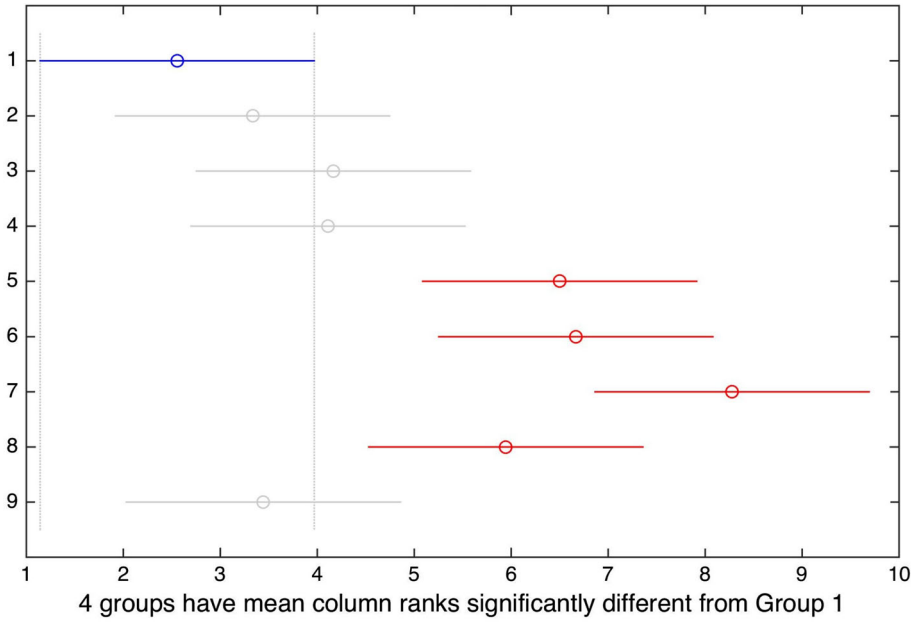


Fig. 5 Post hoc test for fitness function measure

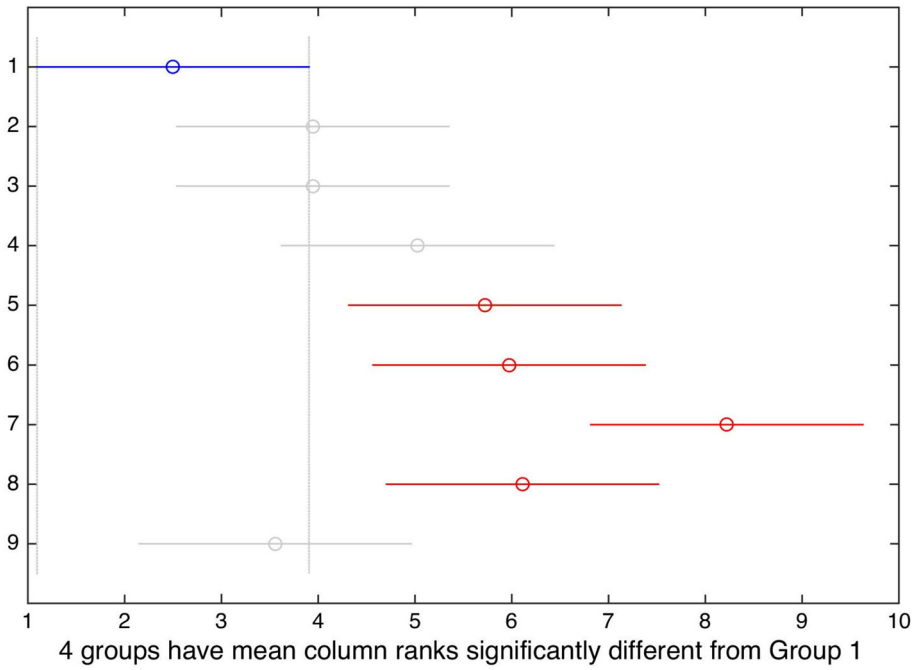


Fig. 6 Post hoc test for Min measure

Table 8 Results of Friedman test

	MPASCA	MPA	HHO	HGSO	WOA	GWO	GA	SSA	SCA
FV	2.56	3.33	4.17	4.11	6.50	6.67	8.28	5.94	3.44
Min	2.50	3.94	3.94	5.03	5.72	5.97	8.22	6.11	3.56
Max	2.61	3.67	3.78	4.17	6.81	7.19	7.97	5.86	2.94

