IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received February 13, 2021, accepted April 5, 2021, date of publication April 14, 2021, date of current version April 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3073261

An Efficient Marine Predators Algorithm for Feature Selection

DIAA SALAMA ABD ELMINAAM[1,2], AYMAN NABIL[2], SHIMAA A. IBRAHEEM[3], AND ESSAM H. HOUSSEIN[4]

[1]Faculty of Computers and Artificial Intelligence, Benha University, Banha 13518, Egypt
[2]Faculty of Computers and Information, Misr International University, Cairo 611310, Egypt
[3]Computer Science Department, Higher Technological Institute, 10th of Ramadan City 32987, Egypt
[4]Faculty of Computers and Information, Minia University, Minya 61519, Egypt

Corresponding author: Diaa Salama Abd Elminaam (diaa.salama@fci.bu.edu.eg)

ABSTRACT Feature Selection (F.S.) reduces the number of features by removing unnecessary, redundant, and noisy information while keeping a relatively decent classification accuracy. F.S. can be considered an optimization problem. As the problem is challenging and there are many local solutions, stochastic optimization algorithms may be beneficial. This paper proposes a novel approach to dimension reduction in feature selection. As a seminal attempt, this work uses binary variants of the recent Marine Predators Algorithm (MPA) to select the optimal feature subset to improve classification accuracy. MPA is a new and novel nature-inspired metaheuristic. This research proposes an algorithm that is a hybridization between MPA and k-Nearest Neighbors (k-NN) called MPA-KNN. K-Nearest Neighbors (k-NN) is used to evaluate the selected features on medical datasets with feature sizes ranging from tiny to massive. The proposed methods are evaluated on 18 well-known UCI medical dataset benchmarks and compared with eight well-regarded metaheuristic wrapper-based approaches. The core exploratory and exploitative processes are adapted in MPA to select the optimal and meaningful features for achieving the most accurate classification. The results show that the proposed MPA-KNN approach had a remarkable capability to select the optimal and significant features. It performed better than the well-established metaheuristic algorithms we tested. The algorithms we used for comparison are Grey Wolf Optimizer (GWO), MothFlame Optimization Algorithm (MFO), Sine Cosine Algorithm (SCA), Whale Optimization Algorithm (WOA), Slap Swarm Algorithm (SSA), Butterfly Optimization Algorithm (BFO), and Harris Hawks Optimization (HHO). This paper is the first work that implements MPA for Feature Selection problems. The results ensure that the proposed MPA-KNN approach has a remarkable capability to select the optimal and significant features and performed better than several metaheuristic algorithms. MPA-KNN achieves the best averages accuracy, Sensitivity, and Specificity rates of all datasets.

INDEX TERMS Feature selection, marine predators algorithm, metaheuristics, k-nearest neighbors, exploitation phase.

I. INTRODUCTION

The rapid growth in the amount of stored, processed, and retrieved data in systems today has made extracting essential and meaningful information from those systems difficult [1], [2]. This is because the collected data often contain redundant and irrelevant data that can harm any operation's chances of success.

Medical datasets are one of the most critical fields in the real world, but they can be filled with irrelevant and redundant features that lead to increased dimensionality. This negatively affects the accuracy, cost, and speed of the learning process [3]. Diagnosing and treating diseases requires high accuracy and sensitivity. Analyzing medical data efficiently and accurately helps doctors to diagnose patients quickly and reduce treatment costs, which will improve health care in our society. So, it is essential to find a way to avoid these redundant features and their impact. The way to do that is Dimensionality Reduction (DR) [2], [4].

Dimensionality Reduction (DR) is a technique that excludes non-important, redundant, and irrelevant features from datasets without reducing the amount of information

The associate editor coordinating the review of this manuscript and approving it for publication was Jiankang Zhang.

they contain [2], [5]. DR techniques can be divided into two classes. The first is Feature Extraction (FE). The FE process converts high-dimensional data to low-dimensional data by extracting new feature spaces from the original dataset features. The second class is Feature Selection (FS). FS is the process that extracts the lowest number of the most informative and meaningful features from existing features in the original dataset without losing information. It has proved to be a useful tool for dealing with redundant and irrelevant features without loss of information [2], [5], [6].

There are three categories of FS: supervised [7], unsupervised [8], [9], and semi-supervised [10]. These categories are determined by the class label's dependency degree in the dataset [5]. The general framework of feature selection is shown in **fig. 1**.

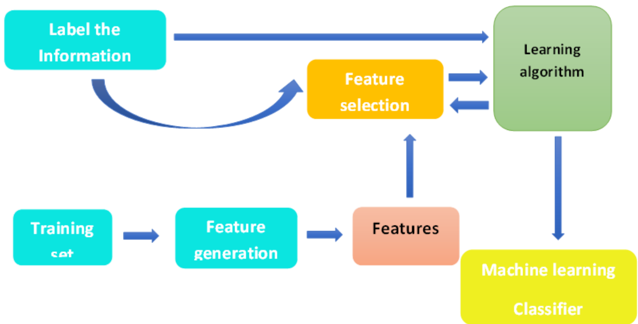


FIGURE 1. Feature selection framework.

FS techniques can be divided into three methods based on the level of the learning algorithm involved in evaluating feature subsets [5], [11], as shown in 2. The first method, the filter method, uses filters that discard learning algorithms while selecting features that do not depend on learning tasks such as classification. In the filter method, features are ranked according to the internal relationships among the data. Filters are regarded as a fast method. Some of the popular filter models include Chi-Square [12], information Gain (IG) [13], Gain Ratio [14], and ReliefF [15]. The second method involves wrappers. Wrappers use a specific learning algorithm to evaluate each feature selected (classification). In this method, wrappers determine a predictor and select the feature subset, improving classification accuracy [16], [17]. Even though wrappers can optimize classification accuracy, the optimization process is slow. The final method is the embedded method. In the embedded method FS is included in the training process in cases where coordination between learning speed and model performance is desirable [18].

Searching for the optimal feature subset is a challenging problem for FS methods. Three search strategies can be used with FS [19]:

- A complete search that tries to generate all possible feature subsets to select the best subset.
- Random searches, in which feature subsets are selected randomly with the hope of finding the best set. In the worst cases this can become a complete search.

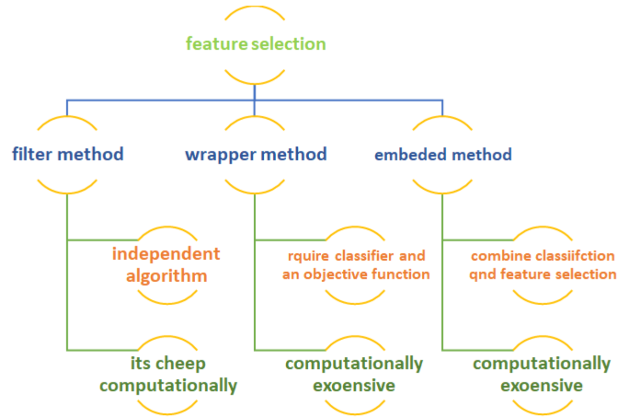


FIGURE 2. Feature selection methods and their branches.

- Heuristic techniques, which are guided by the random search process based on heuristic information. These provide more ways to trade off between local and global searches to find a suitable solution within a reasonable running time, rather than searching indefinitely for the best one.

Metaheuristic approaches have been widely used in recent years and are more efficient in dealing with the optimization process, including machine learning and FS, than other existing techniques [20].

There are many categories of metaheuristics in related work such as showed in **fig. 3**. like evolutionary algorithms (e.g., genetic algorithms (GA)) [21] and swarm intelligence (SI) techniques (e.g., particle swarm optimization (PSO)), that have been used to solve the FS problem [22], [23] physics based techniques(e.g Gravitational Search Algorithm (GSA))

SI techniques include unlimited algorithms such as Grey Wolf Optimizer (GWO) [24], Water Cycle Algorithm (WCA) [25], Whale Optimization Algorithm (WOA) [26], [27], Firefly Algorithm (FA) [28], Salp Swarm Algorithm (SSA) [29]–[31], Emperor Penguin Colony [32] squirrel search algorithm [33], [34], slime mould algorithm (SMA) [35], Butterfly Optimization Algorithm [36], [37], Moth Flame Optimization [38], [39] and Marine Predators Algorithm (MPA), which is the most recent and newest SI algorithm [35].

Marine predators algorithm (MPA) is most recent and new S.I. algorithm. MPA proposed in 2020 by Faramarzi *et al.* [35], a predator-like predatory behavior that can be inspired by sea predators like sharks, monitor lizards, sunfish, equine fishes and swordfish, etc [35].The MPA's performance is tested on twenty-nine test functions, a test suite of CEC-BC-2017, randomly generated landscape, three engineering benchmarks, and two real-world engineering design problems in the areas of ventilation and building energy performance. MPA gained the second rank and demonstrated very competitive results compared to *LSHADEcnEpSin* as the best performing method and one of the CEC 2017 competition winners.

MPA demonstrated very competitive results. It proves efficiency in optimization problems. MPA has many advantages, such as fewer parameters, simple setting, easy to implement and accurate calculation. MPA gained the second rank and demonstrated very competitive results compared to LSHADE-cnEpSin as the best performing method and one of the winners of CEC 2017 competition. The statistical results analysis revealed that MPA can be nominated as a high-performance optimizer and is a significantly superior algorithm than G.A., PSO, GSA, C.S., and SSA [37].

Moreover, this algorithm has an advantage over other algorithms in that it memorizes optimization results, referring that marine predators have the advantage of good memory in reminding their associates and the location of successful foraging. MPA algorithm requires less iteration. All of these benefits are very useful for solving the FS problem. Its simple procedure, low computational encumbrance, significant convergence speed, near-global solution, independence to the problem, and gradient-free nature [40], [41].

MPA is a new algorithm that follows a foraging strategy called Lévy as well as Brownian movements in ocean predators and optimal encounter rate policy in biological interaction between predator and prey. Many types of marine creatures, such as sharks, tuna, marlins, sunfish, and swordfish exhibit Lévy-like behavior while searching for prey [42]. Essential and authentic research studies that have collected behavior data on marine predators show that that the Lévy strategy evolved as an optimal search policy among predators in response to patchy prey distribution [41], [43], [44].

- https://github.com/afshinfaramarzi/Marine-Predators-Algorithm
- https://www.mathworks.com/matlabcentral/fileexchange/74578-marine-predatorsalgorithm-mpa
- http://www.alimirjalili.com/MPA.html

The significant contributions of this paper are as follows:

- Introducing a novel algorithm for the FS problem based on MPA that applies and obeys the rules of optimal foraging strategy and confrontation rate policy of Marine ecosystems between Predator and Prey.
- Comparing MPA's performance with established swarm intelligence algorithms such as GWO, MFO, SCA, WOA, SSA, BOA, and HHO. Furthermore, a fair comparison is realized with some literary works regarding the accuracy and the number of selected features.
- The MPA approach is evaluated on eighteen datasets, where 9 of them maintain significantly high dimensions exceeding 12,0 0 0 features, with small instances. It is noteworthy to mention that implementing the metaheuristic algorithms on FS problems with this much high dimensionality is rare in literature.
- The impact of the classifier type based on MPA is realized using k-Nearest Neighbors (k-NN).

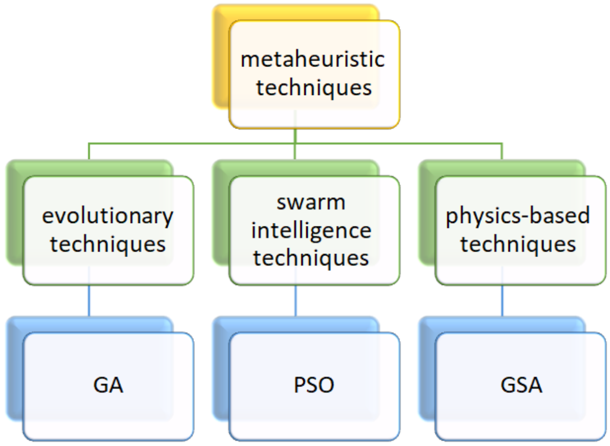


FIGURE 3. Categories of meta-heuristic techniques.

The rest of this paper is organized as follows: **Section II** demonstrates the literature review on FS metaheuristic algorithms. **Section III-A** introduces the preliminaries of the main algorithms used in this work. The proposed approach is described in detail in **Section IV**. The experiments and results are presented in **Section V**. Finally, conclusions and future work are reviewed in **Section VI**.

II. LITERATURE REVIEW

Metaheuristic has been a useful and essential tool for searching for optimal solutions (feature space) based on its global and local search competence. Many recent algorithms are used by machine learning models to deal with a considerable series of F.S. problems.

As shown in **fig.3**, the classification of metaheuristic algorithms can be classified into three classes. Evolutionary Algorithms (E.A.s), (e.g. (G.A.s). The second class is swarm intelligence algorithms (S.I.s); (e.g., PSO, GWO, WOA, HHO, and …so on), while the third class is physical algorithms [5].

Searching for the best subset of features requires an evaluation framework to compare subsets and find the best one. From an evaluation perspective, F.S. can be classified as filters and wrappers, as discussed before. Generally speaking, three criteria should be used when using a wrapper feature selection model: a classifier (k -Nearest Neighbors, Support Vector Machine, Decision Tree, etc.), feature subset selection criteria (Accuracy, false-positive elimination rate, area under the ROC), and a searching (optimization) technique to find the best combination of features [45].

Genetic algorithms (G.A.s) are the first type of evolution-based algorithm used to solve problems that are inherently complex and non-linear. Several G.A. approaches aimed at improving the F.S. In [46], [47] proposed approach using the genetic algorithm to select the best standard features from the various existing feature selection methods. This is done by combining G.A. as an evaluator to design wrapper F.S. approaches with SVM classifier. [48]–[50] use KNN as another wrapper F.S. approach. In [51] proposed an evolutionary particle swarm optimization (PISO) scheme for facial expression recognition. The system first employs a modified local binary pattern to generate an initial facial representation, Then a PSO variant embedded with the micro genetic algorithm (mGA) is proposed to accomplish feature optimization. Multiple classifiers are used in this system. In [52] proposed a two_hybrid approach that uses GA in selection in credit risk assessment.

Besides, many SI algorithms were used successfully in different FS methods like PSO, the most famous SI algorithm. So many researches interested in it and its improvement, like in [53], improve multi_objective particle swarm optimization (PSO), to search for feature subsets. While [54] combined binary PSO with C4.5 as a classifier for the fitness function for the selection of essential features. Reference [55] propose co-evolution binary particle swarm optimization with a multiple inertia weight strategy (CBPSO-MIWS) to solve the limitation of premature convergence and the setting of inertia weight of PSO.

As previously mentioned, various metaheuristic methods were adopted as the wrapper approach for FS, combined with many popular classifiers as RWN and SVM instead of k-NN. Many hybrid models combine two algorithms, such as SI and evolutionary algorithm, to solve FS problems such as in [56]. It proposed a binary genetic swarm optimization (BGSO) model that combines GA with PSO by using GA and PSO's exploitation and exploration capability.

Other successful and recent algorithms designed for both optimization and F.S. applications are WOA and GWO, HHO, and SSA. In [26] proposed, BWOA_S is used to select the optimal feature set for dimensionality reduction and classifications. In [57], proposed two binary variants of the WOA algorithm to search the optimal feature subsets for classification. Reference [24] introduced a comprehensive review with the mathematical model of feature selection based on GWO.

The latest researches throw light on HHO, SSA, and their hybrid with other approaches. In [58] Proposed two-hybrid approaches HHO-SVM and HHO-kNN, which combine Harris hawks Optimization (HHO) with Support Vector Machines (SVM) and the k_Net1arest Neighbors (k-NN) for chemical descriptor selection and compound activities. While [59] proposed an Improved Followers of Salp swarm Algorithm ISSAFD using the Sine Cosine algorithm and Disrupt operator. In [5] proposed a wrapper F.S. method called TVBSSA, which combines improved binary Salp Swarm Algorithm with Random Weight Network (RWN) classifier. In [60] Proposed hybrid model PSOGWO, which combines GWO, and PSO solves feature selection problems and finds the best feature subset. In [61] proposes a hybrid model hybrid Binary Bat Enhanced Particle Swarm (BBA) Optimization, which combines PSO with BBA to solve the F.S. problem.

Based on physical techniques, Henry gas solubility optimization (HGSO) belongs to the third class of physical algorithm, which selects meaningful features with k-NN and SVM to enhance the classification accuracy in [62].

Metaheuristic approaches to F.S. and multiple problems are still being developed, and many new approaches are being deployed in the literature. In [63] shows a review of multiple meta-heuristics techniques like GA, Harmony Search (H.S.), Artificial Bee Colony (ABC), Simulated Annealing (S.A.), Cat Swarm Optimization (CSO), Differential Evolution (D.E.), PSO, Advanced Bee Swarm Optimization, WOA, Gravitational Search Algorithm (GSA), Flower Pollination Algorithm (FPA), Shuffled Complex Evolution, and Wind-Driven Optimization. These techniques are applied to estimate the solar cell parameters to enhance the efficiency of such devices. In [64], [110] proposed an alternative method to predict FSW parameters by using an adaptive neuro-fuzzy inference system (ANFIS) integrated with harris hawks optimizer (HHO).

Here we will explore whether there are still new and exciting uses of metaheuristics in F.S. problems. No- Free-Lunch (NFL) theorem holds no algorithmic solution for solving all optimization problems [65], [115]. This statement means that most proposed metaheuristics fail when the problem is changed differently. Thus, there always is the opportunity to propose new metaheuristic-based F.S. methodologies to solve F.S. problems better. This motivated our attempts to propose a new method based on the recent Marine Predators Algorithm (MPA) and its works.

Since the MPA was proposed, there have been some preliminary studies on the MPA. For example, [66], [105], [106] proposed a forecasting model based on an improved version of the ANFIS and An improved Marine Predators Algorithm (MPA), called Chaotic MPA (CMPA), is used to enhance the ANFIS and to avoid its shortcomings. In [67], [107] introduced an improved hybrid classification approach for COVID-19 by combining CNNs to extract features and MPA to select the most relevant features. In [68][108] shows the effectiveness of S.I. algorithms for the MLT approach for image segmentation by proposing a hybrid approach MPAMFO that combines MPA with MFO. In [69], [109] combined (MPA) with Random Vector Functional Link (RVFL) network to improve the prediction accuracy of tensile elongation (T.E.) and ultimate tensile strength (UTS).

For the MPA, solving F.S. is still Non-existent work. This paper is the first work that uses MPA to solve the F.S. problem with high accuracy performance than other metaheuristic algorithms.

In **Table 1** shows all researchers with related metaheuristic approaches used in the FS domain

III. PRELIMINARIES

A. MARIEN PREDERATOR ALGORIHM (MPA)

MPA is a new algorithm that follows the foraging strategy called Lévy and Brownian movements in ocean predators and optimal encounter rate policy in biological interaction between Predator and Prey. Moreover, there are no algorithms

TABLE 1. Related approaches of feature selection.

References	Algorithms
[8]	Classify metaheuristic algorithms
[46]	GA-SVM
[47]	GA-SVM
[48]	Combine metaheuristic algorithms with K-NN
[5]	Combine metaheuristic algorithms with K-NN
[49]	Combine metaheuristic algorithms with K-NN
[50]]	Combine metaheuristic algorithms with K-NN
[51]	GA-ANN for FS approaches
[52]	hybrid approach with GA
[53]	PSO
[54]	PSO with C4.5
[55]	PSO with a multiple inertia weight strategy (CBPSO-MIWS)
[8]	binary WOA
[26]	binary variants of the WOA
[57]	GWO
[24]	HHO-SVM and HHO-kNN
[58]	Salp swarm Algorithm ISSAFD
[59]	SSA- RWN
[62]	GA-PSO
[56]	GWO – PSO
[60]	BBA-PSO
[66]	ANFIS- MPA
[67]	CNN-MPA
[68]	MFO-MPA
[69]	RVFL-MPA
[64]	ANFIS-MPA
[63]	a multiple SI algorithms
[70]	ANFIS- MPA

that can memorize the pattern of optimization results. MPA algorithm has an advantage over other algorithms in that it memorizes optimization results, referring that marine predators have the advantage of good memory in reminding their associates and the location of successful foraging. MPA algorithm requires less iteration and all of these benefits are very useful for solving the F.S. problem.

MPA inspires the foraging activity of ocean predators and the confrontation rate between predators and Prey in a marine ecosystem. In MPA, both Predator and Prey are hunting each other, and meanwhile, both are also looking for food. The predator and prey animals are sighted as thorough-going review boards. They follow a phenomenon known as the survival of the fittest, which increases predators' opportunity to find Prey. In the meta-heuristic algorithm, the next position is determined based on the current position and the next position's probability. Some strategies for predicting the behavior of marine predators are made. The Lévy strategy and the Brownian process complement each other and are best suited for describing the MPA optimization process.

Predators must select an optimal strategy to increase their encounter rates with prey in natural environments, in order to increase their chances of survival [71]. The random walk is an effective strategy that many animals in nature use in their foraging patterns. This is known as a random or stochastic process. The animal's next position can be modeled mathematically based on its current position and the probability that it will transition to the next position[60]. These optimal strategies have developed spontaneously in nature, and predators have chosen them to survive.

The search patterns class of random walk strategies is based on Lévy walks. The term Lévy walk refers to animals' movements when they are foraging. These specialized random walks have essential characteristics that combine ''walk clusters'' of short segments (distance moved per unit time) with more comprehensive relocation based on probability distribution with a power-law tail [41].

The velocity ratio of Prey to Predator is the crucial factor in transferring optimization from stage to stage. The high-velocity ratio is prominent in the first stage, while unity and low-velocity ratios are notable at the second and third stages. The MPA algorithm has several advantages, including a low number of designed variables, simple procedure, low computational encumbrance, significant convergence speed, near-global solution, independence to the problem, and the gradient-free nature.

Some studies proved that the Prey to Predator velocity ratio notably affects different strategies based on their imitation. The simulation of the effects of the velocity ratio and the Predator to Prey size ratio determines the maximum encounter rate between Predator and Prey [72]. This tradeoff between the Lévy strategy and Brownian movement creates an ideal optimization method. This is the main inspiration of the current study.

Studies illustrate examples of the variables that influence the decision-making involved in both the Brownian and Lévy strategies [44], and show whether Prey is moving using the Brownian or Lévy strategy. Lévy is the optimal strategy for the Predator. If the unit velocity ratio ($v = 1$), when Prey moves according to the Lévy strategy, the best strategy for its predator is Brownian. With high velocity, Prey is either moving in Brownian or Lévy. When Lévy is not the best strategy for Predators, they will not move because Prey will come to them on their own. This indicates that a combination of Lévy and Brownian strategies is optimal for foraging [44], [72].

The following points summarize the strategies that marine predators use to optimize searching, foraging, interactions, and memories:

- Marine predators use the Lévy strategy where there is a low concentration of prey and use the Brownian strategy in areas with numerous prey.
- During the course of their lifetimes even when traversing different habitats, marine predators show the same percentages of Lévy and Brownian movement.
- With a small velocity ratio, the best strategy for a predator is Lévy. When prey is moving in Brownian or Lévy at the unit of velocity ratio ($v = 1$), if the prey moves in Lévy, the best strategy for a predator is Brownian. If there is a high velocity ratio and the predator is not moving at all, prey's best strategy is to use Brownian or Lévy movement.
- Marine Predators with good memory memorize foraging locations as well as reminding of their associates.

Following is an explanation of the two main random walks: (i) Lévy motion and (ii) Brownian Lévy motion in the MPA mathematical model.

1) MPA MATHEMATICAL MODEL

Lévy MOTION

A Lévy process named based on French mathematician Paul Lévy is known as a stochastic process with independent, stationary increments. It determines the motion of a point successive displacements are random, in which displacements in pairwise disjoint time intervals are independent, and displacements in different time intervals of the same length have identical probability distributions. A Lévy process is known as the continuous-time analog of a random walk.

Lévy flight is one of random walk type in which the step sizes are demonstrated by a probability function defined by Lévy distribution (power-law tail) as shown in Eq.(1):

$$L\left(x_j\right) \approx \left|x_j\right|^{1-\alpha} \tag{1}$$

where the flight length denoted to x_j while $1 < \alpha \leq 2$ is the power-law exponent [42].