The significance of bold indicates the best results

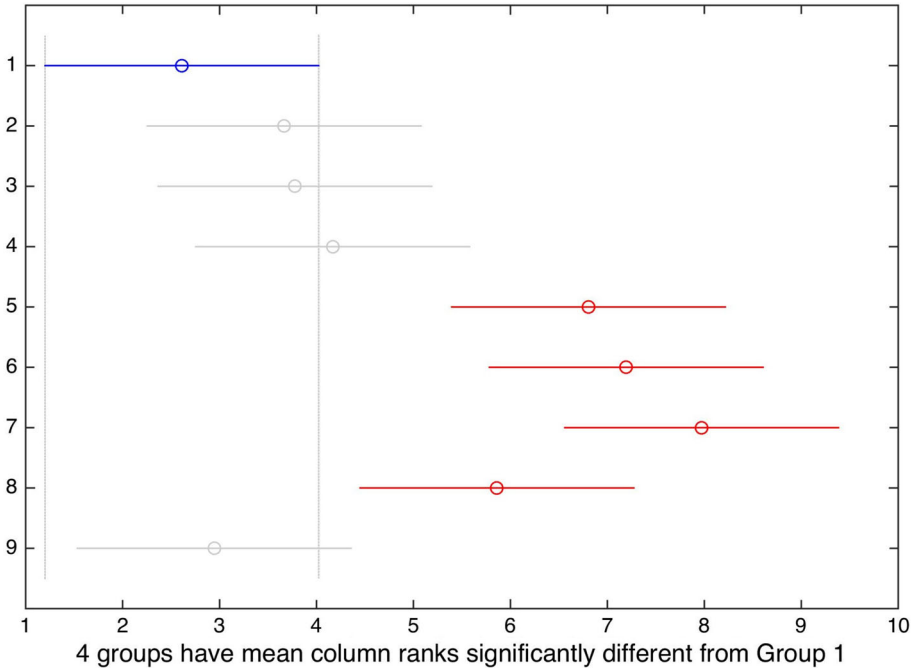


Fig. 7 Post hoc test for Max measure

Table 9 Description of the used datasets

Dataset	Features	Instances	Class
TBI	42	104	3
CHD	88	79	2
ART	104	29	3

of the classification accuracy, whereas the MeanAcc shows that the MPASCA, SCA, and WOA successfully classified all samples correctly, followed by HGSO. Regarding the feature selection results, the MPASCA was ranked third after HGSO and WOA, respectively, with small differences.

In the CHD dataset, the MPASCA achieved the first rank in the BestAcc and MeanAcc measures; it was obtained 0.90 and 0.86, followed by SCA with 0.89 and 0.83, respectively. The MPA was ranked third whereas, the last rank was obtained by the HGSO. In the feature

Table 10 Results of the real application experiment

Dataset	Measure	MPASCA	MPA	HHO	HGSO	WOA	SCA	GWO	GA	SSA
TBI	BestAcc	1.00	1.00	0.97	1.00	1.00	0.97	1.00	0.97	1.00
	MeanAcc	1.00	0.94	0.97	0.98	1.00	0.97	0.91	0.97	1.00
	Selected features	5.00	14.20	14.00	2.80	4.60	21.80	8.40	29.00	17.60
CHD	BestAcc	0.90	0.83	0.78	0.72	0.83	0.89	0.83	0.78	0.78
	MeanAcc	0.86	0.80	0.73	0.68	0.73	0.83	0.71	0.77	0.67
ART	Selected features	5.80	13.60	25.40	4.80	7.40	15.60	19.80	63.00	41.00
	BestAcc	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	MeanAcc	1.00	0.97	1.00	1.00	0.97	1.00	1.00	0.90	1.00
	Selected features	6.00	7.00	34.00	7.40	6.00	9.80	21.80	80.40	50.60

selection measure, the MPASCA came in the second order after the HGSO, but the quality of the obtained features by MPASCA was better than those of HGSO.

In the ART dataset, all algorithms performed equally in the BestAcc measure, and they classified all samples correctly. In MeanAcc measure, the MPASCA performed similarly to the HHO, HGSO, SCA, GWO, and SSA. The worst result was recorded by the GA algorithm. In the feature selection measure, the MPASCA and WOA obtained the smallest features followed by the MPA, HGSO, and SCA, respectively.

In general, the MPASCA showed good performance in all datasets and measures besides showed the best classification results, especially in the CHD dataset.

From all previous results and discussion, it can be noticed that the MPASCA as a feature selection method provides better accuracy than other comparison FS techniques. This was obtained by using the operators of SCA during the exploration phase of MPA. This makes the solutions are competitive in this phase and allow them to find the feasible region that contains the optimal solution. However, MPASCA has some limitations that influence the quality of solutions, such as its exploitation ability needs to be improved, and this can be achieved by using the local search method.

6 Conclusion

In this study, a modified version of the marine predators algorithm (MPA) is proposed as a feature selection (FS) method. In general, MPA established its ability to find the solution for global optimization and engineering problems. However, by analyzing its behavior during the optimization process, it has been found that its exploitation is weaker than its exploration, which can degradation its performance when applied to real-world applications, such as FS. Therefore, the operators of the sine–cosine algorithm (SCA) have been combined with MPA in the exploitation phase to enhance its ability to stagnate at the local point. To validate the performances of the proposed MPASCA, a set of experiments are conducted using 18 datasets. In addition, it compared to eight FS algorithms. The experimental results proved the superiority of the developed MPASCA overall other algorithms in terms of classification performance. Furthermore, we evaluate the MPASCA with real-world application, such as metabolomics dataset. Also, it showed better performance than compared algorithms.

In future works, the developed MPASCA model can be extended as a multi-objective feature selection and image segmentation. In addition, it can be applied to IoT as task scheduler techniques.

Funding The authors received no specific funding for this work.

Declarations

Conflict of interests The authors declare that they have no conflict of interests.