In [73], The integral form of the probability density of the Lévy stable process is determined by as shown in Eq. (2):

$$f_L(x; \alpha, \gamma) = \frac{1}{\pi} \int_0^{\infty} \exp\left(-\gamma q^{\alpha}\right) \cos(qx) dq \tag{2}$$

where α shows the distribution index and controls the properties of the process's scale while γ selects the unit of scale. In just a few cases, Eq.(2) has an analytical solution. When $\alpha = 2$, it shows a Gaussian distribution, and when $\alpha = 1$, it represents a Cauchy distribution [70]. in Eq.(3), the solution generally requires using the method of series expansion only when x has enormous value as follows:

$$f_L(x; \alpha, \gamma) \approx \frac{\gamma \Gamma(1+\alpha) \sin\left(\frac{\pi\alpha}{2}\right)}{\pi x^{(1+\alpha)}}, x \to \infty \tag{3}$$

Gamma function Γ in which for integer α numbers $\Gamma(1+\alpha)$ is equal to $\alpha!$

BROWNIAN MOTION

Brownian motion is the process in which the probability function of Gaussian distribution determines the step length when the mean is equal to zero ($\mu = 1$) and variance ($\sigma 2 = 1$). In point x for this motion, The Probably Density Function (PDF) is defined as in Eq.(4) [74]:$(x; \alpha, \gamma)$.

$$\begin{aligned} f_B(x; \mu, \sigma) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \\ &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \end{aligned} \tag{4}$$

There is a study proposed a new and practical algorithm that uses the Magneta method for extracting random numbers based on Lévy distribution as showed in eq.(5) where the arbitrary value of index distribution (α) ranged in 0.3 and 1.99 [73]:

$$\text{Levy}(\alpha) = 0.05 \times \frac{x}{|y|^{1/\alpha}} \tag{5}$$

where the normal distribution variables x and y while standard deviations of σx and defined as follows in Eq.(6), Eq.(7), and Eq.(8):

$$x = \text{Normal}\left(0, \sigma_x^2\right) \tag{6}$$

$$y = \text{Normal}\left(0, \sigma_y^2\right) \tag{7}$$

$$\sigma_x = \left[\frac{\Gamma(1+\alpha) \sin\left(\frac{\pi\alpha}{2}\right)}{\Gamma\left(\frac{(1+\alpha)}{2}\right) \alpha 2^{\frac{(\alpha-1)}{2}}}\right]^{1/\alpha} \quad \text{and } \sigma_y = 1 \text{ and } \alpha = 1.5 \tag{8}$$

2) MPA OPTIMIZATION PROCESS

This process is divided into three main phases of optimization based on various velocity ratios and at the same time imitating the natural life of predators and prey. These phases are defined as follows:

- ***The first phase:*** when there is a high-velocity ratio and the prey is moving faster than the predator.

As per the rules extracted and shown in fig 1 when there is a high-velocity ratio ($v \geq 10$), Predator's best strategy is not moving. In this initial iteration of optimization, exploration is important. The mathematical model of this rule is applied as in eq(9):

while Iter $< \frac{1}{3}$ *max_Iter*

$$\begin{aligned} \text{stepsize}_i &= \vec{R}_B \otimes \left(\text{Elite}_i - \vec{R}_B \overline{\otimes \text{Prey}_i}\right) \quad i = 1, \ldots n \\ \overrightarrow{\text{Prey}}_i &= \overrightarrow{\text{Prey}}_i + P \cdot \vec{R} \otimes \text{stepsize}_i \end{aligned} \tag{9}$$

Based on the Normal distribution of Brownian motion, the vector of R_B has random numbers. Multiplying R_B by Prey imitates the movement of prey.

P is a constant number equal to 0.5, and R is a uniform random numbers vector in [0,1]. This phase happens in 1/3 of iterations, when (Iter $\to$ the current iteration, Max _iter $\to$ the maximum one), the velocity of movement is high to allow high levels of exploration.

- ***The second Phase:*** when both Predator and Prey are moving simultaneously. This scenario happens when the exploration tries to be fleetingly converted to exploitation. Exploration and exploitation matters are included in this Phase, when Prey is responsible for exploitation and Predator for exploration.

Based on the rule generated; if the unit velocity ratio ($v \approx 1$), and Prey's best strategy is to move in Lévy, the best strategy for Predator is Brownian. The mathematical model of this rule is applied as shown in Eq. (10).

While 1 / 3 max_Iter $\prec$ *Iter* $\prec$ 2/3 *max_Iter*

$$\begin{aligned} \overrightarrow{\text{stepsize}_I} &= R_L \otimes \left(\overrightarrow{\text{Elite}_I} - R_L \overline{\otimes \text{prey}_I}\right) \quad i = 1, \ldots n/2 \\ \overrightarrow{\Pr ey_i} &= \overrightarrow{\Pr ey_i} + P. \vec{R} \otimes stepsize_i \end{aligned} \tag{10}$$

For the first half of the population, R_L denotes a vector of random numbers based on Lévy distribution. The movement of Prey in the Lévy manner is simulated by the multiplication of R_L Adding step size to the prey position further simulates the movement of prey. Most step sizes in the Lévy distribution are small. For the second half of the population, this study represents:

$$stepsize_i = \overrightarrow{R}_B \otimes \left(R_B \otimes \overrightarrow{E}lite_i - \overrightarrow{Prey_i} \right) \quad i = n/2, \ldots n$$
$$\overrightarrow{Prey_i} = \overrightarrow{Elite_i} + P.CF \otimes \overrightarrow{s}tepsize_i \tag{11}$$

where $CF = (1 - \frac{iter}{Max\_\{iter\}})^{2\frac{iter}{\max\_\{iter\}}}$

While CF is an adaptive parameter to control the step size for predator movement. The movement of Predator in the Brownian manner is simulated by the multiplication of R_B Furthermore, Elite simulates the movement of Prey as it updates its position depending on the movement of predators in Brownian motion.

- ***The Third Phase*** when Predator is moving faster than Prey with a low-velocity ratio. This phase helps with and is associated with high exploitation capability.

The best strategy for Predator is Lévy when there is a low-velocity ratio (v = 0.1). In this phase, the movement of Predator in Lévy strategy is simulated by the multiplication of R_L Furthermore, while the Predator's movement helps Prey update its position, Elite is simulated by adding the step size to the Elite position. This is presented as:

While Iter $\prec 2/3$ *max_Iter*

$$stepsize_i = \overrightarrow{R}_L \otimes \left(R_L \otimes \overrightarrow{E}lite_i - \overrightarrow{Prey_i} \right) \quad i = 1, \ldots n$$
$$\overrightarrow{Prey_i} = \overrightarrow{Elite_i} + P.CF \otimes \overrightarrow{s}tepsize_i \tag{12}$$

In each phase, a limited period of iteration is defined. These phases are specified based on the rules governed by the nature of predator and prey movement that are imitated in Predator and Prey's behavior. These three phases simulate the step size taken by a predator to catch prey. The rules assume that the percentage of Lévy and Brownian movement will remain constant throughout a predator's lifetime. The Predator is not moving at all in the first phase and it is moving in a Brownian pattern in the second phase. In the third phase, it uses the Lévy strategy. Because prey also has the potential to be a predator, this scenario also happens to prey. Specifically, Prey is moving in Brownian in **Phase 1**, while it follows Lévy behavior in **phase 2**.

Environmental issues can also cause changes in the behavior of marine predators. One example of this is the effects of Fish Aggregating Devices (FADs), also known as eddy formation. Sharks spend more than 80% of their time near FADs, and during the other 20%, of their time, they take long jumps in different dimensions, probably to find environments with different prey [75]. The FADs are considered local optima and their effect is trapping these points in search space. During the simulation, these longer jumps avoid stagnation in local optima. The mathematical model of FAD's effect is defined in.(13):

$$\overrightarrow{prey_l} = \begin{cases} \overrightarrow{prey_i} + CF \left[X_{\min} + \vec{R} \otimes \left(\bar{X}_{\max} - \bar{X}_{\min} \right) \right] \otimes \vec{U} \\ \quad \text{if } r \leq \text{ FADs} \\ \overrightarrow{prey_l} + [\text{FADs}(1-r)+r] \left(\overrightarrow{prey_{r1}} - \overrightarrow{prey_{r2}} \right) \\ \quad \textit{if } r > \text{FADs} \end{cases} \tag{13}$$

The uniform random number in [0, 1] is r, while the vectors that form the lower and upper boundary of the dimensions are X_max, and X_min. It is likely that FADs will affect the optimization process when FADs = 0.2. The binary vector U with arrays includes zero and one. This is defined by generating a random vector in [0, 1]. Simultaneously, the array changes to 0 if the array is less than 0.2 and the array changes to 1 if it is greater than 0.

3) MARIEN PREDERATOR ALGORIHM (MPA) MEMORY

Marine predators have excellent memory for recalling the places where they have been successful in foraging. Memory saving in MPA simulates marine predator's memory. Once it updates and implements the Prey and FADs effects, respectively, this matrix must be evaluated for its fitness to update the Elite. The fitness of the current iteration for each solution is compared to its equivalent in the preceding iteration, and the current one replaces the earlier solution if it is more appropriate. This process enhances the quality of the solution with each iteration [2], [66], [68], [76] and simulates predators returning to prey-abundant areas with efficient foraging.

4) MPA PHASES, EXPLORATION AND EXPLOITATION

Based on the optimization phases presented before, in the first phase, the Prey moves in Brownian motion. In initial iterations, the prey is distributed evenly throughout the search space. Predator and Prey's distance is relatively large, so useful exploration of the domain is reached based on Brownian motion. This allows prey to explore their neighborhood separately. After that, each Prey with a new position is evaluated for fitness if the position is better than the one it replaced. The Prey's positions can be considered significant food areas, and the saving process is equivalent to prey's ability to remember significant food areas. Once the Prey is more successful in foraging for its food, it can be considered a Predator. This means that the Prey's fitness value is calculated and replaces the top Predator if its fitness is better. During this time, the Prey is still looking for food while Predators start foraging.

Once the Predator starts foraging, the optimization's second phase starts. During this phase the optimization transitions from exploration to exploitation. To achieve success in both exploration and exploitation in this phase, the Predator and Prey must both search for their food. Half of the population in this phase is in charge of exploration, and the other half is in charge of exploitation. Predators use the Brownian strategy to search for Prey. At the same time, Prey starts to search in its close neighborhood. If it cannot find any food nearby, it then takes a long jump by switching to the Lévy strategy. While the predator and prey locations become closer to each other and the distance length is smaller than it was in the previous phase, FADs' effect combined with the long steps in the Lévy strategy helps the MPA achieve better performance and avoid local optima stagnation.

The MPA algorithm requires high exploitation ability in the final phase of the optimization process. In this phase, the Predator starts switching its behavior from the Brownian to the Lévy strategy to search a particular neighborhood efficiently. Predators help limit the search areas for exploitation within a particular neighborhood based on the adaptive defined convergence factor (CF). CF also avoids wasting search effort by using the Lévy strategy's long step sizes for the domain's non-promising regions. The pseudo-code of MPA is shown in Algorithm 1

Algorithm 1 Pseudo-Code of MPA Algorithm [35]

Initialize a population of search agents (Prey) i=1,2,...,n
while stop condition not met **do**
  Compute the initial fitness function and construct the matrix of Elite
  if $Iter < Max\_Iter/3$ **then**
    Update the prey with the aid of Eq. (9)
    if $Max\_Iter/3 < Iter < 2 \star Max\_Iter/3$ **then**
      For the first half of population (i =1,...,n/2)
      Update the prey via Eq. (10)
      For the second half of population (i =n/2,...,n)
      Update the prey via Eq. (11)
      if $Iter > 2 \star Max\_Iter/3$ **then**
        Update the prey via Eq. (12)
      end if
    end if
  end if
  Complete the memory saving and Elite update
  Implementing FADs effect and update using Eq. (13)
end while

B. k-NEAREST NEIGHBOR (k-NN)

K-Nearest Neighbor (k-NN) is used for classifying a record based on the majority vote of the nearest k neighbors in the training dataset [75]. The Euclidean distances between the unclassified record and the classified records are calculated and sorted. A single observation from the original dataset is selected as the validation data and the remaining observations are selected as training data. That means that each observation is used once as the validation data. This method is called leave-one-out cross-validation (LOOCV) of one nearest neighbor (1-NN), and is depicted in Algorithm 2.

Algorithm 2 K-Nearest Neighbor Procedure. [45], [48]

for $i \in \{1, \ldots, numberofsamples\}$ **do**
  for $j \in \{1, \ldots, numberofsamples\}$ **do**
    for $k \in \{1, \ldots, numberoffeaturesinthedataset\}$ **do**
      $dist_i = dist_i + \left(data_{ik} - data_{jk}\right)^2$
    end for
    if $dist_i < nearest$ **then**
      $class_i = class_j$
      $nearest = dist_i$
    end if
  end for
end for
for $i \in \{1, \ldots, numberofsamples\}$ **do**
  if $class = realclassoftestingdata$ **then**
    $correct = correct + 1$
  end if
end for
$Fitnessvalue = \frac{correct}{numberoftestingdata}$

IV. THE PROPOSED MPA BASED CLASSIFIER ALGORITHM

A. MPA-KNN PROPOSED APPROACH

The proposed algorithm is a hybridization of the MPA algorithm with KNN for classification, feature selection, and parameter optimization. In MPA, the best location in a marine predator's memory is identified as KNN parameters and the selected features are set for all cross-validation folds. The flowchart of the proposed KNN-MPA approach is shown in **Fig. 4**, which defines the proposed algorithm's three phases. The first is the preprocessing phase; the second is the feature selection and optimization phase; and the last is the classification and cross-validation phase. The pseudocode of the proposed MPA-KNN classifier algorithm is shown below in Algorithm **3**.