References

1. Cohen AM, Hersh WR (2005) A survey of current work in biomedical text mining. *Brief Bioinform* 6(1):57–71
2. Donoho DL et al (2000) High-dimensional data analysis: the curses and blessings of dimensionality. *AMS Math Challenges Lecture 1*(2000):32
3. Fan C, Xiao F, Zhao Y (2017) A short-term building cooling load prediction method using deep learning algorithms. *Appl Energy* 195:222–233
4. Tubishat M, Idris N, Shuib L, Abushariah MAM, Mirjalili S (2020) Improved salp swarm algorithm based on opposition based learning and novel local search algorithm for feature selection. *Expert Syst Appl* 145:113122
5. Jiang Y, Luo Q, Wei Y, Abualigah L, Zhou Y (2021) An efficient binary gradient-based optimizer for feature selection. *Math Biosci Eng MBE* 18(4):3813–3854
6. Abdel-Basset M, El-Shahat D, El-henawy I, de Albuquerque VHC, Mirjalili S (2020) A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. *Expert Syst Appl* 139:112824
7. De la Hoz E, De La Hoz E, Ortiz A, Ortega J, Martínez-Álvarez A (2014) Feature selection by multi-objective optimisation: application to network anomaly detection by hierarchical self-organising maps. *Knowl-Based Syst* 71:322–338
8. Elaziz MA, Ewees AA, Ibrahim RA, Lu S (2020) Opposition-based moth-flame optimization improved by differential evolution for feature selection. *Math Comput Simul* 168:48–75
9. Wang H, Jing X, Niu B (2017) A discrete bacterial algorithm for feature selection in classification of microarray gene expression cancer data. *Knowl-Based Syst* 126:8–19
10. Beniwal S, Arora J (2012) Classification and feature selection techniques in data mining. *Int J Eng Res Technol (ijert)* 1(6):1–6
11. Abualigah LM, Khader AT, Hanandeh ES (2018) A new feature selection method to improve the document clustering using particle swarm optimization algorithm. *J Comput Sci* 25:456–466
12. Wang X, Yang J, Teng X, Xia W, Jensen R (2007) Feature selection based on rough sets and particle swarm optimization. *Pattern Recogn Lett* 28(4):459–471
13. Ibrahim AM, Tawhid MA, Ward RK (2020) A binary water wave optimization for feature selection. *Int J Approx Reason* 120:74–91
14. Ibrahim RA, Oliva D, Ewees AA, Lu S (2017) Feature selection based on improved runner-root algorithm using chaotic singer map and opposition-based learning. In: *International conference on neural information processing*, pp 156–166. Springer
15. Al-Qaness MAA, Fan H, Ewees AA, Yousri D, Elaziz MA (2021) Improved anfis model for forecasting wuhan city air quality and analysis covid-19 lockdown impacts on air quality. *Environ Res* 194:110607
16. Li Y, Luo C, Chung SM (2008) Text clustering with feature selection by using statistical data. *IEEE Trans Knowl Data Eng* 20(5):641–652
17. Bharti KK, Singh PK (2016) Opposition chaotic fitness mutation based adaptive inertia weight bpsos for feature selection in text clustering. *Appl Soft Comput* 43:20–34
18. Lei X, Ma A (2021) Coarse-to-fine waterlogging probability assessment based on remote sensing image and social media data. *Geo-Spatial Inf Sci* 24(2):279–301
19. Al-qaness MAA, Abbasi AA, Fan H, Ibrahim RA, Alsamhi SH, Hawbani A (2021) An improved yolo-based road traffic monitoring system. *Computing* 103(2):211–230
20. Ibrahim RA, Elaziz MA, Ewees AA, Selim IM, Lu S (2018) Galaxy images classification using hybrid brain storm optimization with moth flame optimization. *J Astron Telesc Instrum Syst* 4(3):038001
21. Ambusaidi MA, He X, Nanda P, Tan Z (2016) Building an intrusion detection system using a filter-based feature selection algorithm. *IEEE Trans Comput* 65(10):2986–2998
22. Aljawarneh S, Aldwairi M, Yassein MB (2018) Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *J Comput Sci* 25:152–160
23. Tsang C-H, Kwong S, Wang H (2007) Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection. *Pattern Recogn* 40(9):2373–2391
24. Ibrahim RA, Ewees AA, Oliva D, Elaziz MA, Lu S (2019) Improved salp swarm algorithm based on particle swarm optimization for feature selection. *J Ambient Intell Humaniz Comput* 10(8):3155–3169
25. Abd Elaziz ME, Ewees AA, Oliva D, Duan P, Xiong S (2017) A hybrid method of sine cosine algorithm and differential evolution for feature selection. In: *International conference on neural information processing*, pp 145–155. Springer
26. Al-qaness MAA (2019) Device-free human micro-activity recognition method using wifi signals. *Geo-Spatial Inform Sci* 22(2):128–137
27. Yousefpour A, Ibrahim R, Hamed HNA (2017) Ordinal-based and frequency-based integration of feature selection methods for sentiment analysis. *Expert Syst Appl* 75:80–93