B. FITNESS FUNCTION

A critical aspect of F.S. methods is the assessment of the quality of the selected samples. Because this technique utilizes a wrapper, a learning algorithm should be involved in the evaluation process. In this work, the well-known classifier, i.e., a k Nearest Neighbors (k-NN) classifier, is used as a classifier, and the proposed fitness function controls the Accuracy of the selected features. When the features selected in a particular subset are relevant, the obtained Accuracy will be better. Accurate classifying items is the goal of F.S. methods.

The solutions MPA obtains must be evaluated during the iterative process to verify each iteration's performance. The fitness function used by the MPA is defined as:

$$fobj = \alpha + \beta \frac{|\mathrm{R}|}{|\mathrm{C}|} - \mathrm{G} \tag{14}$$

$$\beta = \alpha \tag{15}$$

$$fobT \tag{16}$$

The classification error rate is defined as R, where C refers to the total number of features in the data set. The two parameters, α, and β, refer to the importance of classification quality and subset length. In contrast, α is defined in the range [0, 1] where G refers to the group column for the classifier, and T is considered the condition in which every algorithm is compared with the fitness function. The objective function must be greater than T to maximize the solution.

C. FEATURE SELECTION

The feature selection process is commonly used before machine learning algorithms. It is a reprocessing step that is used to find a subset of features that are clear and without redundancy or duplication. The process enhances the

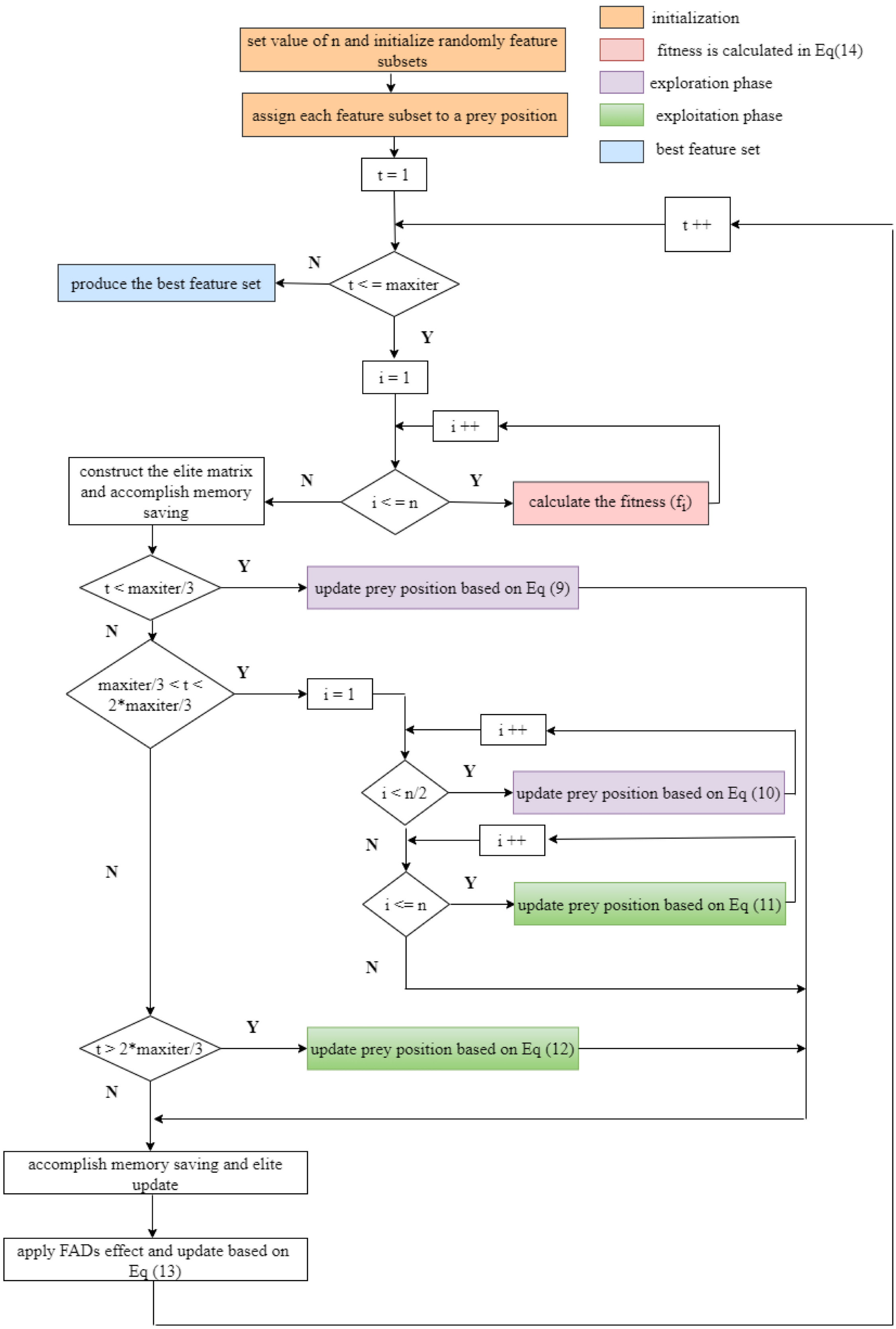


FIGURE 4. Flow chart of MPA-KNN classifier.

Algorithm 3 Pseudo-Code of the Proposed MPA-kNN Classifier Algorithm

KNN

Inputs: The population size N, the maximum number of iterations T, Dataset set D, group classifier G, feature X, and new fitness function MPA- KNN (fobj).
Outputs: The accuracy value for each iteration (best location) and the best accuracy value.
Initialize the random population X i (i = 1, 2,..., N)
while *stoppingconditionisnotmetn* **do**
  New fitness function calculated based on call feature selection method, Call K-NN classifier
  Update prey based on Eq. (9)
  for the first half of the populations (i = 1,...,n/2) **do**
    Update prey based on Eq. (10)
    for the other half of the populations (i = n /2,...,n) **do**
      New fitness function calculated based on call feature selection method
      Call K-NN classifier
      Update prey based on Eq. (11)
    end for
  end for
  Update prey based on Eq. (12)
  Accomplish memory saving, and Elite update Applying FADs effect and update based on Eq. (13)
end while

accuracy of prediction and the understanding of information and data for machine learning algorithms by selecting features that can be better correlated. This means that when two features are perfectly correlated, only one of them is used because it sufficiently describes the data. For the feature vector sized N, the variant combinations of features must be 2 N which is considered a huge space that would be difficult to be explore exhaustively. Evaluating it directly becomes an NP-hard problem because of the increment in the number of features [58], [77]. A vast search space is considered a big challenge for the feature selection process but metaheuristic algorithms can solve feature selection problems perfectly. MPA is adapted to select the significant and best features, as shown in **Fig. 4**, while it searches in the feature space to select the best feature subset. The best subset of features (best solution) must maximize the classification accuracy and minimize the classification error rate. In addition, it must have a minimum number of selected features.

The methods employed for feature selection are introduced in this section.

V. EXPERIMENT RESULTS AND DISCUSSION

MPA- KNN is used to select the optimal subset of features from original ones that reduce problem domain. this approach helps medical fields, especially in diagnosing or prognosis diseases besides, it allows the government to forecast the number of infected people in the country, besides a lot of medical and treatment area.

The MPA approach has many advantages in that it can memorize patterns of optimization results based on their marine predators can memorize their associates and the locations of their successful foraging. Additionally, MPAs requires far fewer iterations. It has a wide range of advantages such as simple procedure, reduced computational demand, high convergence speed, independent to the problem, and gradient-free nature. All of these crucial benefits are of great use for solving Fire Safety Problems.

Here we present the results of the proposed MPA-KNN Hybrid approaches and spotlight the effectiveness of these proposed improvements, demonstrating the MPA algorithm's performance in the F.S. process. In this work, we use 10fold cross-validation for k-NN to eliminate possible bias in selecting the test and training sets. We compared a proposed adaptive algorithm with others, GWO, MFO, SCA, WOA, SSA, BOA, and HHO. All of the algorithms were tested on Eighteen medical datasets (UCI repository [78]) and evaluated on their predictive Accuracy. All of the algorithms are hybrid with standard machine learning classifiers in the literature, which is KNN. Matlab is used for the implementation of this study. Each compared algorithm has been run 30 times, with 30 agents and 1000 iterations.

A. DATA DESCRIPTION

In this study, datasets were used to compare and investigate the effectiveness of the FS algorithm. These test cases were utilized in multiple works and research studies, which cover many characteristics. These data sets were obtained from the well-regarded UCI repository (UCI repository [78]). **Table 2** represents the essential characteristics of 18 well-regarded datasets used in this study to compare approaches' efficacy.

TABLE 2. Dataset including the number of features and instances.

No	Data Set	No of Features	No Instancces
1	AA	30	569
2	base_leuk1	11225	72
3	BeastEW	30	596
4	CongressEW	16	435
5	Exaclty	13	1000
6	IonosphereEW	34	351
7	KrvskpEW	36	3196
8	Lymphography	18	148
9	M-of_n	13	1000
10	ND	1665	68
11	penglungEW	325	73
12	Prost	10509	102
13	SonarEW	60	208
14	SpecEW	22	267
15	Vote	16	300
16	WaveformEW	40	5000
17	WineEW	13	178
18	Zoo	16	101

B. PERFORMANCE EVALUATION MEASURES

The proposed approach was evaluated and validated based on a set of measures accuracy sensitivity, Specificity, and the number of selected features. At the same time, the best fitness value fobj defined at run i these measures defined as follow:

- **The average (mean)** of the fitness function values obtained by running algorithm M for times. The Mean fitness function can be calculated by see Eq.(17):

$$\text{Mean } = \frac{\sum_{i=1}^{M} \text{ fobj}}{M} \tag{17}$$

- **Standard Deviation** (*StdDev*) It is used to vary the fitness function value computed from the running algorithm M times. It shows an indicator of the stability of an algorithm. *StdDev* can be calculated by see Eq.(18):

$$\text{Std} = \sqrt{\frac{1}{M-1}\sum_{i=1}^{M}(\text{ fobj } - \text{ mean })^2} \tag{18}$$

- **Accuracy** It is a measure that is defined as a total, correctly identified examples out of all the examples. Accuracy is determined as: False-Negative see Eq.(19):

$$\text{accuracy} = \frac{tp+tn}{tp+fp+fn+tn} \tag{19}$$

  While TP refers to True Positive, which means correctly identified, (FP) refers to incorrectly identified or False Positive, TN is defined as True Negative or correctly rejected. FN refers to False-Negative, which means incorrectly rejected.
- **Average Accuracy** (AVG_{ACC}) It is a measure that is defined as a average, correctly identified examples out of all the examples. Average Accuracy is calculated by: Eq.(20):

$$\text{avgAccurcy} = \frac{\Sigma_{i=1}^{M} \text{ Accurcy}}{M} \tag{20}$$

  while M is The number of run times of the optimization algorithm to select a subset of feature.
- **sensitivity** Sensitivity and Specificity are useful in medical applications and all other applications with images and visual examples. a test can correctly identify those with the disease (True positive rate).see Eq.(21):

$$\text{sensitivity } = \frac{tp}{tp+fn} \tag{21}$$

- **Specificity or True Negative Rate (TNR)** It defines the proportion of actual negatives that are correctly identified as shown in Eq.(22):

$$\text{Specificity } = \frac{tn}{tn+fp} \tag{22}$$

  It refers to the ability of the test to identify those without the disease (True correctly negative rate)
  Sensitivity and Specificity approximate the probability of the positive and negative class, which is true; this means it estimates the algorithm's efficiency and productivity based on a single class.
- Number of selected features as a result of the feature selection process of all algorithms

TABLE 3. Parameters setting of the tested algorithms used in the evaluation and comparison.

Algorithm	parameter	value
GWO	R	Rand
MFO	B and Population Size(Flam _no)	1 and 50
SCA	A	2
WOA	FE and NP	[-1:-2]
SSA	C rand	Rand
BOA	PS	0.8
HHO	Beta	1.5
MPA	FADs = 0.2	P=0.5

C. RESULTS ANALYSIS

The proposed algorithm was implemented in Matlab, and the integrated development environment is Matlab R2016a., which was run on an Intel(R) Core i7 2.80 GHz CPU with 8 GB RAM and the Windows 10 operating system. The proposed adaptive algorithm (i.e., the MPA-KNN) was evaluated against other popular algorithms, such as Grey Wolf Optimizer (GWO), Moth-Flame Optimization (MFO) algorithm, Sine Cosine Algorithm (SCA), Whale Optimization Algorithm (WOA), Slap Swarm Algorithm (SSA), Butterfly Optimization Algorithm (BOA), and Harris Hawks Optimization (HHO). All of the algorithms were tested on a set of 18 publicly available benchmark datasets.

The number of elements in the population used for these algorithms is equivalent to randomly distributed at the beginning in a multidimensional bounded search space. All of these algorithms are evaluated based on the fitness function, which was discussed earlier00.

All tested algorithms are tested based on a set of parameters defined as follows: number of iterations is 1000; number of search agents is 30; number of experiments (runs) is 30; the lower bound is 0 and the upper bound is 1; α in the fitness function is 0.99; and β in the fitness function is 0.01. The set of parameters used in all the algorithms is previewed in Table **Table 3**. All the algorithms are tested using the k-NN classifier. The experiments were performed based on 1000 iterations and 30 independent runs.

The proposed algorithms are quantitatively compared using the following metrics:

- The average classification accuracy and the standard deviation are calculated for the 30 runs.
- The average and the standard deviation of the Sensitivity of the features subset.
- The average and the standard deviation of the Specificity of the features subset.
- The number of selected features
- Convergence curves of the best-so-far solutions.

TABLE 4. Comparison between the proposed approaches based on average classification accuracy (*AvgAcc*).

benchmark	GWO		MFO		SCA		WOA		SSA		BOA		HHO		MPA	
	AvgAcc	*StdDev*	*AvgAcc*	*StdDev*	*AvgAcc*	*StdDev*	*AvgAcc*	*StdDev*	*AvgAcc*	*StdDev*	*AvgAcc*	*StdDev*	*AvgAcc*	*StdDev*	*AvgAcc*	*StdDev*
AA.	1	0	0.447	0.023335	0.964	0.004243	0.956	0.002828	0.892	0.005657	0.912	0.012728	0.947	0.005657	1	0
base_leuk1	1	0	0.7333	0.011809	1	0	1	0	1	0	1	0	0.933	0.004950	1	0
BreastEW	0.97	0.014100	0.605	0.003536	0.982	0.000707	0.973	0.004950	0.982	0.002121	0.903	0.012021	0.929	0.007778	**0.982**	**0.002263**
CongressEW	0.97	0.007000	0.574	0.138593	0.977	0.002121	0.931	0.006364	0.988	0.000707	0.954	0.011314	0.977	0.009192	0.977	**0.000707**
Exactly	0.76	0.028000	0.385	0.010607	0.75	0.014142	1	0	1	0	0.56	0.049497	0.635	0.010607	1	0
IonosphereEW	0.951	0.027000	0.774	0.000707	0.971	0.006364	0.957	0.002121	0.985	0.001414	0.901	0.013435	0.929	0.007778	**0.985**	**0.000707**
KrvskpEW.	0.982	0.005657	0.401	0.006364	0.973	0.004950	0.979	0.000707	0.987	0.002121	0.79	0.014142	0.823	0.007071	**0.993**	**0.001414**
Lymphography	0.933	0.004950	0.3	0.007071	0.933	0.007778	0.857	0.009192	0.966	0.002828	0.7	0.010607	0.655	0.014142	**0.966**	0
M-of-n	1	0	0.305	0.003536	0.86	0.028284	1	0	1	0	0.785	0.004950	0.765	0.010607	1	0
ND	1	0	0.785	0.001414	1	0	1	0	0.785	0.010607	0.714	0.011314	0.928	0.008485	1	0
PenglungEW	1	0	0.666	0.024042	1	0	1	0	1	0	0.923	0.004950	1	0	1	0
Prost	1	0	0.476	0.002828	1	0	0.952	0.005657	0.904	0.011314	0.666	0.024042	0.809	0.021920	1	0
SonarEW	0.97	0.007071	0.547	0.003536	0.976	0.009899	0.976	0.002828	1	0	0.881	0.013435	0.833	0.047376	0.976	0.002828
SpectEW	0.88	0.014142	0.5	0.070711	0.888	0.029698	0.74	0.010607	0.907	0.001485	0.759	0.007778	0.833	0.075660	0.87	**0.000707**
Vote	0.98	0.007071	0.416	0.002828	0.983	0.008485	1	0	0.983	0.004950	0.966	0.009192	0.9	0.007071	0.983	0.003323
WaveformEW	0.771	0.020506	0.349	0.021920	0.74	0.021213	0.764	0.004243	0.769	0.002828	0.765	0.010607	0.703	0.004950	**0.785**	**0.001344**
WineEW	1	0	0.785	0.003536	1	0	1	0	1	0	0.888	0.008485	0.944	0.004243	1	0
Zoo	0.943	0.012021	0.238	0.043841	1	0	1	0	1	0	0.95	0.007071	1	0	1	0

TABLE 5. Comparison between the proposed approaches based on average classification sensitivity (*Avgsens*).

benchmark	GWO		MFO		SCA		WOA		SSA		BOA		HHO		MPA	
	AvgAcc	*StdDev*	*AvgAcc*	*StdDev*	*AvgAcc*	*StdDev*	*AvgAcc*	*StdDev*	*AvgAcc*	*StdDev*	*AvgAcc*	*StdDev*	*AvgAcc*	*StdDev*	*AvgAcc*	*StdDev*
AA.	1	0	0.328	0.008485	1	0	0.953	0.000707	1	0	0.913	0.008485	0.949	0.004950	1	0
base_leuk1	1	0	0.625	0.010607	1	0	1	0	1	0	1	0	1	0	1	0
BreastEW	0.98	0.010607	0.728	0.009899	1	0	0.97	0.002121	0.985	0.001414	0.91	0.007778	0.945	0.004950	1	0
CongressEW	0.96	0.009192	0.833	0.012021	0.98	0.006364	0.86	0.002121	0.981	0.001414	0.964	0.004243	0.98	0.004950	**0.964**	**0.001414**
Exactly	1	0	0.198	0.008485	0.838	0.001414	1	0	1	0	0.671	0.006364	0.794	0.004243	1	0
IonosphereEW	1	0	0.977	0.002121	1	0	1	0	0.977	0.002121	1	0	1	0	0.979	0.004243
KrvskpEW	0.97	0.011314	0.22	0.056569	0.976	0.002828	0.984	0.001414	0.99	0.001768	0.855	0.003536	0.788	0.008485	**0.991**	0
Lymphography	0.93	0.007071	0.312	0.012728	1	0	0.15	0.035355	0.2	0.027577	0.1	0.031820	0.523	0.012021	1	0
M-of-n	1	0	0.619	0.003536	0.901	0.013435	1	0	1	0	0.634	0.004950	0.65	0.012021	1	0
ND	1	0	0.888	0.002828	1	0	1	0	1	0	0.666	0.003536	1	0	1	0
PenglungEW	1	0	0.5	0.036770	1	0	1	0	1	0	1	0	1	0	1	0
Prost	1	0	1	0	1	0	0.909	0.007778	0.818	0.002121	0.666	0.002828	0.714	0.011314	1	0
SonarEW	1	0	0.454	0.002828	0.958	0.003182	1	0	1	0	0.789	0.002121	0.761	0.013435	0.95	0.002475
SpectEW	0.14	0.042426	0.21	0.063640	0.833	0.003041	0.294	0.032527	0.7	0.014142	0.666	0.024042	0.687	0.009192	0.222	0.002121
Vote	0.96	0.012021	0.782	0.005657	1	0	1	0	0.961	0.001556	1	0	0.84	0.012021	**0.963**	**0.001414**
WaveformEW	0.74	0.014142	0.383	0.012021	0.701	0.034648	0.724	0.018385	0.742	0.005657	0.73	0.007071	0.705	0.012021	**0.771**	**0.000707**
WineEW	1	0	0.961	0.006364	1	0	1	0	1	0	1	0	1	0	1	0
Zoo	0.775	0.004950	0.333	0.012021	1	0	1	0	0.666	0.000707	0.777	0.009192	1	0	1	0

Table 4 shows the comparison between all approaches in performance based on average classification accuracy and standard deviation.

The results in **Table 4** show that MPA hybrid approaches easily performed better than other approaches in almost all the datasets. At the same time, this approach has the best accuracy rate in 77.7% of the dataset. It noticed that MPA has the best value in average classification accuracy and the best value in *StdDev*. MPA ranked as the first algorithm in performance MPA in average Accuracy and *StdDev*.SSA is ranked the second algorithm in performance after MPA in average Accuracy and *StdDev*. It provides the second-best Accuracy in 12 out of the 18 datasets used. The MPA and SSA algorithms share the best performance results in nine of the datasets. WOA and GWO have the third-best avg Accuracy with eight datasets, while GWO has the fourth-best value of *StdDev*. SCA followed with seven datasets, SO it has the fourth-best in avg Accuracy and the third-best value of *StdDev* with WOA. The MPA algorithm achieved the best result of all the datasets. HHO and BOA give the best solution in too few datasets (Penglung, Zoo) and one dataset (base leuk1), respectively after that come MFO algorithms, which ranked last and performed the worst overall.

The average of Sensitivity and Specificity are shown in **Table 5**, and **Table 6** respectively, with their *StdDev* values. The higher an algorithm's sensitivity and average specificity values, the better its performance. **Table 5**, shows that MPA has a high average sensitivity value (Avg_{Sens}) in about 66.6% of the datasets, while it has the best value of Std_{Dev} in about 77.7%. **Table 6**,shows that MPA provides a higher specificity rate (AvgSpec) than other algorithms in about 83.3% of the datasets.

The proposed MPA is highly performed than all competing algorithms. For instance, in (CongressEW) despite SSA has the best Avg_{Sens}, it is easy to notice that MPA has the best result in Std_{Dev} as shown in **Table 5**. (WaveformEW). In **Table 6**, we noticed that MPA has the best result in *StdDev* despite WOA has a higher value of Avg_{Spec}.

The number of selected features that each approach achieved through its evaluation is presented in **Table 7** MPA's number of selected features show that it is very efficient, and it is a suitable choice for the FS process, because it gives a minimum number of meaningful selected features for all datasets. In a dataset with a large number of features (base leuk1, ND, PenglungEW, Prost) the results showed that MPA reduces the number of features selected and provides a more significant and logical result than other competing algorithms. Using a dataset with a small number of features ensures that MPA will perform well in selecting the minimum number of significant features, such as sonarEW, and vote, WinEW; or more specifically IonosphereEW, M-of-N, SonarEW, and Zoo, for example.