28. Shao Z, Sumari NS, Portnov A, Ujoh F, Musakwa W, Mandela PJ (2021) Urban sprawl and its impact on sustainable urban development: a combination of remote sensing and social media data. *Geo-Spatial Inf Sci* 24(2):241–255
29. Potie N, Giannoukakos S, Hackenberg M, Fernandez A (2020) Applying feature selection to improve predictive performance and explainability in lung cancer detection with soft computing. In: *Proceedings of the 53rd Hawaii international conference on system sciences*
30. Suji RJ, Rajagopalan SP (2016) Multi-ranked feature selection algorithm for effective breast cancer detection
31. Sharif M, Khan MA, Iqbal Z, Azam MF, Lali MIU, Javed MY (2018) Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection. *Comput Electron Agric* 150:220–234
32. Phadikar S, Sil J, Das AK (2013) Rice diseases classification using feature selection and rule generation techniques. *Comput Electron Agric* 90:76–85
33. Sawhney H, Jeyasurya B (2006) A feed-forward artificial neural network with enhanced feature selection for power system transient stability assessment. *Electr Power Syst Res* 76(12):1047–1054
34. Abedinia O, Amjady N, Zareipour H (2016) A new feature selection technique for load and price forecast of electrical power systems. *IEEE Trans Power Syst* 32(1):62–74
35. Chou T-S, Yen KK, Luo J (2008) Network intrusion detection design using feature selection of soft computing paradigms. *Int J Comput Intell* 4(3):196–208
36. Chen XW, Wasikowski M (2008) Fast: a roc-based feature selection metric for small samples and imbalanced data classification problems. In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp 124–132
37. Alhaj YA, Xiang J, Zhao D, Al-Qaness MAA, Elaziz MA, Dahou A (2019) A study of the effects of stemming strategies on arabic document classification. *IEEE Access* 7:32664–32671
38. Uysal AK, Gunal S (2012) A novel probabilistic feature selection method for text classification. *Knowl-Based Syst* 36:226–235
39. Wang S, Jia H, Abualigah L, Liu Q, Zheng R (2021) An improved hybrid aquila optimizer and harris hawks algorithm for solving industrial engineering optimization problems. *Processes* 9(9):1551
40. Wang S, Liu Q, Liu Y, Jia H, Abualigah L, Zheng R, Wu D (2021) A hybrid ssa and sma with mutation opposition-based learning for constrained engineering problems. *Comput Intell Neurosci*, 2021
41. Neggaz N, Ewees AA, Elaziz MA, Mafarja M (2020) Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection. *Expert Syst Appl* 145:113103
42. Mafarja MM, Mirjalili S (2017) Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing* 260:302–312
43. Chen Y, Zhu Q, Huarong X (2015) Finding rough set reducts with fish swarm algorithm. *Knowl-Based Syst* 81:22–29
44. El Aziz MA, Hassanien AE (2018) An improved social spider optimization algorithm based on rough sets for solving minimum number attribute reduction problem. *Neural Comput Appl* 30(8):2441–2452
45. Abualigah LM, Khader AT, Al-Betar MA, Alomari OA (2017) Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering. *Expert Syst Appl* 84:24–36
46. Macêdo F, Barbosa G, Neto A: A binary water wave optimization algorithm applied to feature selection. In: *Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional*, pp 448–459. SBC
47. Thaher T, Heidari AA, Mafarja M, Dong JS, Mirjalili S (2020) Binary harris hawks optimizer for high-dimensional, low sample size feature selection. In: *Evolutionary machine learning techniques*, pp 251–272. Springer
48. Zhang X, Xu Y, Yu C, Heidari AA, Li S, Chen H, Li C (2020) Gaussian mutational chaotic fruit fly-built optimization and feature selection. *Expert Syst Appl* 141:112976
49. Zakeri A, Hokmabadi A (2019) Efficient feature selection method using real-valued grasshopper optimization algorithm. *Expert Syst Appl* 119:61–72
50. Aghdam MH, Ghasem-Aghaee N, Basiri ME (2009) Text feature selection using ant colony optimization. *Expert Syst Appl* 36(3):6843–6853
51. Abualigah L, Yousri D, Abd Elaziz M, Ewees AA, Al-qaness MAA, Gandomi AH (2021) Aquila optimizer: a novel meta-heuristic optimization algorithm. *Comput Ind Eng* 157:107250
52. Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH (2021) The arithmetic optimization algorithm. *Comput Methods Appl Mech Eng* 376:113609
53. Alsalibi B, Abualigah L, Khader AT (2021) A novel bat algorithm with dynamic membrane structure for optimization problems. *Appl Intell* 51(4):1992–2017
54. Faramarzi A, Heidarinejad M, Mirjalili S, Gandomi AH (2020) Marine predators algorithm: a nature-inspired metaheuristic. *Expert Syst Appl*, pp 113377