TABLE 6. Comparison between the proposed approaches based on average classification specificity (*AvgSpec*).

benchmark	GWO		MFO		SCA		WOA		SSA		BOA		HHO		MPA	
	AvgAcc	*StdDev*	*AvgAcc*	*StdDev*	*AvgAcc*	*StdDev*	*AvgAcc*	*StdDev*	*AvgAcc*	*StdDev*	*AvgAcc*	*StdDev*	*AvgAcc*	*StdDev*	*AvgAcc*	*StdDev*
AA.	1	0	0.636	0.009899	0.911	0.006364	0.96	0.003536	0.957	0.005657	0.911	0.006364	0.949	0.007778	1	0
base_leuk1	1	0	1	0	1	0	1	0	1	0	1	0	0.875	0.004950	1	0
BreastEW	0.95	0.006364	0.409	0.007778	0.947	0.002121	0.978	0.001414	0.978	0.004243	0.893	0.004950	0.9 0.011314	**0.951**	**0.0003536**	
CongressEW	1	0	0.151	0.013435	0.972	0.005657	1	0	1	0	0.935	0.003536	0.971	0.006364	1	0
Exactly	0.18	0.019799	0.702	0.033941	0.585	0.010607	1	0	1	0	0.361	0.027577	0.296	0.006364	1	0
IonosphereEW	0.85	0.014142	0.444	0.039598	0.928	0.003536	0.88	0.006364	1	0	0.758	0.008485	0.821	0.010607	1	0
KrvskpEW.	0.98	0.007778	0.6	0.014142	0.963	0.007071	0.975	0.010607	0.894	0.004243	0.726	0.009899	0.863	0.004950	0.996	0.000707
Lymphography	0.93	0.005657	0.285	0.017678	0.833	0.012021	1	0	1	0	1	0	0.888	0.002828	0.923	0.002475
M-of-n	1	0	0.131	0.014142	0.825	0.010607	1	0	1	0	0.854	0.005657	0.817	0.008485	1	0
ND	1	0	0.6	0.012021	1	0	1	0	0.4 0.021213	0.8	0.009899	0.8	0.009899	1	0	
PenglungEW	1	0	1	0	0.5 0.009899	0.846	0.007071	1	0	0.857	0.006505	1	0	1	0	
Prost	1	0	0.421	0.013435	1	0	1	0	1	0	0.666	0.013435	0.857	0.007071	1	0
SonarEW	0.95	0.014849	0.65	0.011314	1	0	0.96	0.003536	1	0	0.956	0.005233	0.904	0.009192	1	0
SpectEW	1	0	0.6	0.010607	0.895	0.004950	0.945	0.007778	0.954	0.003536	0.785	0.009899	0.894	0.004950	1	0
Vote	1	0	0.189	0.007778	0.973	0.001626	1	0	1	0	0.944	0.006435	0.942	0.005657	1	0
WaveformEW	0.87	0.013435	0.634	0.004243	0.848	0.002828	0.889	0.009899	0.875	0.017678	0.851	0.013435	0.831	0.013435	**0.873**	**0.002121**
WineEW	1	0	0.85	0.009192	1	0	1	0	1	0	0.88	0.003536	0.909	0.001414	1	0
Zoo	1	0	0.833	0.006364	1	0	0.87	0.0042426	0.666	0.024042	1	0	0.9	0.006364	1	0

TABLE 7. Comparison between the proposed algorithm based on the number of selected features.

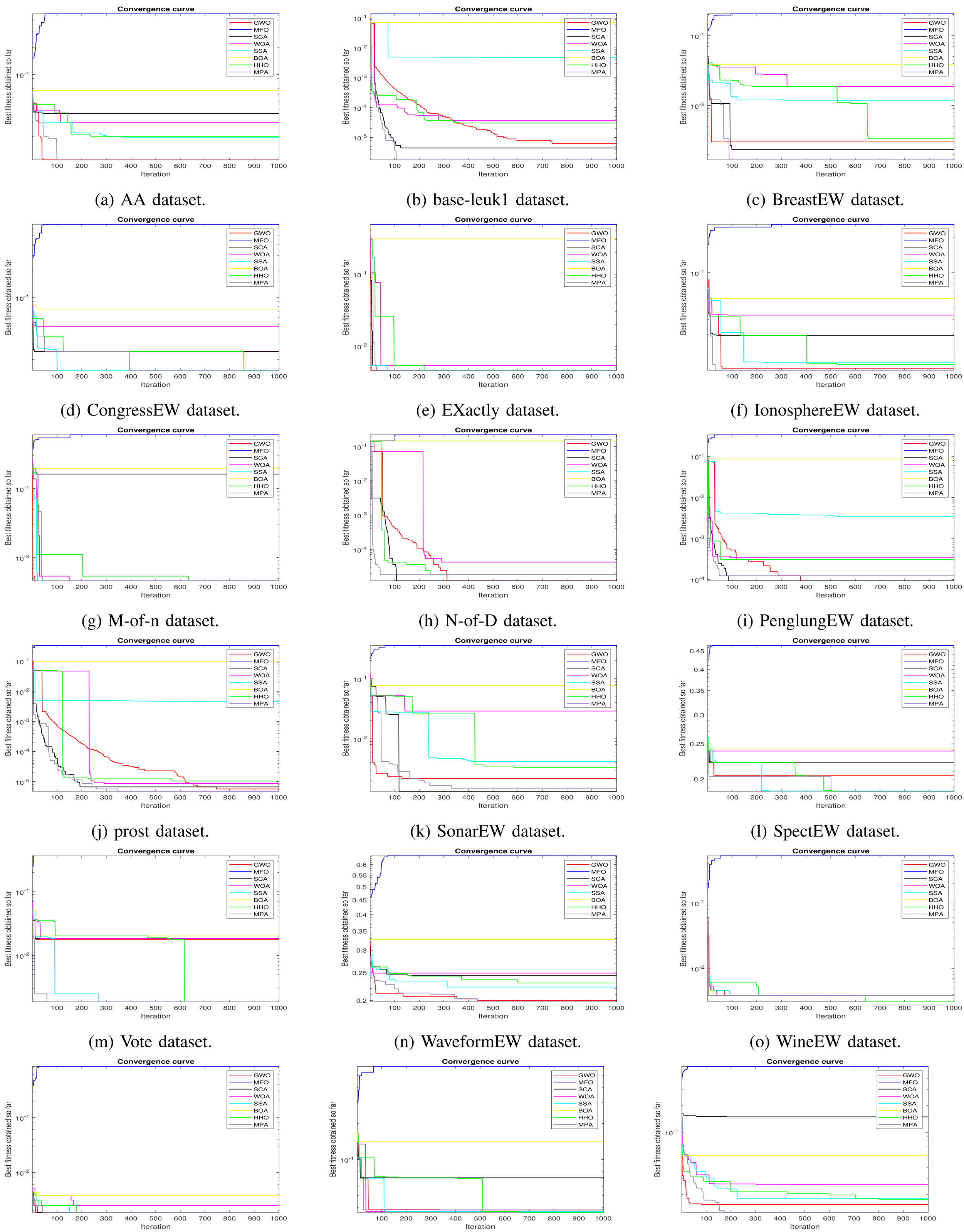
No	DataSet	GWO	MFO	SCA	WOA	SSA	BOA	HHO	MPA
1	AA	7	6	5	16	10	12	15	8
2	base_leuk1	13	6929	4	3	5317	5088	4551	2
3	BreastEW	7	6	8	8	11	12	12	12
4	CongressEW	3	5	1	1	4	2	7	3
5	Exactly	1	3	6	7	6	4	8	6
6	IonosphereEW	4	23	12	7	14	20	10	6
7	KrvskpEW.	11	7	12	23	21	17	15	21
8	Lymphography	1	9	5	1	8	7	5	7
9	M-of-n	6	2	7	6	6	7	7	6
10	ND	1	1180	4	2	708	643	747	1
11	PenglungEW	7	231	5	38	115	120	137	7
12	Prost	9	6852	3	16	4938	4059	4146	5
13	SonarEW	11	31	7	26	16	26	20	8
14	SpectEW	4	9	4	8	5	7	9	5
15	Vote	5	2	3	5	5	5	6	3
16	WaveformEW	14	15	11	25	19	2	25	19
17	WineEW	4	5	2	6	4	6	3	2
18	Zoo	3	4	4	1	3	4	5	1

Reviewing **Table 4**, **Table 5**, **Table 6**, and **Table 7**, we see that the OF MPA and GWO algorithms (in the AA dataset) had the best performance in accuracy, sensitivity and specificity equal to 100%,and MPA selected 8 features. Although SCA selected a minimum number of features (number of selected features = 5) it has an accuracy rate of 96.4%, This means that MPA had higher performance in number of selected features than SCA. Take also, for example, the result obtained by BOA not logical on the dataset WaveformEW shown in **Table 7** In this case, BOA selected only two features from the original huge feature set (number of features selected by BOA = 2). In contrast, MPA selected about 19 features from this dataset. This proves the MPA's effectiveness in solving the FS problem.

The average convergence curves for the hybrid and evaluation algorithms are presented in the figures below for all data sets. **Fig. 5** shows that the average convergence curves illustrate that the MPA and GWO algorithms perform best and reach the best solutions in cases with fewer than 100 iterations(iteration number <100).

MPA and GWO also have a high accuracy rate **Table 4** aand have a small number—ranging from 7 to 8—of selected features. **Fig. 5** ensure that MPA is performing well with a minimum number of iteration(less than or equal 100 iterations) while they have the best result in **Table 6** which equal to 100%. GWO, SCA, and MPA have the best accuracy result in **Table 4** (accuracy rate = 1). In contrast, MPA and SCA have good performance in **Fig. 5** with less than 100 iterations than GWO.in **Table 4**, MPA and other approaches have the best and high accuracy value. However, only MPA reach the best solution with minimum iteration (iterations number = 100), as shown in **Fig. 5**.

Fig. 5 shows that MPA is better than the other algorithms because it obtained the best solution with at most 100 iterations (iteration number < 100). In the (exactly dataset) shown in **Fig. 5**, MPA has high performance with fewer than 50 iterations. Despite WOA, SSA, and MPA, it provides the best accuracy result as shown in **Table 4**. However, WOA and SSA reach the best solution when the number of iterations is greater than 50 (iterations number > 50). In (ionosphere EW and lymphography datasets), MPA and SSA also provide the best accuracy rates, equal to 0.985 and 0.966, respectively (see **Table 4**). As shown in **Fig. 5**, the results prove that MPA is effective in average convergence curves with a minimum number of iterations. MPA provides a small number of selected features ranging from 6 to 7, as shown in **Table 7**. By reviewing (waveform, winEW and zoo datasets) as shown in Table 4, we notice that the best



(a) AA dataset.

(b) base-leuk1 dataset.

(c) BreastEW dataset.

(d) CongressEW dataset.

(e) EXactly dataset.

(f) IonosphereEW dataset.

(g) M-of-n dataset.

(h) N-of-D dataset.

(i) PenglungEW dataset.

(j) prost dataset.

(k) SonarEW dataset.

(l) SpectEW dataset.

(m) Vote dataset.

(n) WaveformEW dataset.

(o) WineEW dataset.

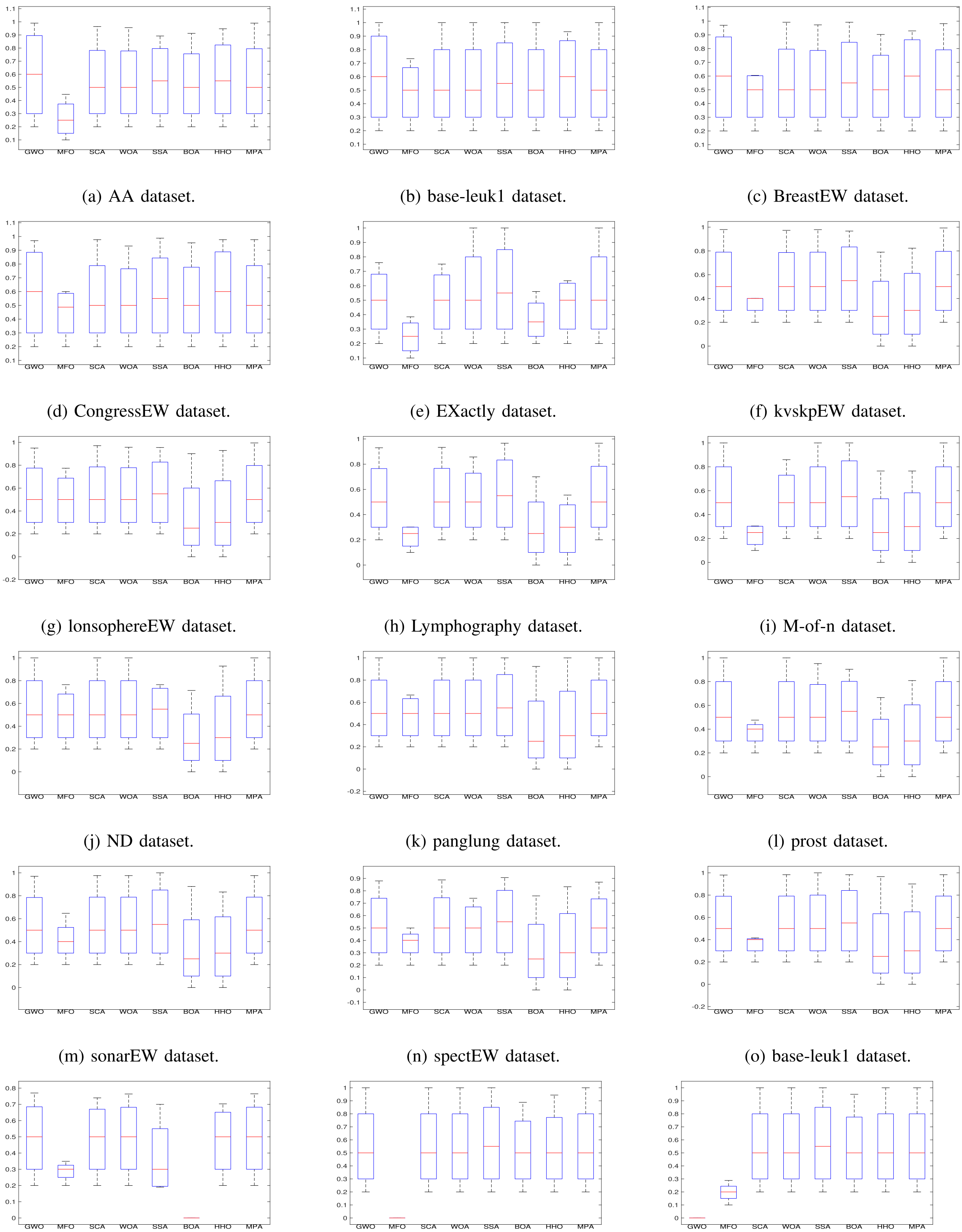
(p) Zoo dataset.

(q) Lymphography dataset.

(r) krvskpEW dataset.

FIGURE 5. Convergence curves using the MPA-kNN and 1000 iterations used as stop criteria over all the eighteen datasets.

accuracy results obtained by MPA and multiple algorithms are 0.785, 1, and 1, respectively. At the same time, MPA obtains the smallest number of features selected. The average convergence curves of these datasets, shown in **Fig. 5** show



(a) AA dataset. (b) base-leuk1 dataset. (c) BreastEW dataset.

(d) CongressEW dataset. (e) EXactly dataset. (f) kvskpEW dataset.

(g) lonsophereEW dataset. (h) Lymphography dataset. (i) M-of-n dataset.

(j) ND dataset. (k) panglung dataset. (l) prost dataset.

(m) sonarEW dataset. (n) spectEW dataset. (o) base-leuk1 dataset.

(p) waveform dataset. (q) winew dataset. (r) zoo dataset.

FIGURE 6. **Boxplots of the results obtained using the MPA-kNN and 1000 iterations used as stop criteria over the eighteen datasets.**

that MPA reaches better performance in few iterations than other algorithms. **Fig. 5** emphasize the results of **Table 4**, **Table 5**, **Table 6**, and **Table 7**. MPA performs well overall and it returns the best solution in the smallest number of iterations compared to other algorithms like GWO, WOA, and SSA.

It is well accepted that MPA reduces the number of features and maximizes accuracy. The convergence curve depicts the relationship between the fitness function and the number of iterations in graphical form. The convergence curve reveals the best-performing algorithm from a comparison between various approaches. when the number of iterations increases, it indicates a direct correlation.

Fig. 6 presents the boxplots for all datasets, in which algorithms are compared graphically and visually based on their average performance. Boxplot has five elements: minimum, maximum, median, first quartile, and third quartile of the data. The box mat has a line inside, which refers to the median value. Note that the boxplots reflect the classification accuracy of algorithms. It can be seen that MPA has higher boxplots compared to the other algorithms. In addition, the median of the MPA algorithm has a higher value compared to the other algorithms. The second-best algorithm is either WOA or SSA depending on the dataset.

Finally, it can be seen that

- **box plots** allow us to observe that MPA-KNN shows superior performance to Grey Wolf Optimizer (GWO), Moth Flame Optimization Algorithm (MFO), Sine Cosine Algorithm (SCA), Whale Optimization Algorithm (WOA), Slap Swarm Algorithm (SSA), Butterfly Optimization Algorithm (BOA), and Harris Hawks Optimization (HHO).
- K-fold cross-validation is a perfect choice to prevent the problem of overfitting.
- KNN classifier is a classification method that provides high quality solutions while it learns effectively from the training data.
- The proposed algorithm's performance is compared to the other seven algorithms performance. The results show our proposed algorithm's superiority in terms of classification, accuracy, the number of selected features, and specificity.
- The experimental results illustrate the effectiveness of the proposed MPA algorithm for feature selection.

VI. CONCLUSION

This paper aimed to propose a new approach to feature selection based on the Marine Predators Algorithm (MPA). It is essential to state that to our knowledge; this is the first work that uses the MPA algorithm to solve the F.S. problem. The experiments are applied on eighteen standard benchmark datasets from UCI datasets to investigate the performance of the proposed MPA approach, and five evaluation criteria are assessed to evaluate different aspects of the performance of comparative algorithms.

The experimental results showed that the proposed MPA-KNN approach achieved superior results compared to the seven well-known meta-heuristic algorithms from recent literature, including DA, WOA, GOA, GWO, SSA, and other algorithms GWO, MFO, SCA, WOA, SSA, BOA, and HHO.

The results proved that for most datasets, when applied with k-NN as the classifiers, the MPA achieved the smallest number of selected features with the best classification accuracy in an Acceptable time.

For significantly large data sets, the MPA showed a significant advantage. MPA ranked as the first algorithm in average Accuracy, Sensitivity, Specificity, and standard deviation and obtained the smallest number of selected features.SSA is ranked the second algorithm in performance after MPA. While SSA has a high average accuracy in some datasets, MPA provides the best results in standard deviation for the same dataset. We conclude that the proposed MPA-KNN approach achieved high performance compared to the other algorithms tested and reduced the number of significant features selected. This can make it easier for physicians to diagnose diseases and develop medical treatments quickly and accurately, making a complicated and sensitive process more efficient and effective and ultimately helping patients.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

CRediT AUTHOR STATEMENT

All authors contributed equally to this paper, where; Diaa Salama Abd Elminaam: Supervision, Methodology, Conceptualization, Formal analysis, Writing - review & editing. Ayman Nabil: Methodology, Formal analysis, Methodology, Writing - review & editing, Implementation the code and running. Shimaa A. Ibraheem: Software, Formal analysis, Resources, Writing - original draft. Essam H. Houssein: Supervision, Conceptualization, Methodology, Formal analysis, Writing - review & editing. All authors read and approved the final paper.

REFERENCES

[1] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, Jul. 2018.
[2] M. Mafarja, A. Qasem, A. A. Heidari, I. Aljarah, H. Faris, and S. Mirjalili, "Efficient hybrid nature-inspired binary optimizers for feature selection," *Cognit. Comput.*, vol. 12, no. 1, pp. 150–175, Jan. 2020.
[3] H. Zamani and M.-H. Nadimi-Shahraki, "Feature selection based on whale optimization algorithm for diseases diagnosis," *Int. J. Comput. Sci. Inf. Secur.*, vol. 14, no. 9, p. 1243, 2016.
[4] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Comput. Surv.*, vol. 50, no. 6, pp. 1–45, 2017.
[5] H. Faris, A. A. Heidari, A. M. Al-Zoubi, M. Mafarja, I. Aljarah, M. Eshtay, and S. Mirjalili, "Time-varying hierarchical chains of salps with random weight networks for feature selection," *Expert Syst. Appl.*, vol. 140, Feb. 2020, Art. no. 112898.
[6] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*, vol. 454. New York, NY, USA: Springer, 2012.
[7] A. M. Al-Zoubi, H. Faris, J. Alqatawna, and M. A. Hassonah, "Evolving support vector machines using whale optimization algorithm for spam profiles detection on online social networks in different lingual contexts," *Knowl.-Based Syst.*, vol. 153, pp. 91–104, Aug. 2018.
[8] R. Zhang, F. Nie, Y. Wang, and X. Li, "Unsupervised feature selection via adaptive multimeasure fusion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2886–2892, Sep. 2019.
[9] Y. Han, Y. Yang, Y. Yan, Z. Ma, N. Sebe, and X. Zhou, "Semisupervised feature selection via spline regression for video semantic recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 2, pp. 252–264, Feb. 2015.