55. Mirjalili S (2016) Sca: a sine cosine algorithm for solving optimization problems. *Knowl-Based Syst* 96:120–133
56. Abualgah LMQ (2019) Feature selection and enhanced krill herd algorithm for text document clustering. Springer, Berlin
57. Faris H, Heidari AA, Ala`M A-Z, Mafarja M, Aljarah I, Eshtay M, Mirjalili S (2020) Time-varying hierarchical chains of salps with random weight networks for feature selection. *Expert Syst Appl* 140:112898
58. Kruczyk M, Baltzer N, Mieczkowski J, Dramiński M, Koronacki J, Komorowski J (2013) Random reducts: a monte carlo rough set-based method for feature selection in large datasets. *Fund Inform* 127(1–4):273–288
59. Bouzayane S, Saad I (2020) A multicriteria approach based on rough set theory for the incremental periodic prediction. *Eur J Oper Res* 286(1):282–298
60. Kifah S, Abdullah S, Arajay YZ (2017) Solving feature selection problem using intelligent double treatment iterative composite neighbourhood structure algorithm. *Int J Comput Vis Robot* 7(3):255–275
61. Li JR, Lin L, Zhang Y-H, YaoChen X, Liu M, Feng KY, Chen L, Kong XY, Huang T, Cai Y-D (2020) Identification of leukemia stem cell expression signatures through monte carlo feature selection strategy and support vector machine. *Cancer Gene Ther* 27(1):56–69
62. Agrawal RK, Kaur B, Sharma S (2020) Quantum based whale optimization algorithm for wrapper feature selection. *Appl Soft Comput*, pp 106092
63. Frank A (2010) Uci machine learning repository. <http://archive.ics.uci.edu/ml>
64. Hashim FA, Houssein EH, Mabrouk MS, Al-Atabany W, Mirjalili S (2019) Henry gas solubility optimization: a novel physics-based algorithm. *Futur Gener Comput Syst* 101:646–667
65. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. *Futur Gener Comput Syst* 97:849–872
66. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
67. Ibrahim RA, Elaziz MA, Lu S (2018) Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization. *Expert Syst Appl* 108:1–27
68. Oreski S, Oreski G (2014) Genetic algorithm-based heuristic for feature selection in credit risk assessment. *Expert Syst Appl* 41(4):2052–2064
69. Mahadevan S, Shah SL, Slupsky CM, Marrie TJ, Saude E, Adamko DJ (2007) Feature selection and classification of metabolomic data using support vector machines. *IFAC Proc Vol* 40(4):43–48
70. Guang-Hui F, Yuan-Jiao W, Zong M-J, Yi L-Z (2020) Feature selection and classification by minimizing overlap degree for class-imbalanced data in metabolomics. *Chemom Intell Lab Syst* 196:103906
71. Guang-Hui F, Zhang B-Y, Kou H-D, Yi L-Z (2017) Stable biomarker screening and classification by subsampling-based sparse regularization coupled with support vector machines in metabolomics. *Chemom Intell Lab Syst* 160:22–31
72. Guang-Hui F, Feng X, Zhang B-Y, Yi L-Z (2017) Stable variable selection of class-imbalanced data with precision-recall criterion. *Chemom Intell Lab Syst* 171:241–250

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Mohamed Abd Elaziz received the B.S. and M.S. degrees in Computer science from the Zagazig University, in 2008 and 2011, respectively. He received Ph.D. degree in mathematics and computer science from Zagazig University, Egypt, in 2014. From 2008 to 2011, he was an Assistant lecturer in Department of Computer Science. He is an associate professor at Zagazig University, Egypt. He is the author of more than 200 articles. ABD ELAZIZ is one of the 2% influential scholars, which depicts the 100,000 top scientists in the world. His research interests include metaheuristic technique, security IoT, cloud computing machine learning, signal processing, image processing, and evolutionary algorithms.



Ahmed A. Ewees received his PhD degree from Damietta University, Egypt. Currently, He works as an associate professor of computer science at Damietta University, Egypt. He cosupervises master and PhD students, as well as leads and supervises various graduation projects. Ahmed published more than 100 scientific research papers in international journals and conferences. His research interests include machine learning, artificial intelligence text mining, natural language processing, image processing, and metaheuristic optimization techniques.



Dalia Yousri received the B.Tech. (Hons.), M.Tech., and Ph.D. degrees from the Faculty of Engineering, Fayoum University, Egypt, in 2011, 2016, and 2020, respectively. She is currently working as a Lecturer with the Faculty of Engineering, Fayoum University. She has published refereed manuscripts in the fields of optimization algorithms, photovoltaic applications, fuel cell, chaotic systems, and fractional calculus with some topics. Her research interests include the modifications of optimization algorithms, modeling and implementation of solar PV systems, fuel cell technologies, and fractional calculus topics. She has acted as a Reviewer for various reputed journals, such as IEEE Access, IET, Energy Conversion and Management (Elsevier), Applied Soft Computing, International Journal of Electronics and Communications, Artificial Intelligence Review (Springer), and Neural Computing and Applications.