[10] R. Li, J. Lu, Y. Zhang, and T. Zhao, "Dynamic AdaBoost learning with feature selection based on parallel genetic algorithm for image annotation," *Knowl.-Based Syst.*, vol. 23, no. 3, pp. 195–201, Apr. 2010.
[11] H. Liu and R. Setiono, "Chi2: Feature selection and discretization of numeric attributes," in *Proc. 7th IEEE Int. Conf. Tools Artif. Intell.*, Nov. 1995, pp. 388–391.
[12] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.
[13] J. Quinlan, *C4. 5: Programs for Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann, 1993.
[14] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF," *Mach. Learn.*, vol. 53, nos. 1–2, pp. 23–69, Oct. 2003.
[15] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, nos. 1–2, pp. 273–324, Dec. 1997.
[16] I.-S. Oh, J.-S. Lee, and B.-R. Moon, "Hybrid genetic algorithms for feature selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1424–1437, Nov. 2004.
[17] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint 1-norms minimization," in *Proc. Adv. neural Inf. Process. Syst.*, 2010, pp. 1813–1821.
[18] E.-G. Talbi, *Metaheuristics: From Design to Implementation*, vol. 74. Hoboken, NJ, USA: Wiley, 2009.
[19] P. Shrivastava, A. Shukla, P. Vepakomma, N. Bhansali, and K. Verma, "A survey of nature-inspired algorithms for feature selection to identify Parkinson's disease," *Comput. Methods Programs Biomed.*, vol. 139, pp. 171–179, Feb. 2017.
[20] J. H. Holland, "Genetic algorithms," *Sci. Amer.*, vol. 267, no. 1, pp. 66–73, 1992.
[21] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proc. IEEE Int. Conf. Syst., Man, Cybern., Comput. Cybern. Simul.*, Oct. 1997, pp. 4104–4108.
[22] M. Mafarja and N. R. Sabar, "Rank based binary particle swarm optimisation for feature selection in classification," in *Proc. 2nd Int. Conf. Future Netw. Distrib. Syst.*, Jun. 2018, pp. 1–6.
[23] Q. Al-Tashi, H. M. Rais, S. J. Abdulkadir, S. Mirjalili, and H. Alhussian, "A review of grey wolf optimizer-based feature selection methods for classification," in *Evolutionary Machine Learning Techniques*. Springer, 2020, pp. 273–286.
[24] A. A. Heidari, R. A. Abbaspour, and A. R. Jordehi, "An efficient chaotic water cycle algorithm for optimization tasks," *Neural Comput. Appl.*, vol. 28, no. 1, pp. 57–85, Jan. 2017.
[25] A. G. Hussien, A. E. Hassanien, E. H. Houssein, S. Bhattacharyya, and M. Amin, "S-shaped binary whale optimization algorithm for feature selection," in *Recent Trends in Signal and Image Processing*. India: Springer, 2019, pp. 79–87.
[26] R. Krithiga and E. Ilavarasan, "A novel hybrid algorithm to classify spam profiles in Twitter," *Webology*, vol. 17, no. 1, pp. 260–279, Jun. 2020.
[27] R. Sawhney, P. Mathur, and R. Shankar, "A firefly algorithm based wrapper-penalty feature selection method for cancer diagnosis," in *Proc. Int. Conf. Comput. Sci. Appl.* London, U.K.: Springer-Verlag, 2018, pp. 438–449.
[28] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017.
[29] H. Faris, M. M. Mafarja, A. A. Heidari, I. Aljarah, A. M. Al-Zoubi, S. Mirjalili, and H. Fujita, "An efficient binary Salp swarm algorithm with crossover scheme for feature selection problems," *Knowl.-Based Syst.*, vol. 154, pp. 43–67, Aug. 2018.
[30] G. I. Sayed, G. Khoriba, and M. H. Haggag, "A novel chaotic Salp swarm algorithm for global optimization and feature selection," *Int. J. Speech Technol.*, vol. 48, no. 10, pp. 3462–3481, Oct. 2018.
[31] S. Harifi, M. Khalilian, J. Mohammadzadeh, and S. Ebrahimnejad, "Emperor penguins colony: A new Metaheuristic algorithm for optimization," *Evol. Intell.*, vol. 12, no. 2, pp. 211–226, Jun. 2019.
[32] T. Zheng and W. Luo, "An improved squirrel search algorithm for optimization," *Complexity*, vol. 2019, pp. 1–31, Jul. 2019.
[33] Y. Wang and T. Du, "An improved squirrel search algorithm for global function optimization," *Algorithms*, vol. 12, no. 4, p. 80, Apr. 2019.
[34] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Gener. Comput. Syst.*, vol. 111, pp. 300–323, Oct. 2020.
[35] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: A nature-inspired Metaheuristic," *Expert Syst. Appl.*, vol. 152, Aug. 2020, Art. no. 113377.
[36] I. Tumar, Y. Hassouneh, H. Turabieh, and T. Thaher, "Enhanced binary moth flame optimization as a feature selection algorithm to predict software fault prediction," *IEEE Access*, vol. 8, pp. 8041–8055, 2020.
[37] M. Shehab, L. Abualigah, H. Al Hamad, H. Alabool, M. Alshinwan, and A. M. Khasawneh, "Moth–flame optimization algorithm: Variants and applications," *Neural Comput. Appl.*, vol. 32, no. 14, pp. 9859–9884, Jul. 2020.
[38] S. Arora and S. Singh, "An improved butterfly optimization algorithm with chaos," *J. Intell. Fuzzy Syst.*, vol. 32, no. 1, pp. 1079–1088, Jan. 2017.
[39] M. Tubishat, M. Alswaitti, S. Mirjalili, M. A. Al-Garadi, M. T. Alrashdan, and T. A. Rana, "Dynamic butterfly optimization algorithm for feature selection," *IEEE Access*, vol. 8, pp. 194303–194314, 2020.
[40] D. W. Sims, E. J. Southall, N. E. Humphries, G. C. Hays, C. J. A. Bradshaw, J. W. Pitchford, A. James, M. Z. Ahmed, A. S. Brierley, M. A. Hindell, D. Morritt, M. K. Musyl, D. Righton, E. L. C. Shepard, V. J. Wearmouth, R. P. Wilson, M. J. Witt, and J. D. Metcalfe, "Scaling laws of marine predator search behaviour," *Nature*, vol. 451, no. 7182, pp. 1098–1102, Feb. 2008.
[41] N. E. Humphries, N. Queiroz, J. R. Dyer, N. G. Pade, M. K. Musyl, K. M. Schaefer, D. W. Fuller, J. M. Brunnschweiler, T. K. Doyle, J. D. Houghton, and G. C. Hays, "Environmental context explains Lévy and Brownian movement patterns of marine predators," *Nature*, vol. 465, no. 7301, pp. 1066–1069, 2010.
[42] G. M. Viswanathan, E. P. Raposo, and M. G. E. da Luz, "Lévy flights and superdiffusion in the context of biological encounters and random searches," *Phys. Life Rev.*, vol. 5, no. 3, pp. 133–150, Sep. 2008.
[43] J.-H. Seo, Y. H. Lee, and Y.-H. Kim, "Feature selection for very short-term heavy rainfall prediction using evolutionary computation," *Adv. Meteorol.*, vol. 2014, pp. 1–15, Jan. 2014.
[44] T. R. Biyanto, Matradji, S. Irawan, H. Y. Febrianto, N. Afdanny, A. H. Rahman, K. S. Gunawan, J. A. D. Pratama, and T. N. Bethiana, "Killer whale algorithm: An algorithm inspired by the life of killer whale," *Procedia Comput. Sci.*, vol. 124, pp. 151–157, Jan. 2017.
[45] A. Zarshenas and K. Suzuki, "Binary coordinate ascent: An efficient optimization technique for feature subset selection for machine learning," *Knowl.-Based Syst.*, vol. 110, pp. 191–201, Oct. 2016.
[46] F. Tan, X. Fu, Y. Zhang, and A. G. Bourgeois, "A genetic algorithm-based method for feature subset selection," *Soft Comput.*, vol. 12, no. 2, pp. 111–120, Sep. 2007.
[47] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary ant lion approaches for feature selection," *Neurocomputing*, vol. 213, pp. 54–65, Nov. 2016.
[48] Y.-S. Jeong, K. S. Shin, and M. K. Jeong, "An evolutionary algorithm with the partial sequential forward floating search mutation for large-scale feature selection problems," *J. Oper. Res. Soc.*, vol. 66, no. 4, pp. 529–538, Apr. 2015.
[49] M. Mafarja and S. Mirjalili, "Whale optimization approaches for wrapper feature selection," *Appl. Soft Comput.*, vol. 62, pp. 441–453, Jan. 2018.
[50] S. C. Yusta, "Different Metaheuristic strategies to solve the feature selection problem," *Pattern Recognit. Lett.*, vol. 30, no. 5, pp. 525–534, Apr. 2009.
[51] S. Oreski and G. Oreski, "Genetic algorithm-based heuristic for feature selection in credit risk assessment," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 2052–2064, Mar. 2014.
[52] K. Mistry, L. Zhang, S. C. Neoh, C. P. Lim, and B. Fielding, "A micro-GA embedded PSO feature selection approach to intelligent facial emotion recognition," *IEEE Trans. Cybern.*, vol. 47, no. 6, pp. 1496–1509, Jun. 2017.
[53] L. Brezocnik, "Feature selection for classification using particle swarm optimization," in *Proc. IEEE EUROCON -17th Int. Conf. Smart Technol.*, Jul. 2017, pp. 966–971.
[54] J. Too, A. R. Abdullah, and N. M. Saad, "A new co-evolution binary particle swarm optimization with multiple inertia weight strategy for feature selection," in *Informatics*, vol. 6. Basel, Switzerland: Multidisciplinary Digital Publishing Institute, 2019, p. 21.
[55] M. M. Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neurocomputing*, vol. 260, pp. 302–312, Oct. 2017.
[56] Q. Al-Tashi, S. J. A. Kadir, H. M. Rais, S. Mirjalili, and H. Alhussian, "Binary optimization using hybrid grey wolf optimization for feature selection," *IEEE Access*, vol. 7, pp. 39496–39508, 2019.

[57] E. H. Houssein, M. E. Hosney, D. Oliva, W. M. Mohamed, and M. Hassaballah, "A novel hybrid harris hawks optimization and support vector machines for drug design and discovery," *Comput. Chem. Eng.*, vol. 133, Feb. 2020, Art. no. 106656.
[58] N. Neggaz, A. A. Ewees, M. A. Elaziz, and M. Mafarja, "Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection," *Expert Syst. Appl.*, vol. 145, May 2020, Art. no. 113103.
[59] N. Neggaz, E. H. Houssein, and K. Hussain, "An efficient henry gas solubility optimization for feature selection," *Expert Syst. Appl.*, vol. 152, Aug. 2020, Art. no. 113364.
[60] M. A. Tawhid and K. B. Dsouza, "Hybrid binary bat enhanced particle swarm optimization algorithm for solving feature selection problems," *Appl. Comput. Informat.*, vol. 16, nos. 1–2, pp. 117–136, Apr. 2018.
[61] G. M. Viswanathan, S. V. Buldyrev, S. Havlin, M. G. E. da Luz, E. P. Raposo, and H. E. Stanley, "Optimizing the success of random searches," *Nature*, vol. 401, no. 6756, pp. 911–914, Oct. 1999.
[62] M. Ghosh, R. Guha, I. Alam, P. Lohariwal, D. Jalan, and R. Sarkar, "Binary genetic swarm optimization: A combination of GA and PSO for feature selection," *J. Intell. Syst.*, vol. 29, no. 1, pp. 1598–1610, Sep. 2019.
[63] D. Oliva, M. A. Elaziz, A. H. Elsheikh, and A. A. Ewees, "A review on meta-heuristics methods for estimating parameters of solar cells," *J. Power Sources*, vol. 435, Sep. 2019, Art. no. 126683.
[64] T. A. Shehabeldeen, M. A. Elaziz, A. H. Elsheikh, and J. Zhou, "Modeling of friction stir welding process using adaptive neuro-fuzzy inference system integrated with harris hawks optimizer," *J. Mater. Res. Technol.*, vol. 8, no. 6, pp. 5882–5892, Nov. 2019.
[65] 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, Dec. 2019.
[66] M. A. A. Al-qaness, A. A. Ewees, H. Fan, L. Abualigah, and M. A. Elaziz, "Marine predators algorithm for forecasting confirmed cases of COVID-19 in Italy, USA, Iran and Korea," *Int. J. Environ. Res. Public Health*, vol. 17, no. 10, p. 3520, May 2020.
[67] 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," *Sci. Rep.*, vol. 10, no. 1, pp. 1–15, Dec. 2020.
[68] M. A. Elaziz, A. A. Ewees, D. Yousri, H. S. N. Alwerfali, Q. A. Awad, S. Lu, and M. A. A. Al-Qaness, "An improved marine predators algorithm with fuzzy entropy for multi-level thresholding: Real world example of COVID-19 CT image segmentation," *IEEE Access*, vol. 8, pp. 125306–125330, 2020.
[69] M. Abd Elaziz, T. A. Shehabeldeen, A. H. Elsheikh, J. Zhou, A. A. Ewees, and M. A. A. Al-qaness, "Utilization of random vector functional link integrated with marine predators algorithm for tensile behavior prediction of dissimilar friction stir welded aluminum alloy joints," *J. Mater. Res. Technol.*, vol. 9, no. 5, pp. 11370–11381, Sep. 2020.
[70] X.-S. Yang, "Firefly algorithm, levy flights and global optimization," in *Research and Development in Intelligent Systems XXVI*. London, U.K.: Springer-Verlag, 2010, pp. 209–218.
[71] F. Bartumeus, M. G. E. da Luz, G. M. Viswanathan, and J. Catalan, "Animal search strategies: A quantitative random-walk analysis," *Ecology*, vol. 86, no. 11, pp. 3078–3087, Nov. 2005.
[72] J. D. Filmalter, L. Dagorn, P. D. Cowley, and M. Taquet, "First descriptions of the behavior of silky sharks, carcharhinus falciformis, around drifting fish aggregating devices in the Indian Ocean," *Bull. Mar. Sci.*, vol. 87, no. 3, pp. 325–337, Jul. 2011.
[73] R. N. Mantegna, "Fast, accurate algorithm for numerical simulation of levy stable stochastic processes," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 49, no. 5, p. 4677, 1994.
[74] A. Einstein, *Investigations Theory Brownian Movement*. Chelmsford, MA, USA: Courier Corporation, 1956.
[75] R. P. Parouha and K. N. Das, "A memory based differential evolution algorithm for unconstrained optimization," *Appl. Soft Comput.*, vol. 38, pp. 501–517, Jan. 2016.
[76] W. Bao, N. Lianju, and K. Yue, "Integration of unsupervised and supervised machine learning algorithms for credit risk assessment," *Expert Syst. Appl.*, vol. 128, pp. 301–315, Aug. 2019.
[77] M. Bennasar, Y. Hicks, and R. Setchi, "Feature selection using joint mutual information maximisation," *Expert Syst. Appl.*, vol. 42, no. 22, pp. 8520–8532, Dec. 2015.
[78] D. Dheeru and E. K. Taniskidou, "Uci machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, Irvine, CA, USA, 2017.

DIAA SALAMA ABD ELMINAAM received the Ph.D. degree in information system from the Faculty of Computers and Information, Menufia University, Egypt, in 2015. He is currently an Associate Professor with the Data Science Department, Faculty of Computers, Misr International University, Egypt, on leave from Benha University, Egypt, in February 2020. He has worked on several research topics. He has contributed more than 80 technical articles in the areas of optimization, meta heuristics, feature selection, opinion mining, machine and deep learning, wireless networks, wireless network security, information security and Internet applications, cloud computing, mobile cloud computing, the Internet of Things, and machine learning in international journals, international conferences, local journals, and local conferences. He majors in optimization, cryptography, network security, the IoT, big data, cloud computing, deep learning.

AYMAN NABIL received the B.S. (Hons.), master's, and Ph.D. degrees from the Faculty of Computer and Information, Helwan University, Egypt, in 2002. He worked with Misr International University, Egypt. He is specializing in uncertainty, and knowledge discovery approaches. He has published articles under artificial intelligence area in international journals, international conferences, local journals, and local conferences.

SHIMAA A. IBRAHEEM was born in KafrShokr, Kalyobia, Egypt, in November 1981. She received the B.S. degree (Hons.) from the Faculty of Computers and Informatics, Zagazig University, Egypt, in 2003, and the Ph.D. degree in information system from the Faculty of Computers and Information, Menufia University, Egypt, in 2016, specializing in uncertainty approaches of AI. Her main research interests include machine learning, optimization techniques, and metaheuristics.

ESSAM H. HOUSSEIN received the Ph.D. degree in computer science "wireless networks based on artificial intelligence," in 2012. He is currently working as an Associate Professor with the Faculty of Computers and Information, Minia University, Egypt. He is also the Founder of the Computing & Artificial Intelligence Research Group (CAIRG), Egypt. He has more than 80 scientific research articles published in prestigious international journals in the topics of optimization, machine learning, image processing, the IoT, and its applications. He also serves as a reviewer for more than 30 journals (Elsevier, Springer, and IEEE). His research interests include wireless sensor networks, the IoT, bioinformatics and biomedical, cloud computing, soft computing, image processing, artificial intelligence, data mining, optimization, and meta-heuristics techniques.