Laith Abualigah is an Assistant Professor at the Computer Science Department, Amman Arab University, Jordan. He is also a distinguished researcher at the School of Computer Science, Universiti Sains Malaysia, Malaysia. He received his first degree from Al-Albait University, Computer Information System, Jordan, in 2011. He earned a Master's degree from Al-Albait University, Computer Science, Jordan, in 2014. He received a Ph.D. degree from the School of Computer Science in Universiti Sains Malaysia (USM), Malaysia, in 2018. According to the report published by Stanford University in 2020, Abualigah is one of the 2% influential scholars, which depicts the 100,000 top scientists in the world. Abualigah has published more than 100 journal papers and books, which collectively have been cited more than 4300 times (H-index = 32). His main research interests focus on Arithmetic Optimization Algorithm (AOA), Bio-inspired Computing, Nature-inspired Computing, Swarm Intelligence, Artificial Intelligence, Meta-heuristic Modeling, and Optimization Algorithms, Evolutionary Computations, Information Retrieval, Text clustering, Feature Selection, Combinatorial Problems, Optimization, Advanced Machine Learning, Big data, and Natural Language Processing. Abualigah currently serves as an associate editor of the Journal of Cluster Computing (Springer), the Journal of Soft Computing (Springer), and Journal of King Saud University - Computer and Information Sciences (Elsevier).

tionary Computations, Information Retrieval, Text clustering, Feature Selection, Combinatorial Problems, Optimization, Advanced Machine Learning, Big data, and Natural Language Processing. Abualigah currently serves as an associate editor of the Journal of Cluster Computing (Springer), the Journal of Soft Computing (Springer), and Journal of King Saud University - Computer and Information Sciences (Elsevier).



Mohammed A. A. Al-qaness received the B.S., M.S., and Ph.D. degrees from Wuhan University of Technology, in 2010, 2014, and 2017, respectively, all in information and communication engineering. He was an Assistant Professor with the School of Computer Science, Wuhan University, Wuhan, China. He is also a Postdoctoral Fellower with the State Key Laboratory for Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University. His current research interests include optimization algorithms, wireless sensing, mobile computing, machine learning, signal and image processing, and natural language processing

Authors and Affiliations

Mohamed Abd Elaziz^{1,8,9,10} · **Ahmed A. Ewees**^{2,3} · **Dalia Yousri**⁴ · **Laith Abualigah**^{5,6} · **Mohammed A. A. Al-qaness**^{7,11} 

✉ Mohammed A. A. Al-qaness
alqaness@whu.edu.cn

Mohamed Abd Elaziz
abd_el_aziz_m@yahoo.com

Ahmed A. Ewees
eweess@du.edu.eg

Dalia Yousri
day01@fayoum.edu.eg

Laith Abualigah
Aligah.2020@gmail.com

- ¹ Department of Mathematics, Faculty of Science, Zagazig University, Zagazig, Egypt
- ² Department of e-Systems, University of Bisha, Bisha 61922, Kingdom of Saudi Arabia
- ³ Department of Computer, Damietta University, Damietta 34517, Egypt
- ⁴ Engineering Department, Faculty of Engineering, Fayoum University, Fayoum, Egypt
- ⁵ Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan
- ⁶ School of Computer Sciences, Universiti Sains Malaysia, 11800 Pulau Pinang, Malaysia
- ⁷ State Key Laboratory for Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China
- ⁸ Artificial Intelligence Research Center (AIRC), Ajman University, Ajman 346, United Arab Emirates
- ⁹ Department of Artificial Intelligence Science & Engineering, Galala University, Suze 435611, Egypt
- ¹⁰ School of Computer Science and Robotics, Tomsk Polytechnic University, Tomsk 634050, Russia
- ¹¹ Faculty of Engineering, Sana'a University, Sana'a 12544, Yemen