IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received May 23, 2020, accepted June 4, 2020, date of publication June 9, 2020, date of current version June 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3001151

MbGWO-SFS: Modified Binary Grey Wolf Optimizer Based on Stochastic Fractal Search for Feature Selection

EL-SAYED M. EL-KENAWY[1], (Member, IEEE), MARWA METWALLY EID[1], MOHAMED SABER[2], AND ABDELHAMEED IBRAHIM[3]
[1]Department of Communications and Electronics, Delta Higher Institute for Engineering and Technology (DHIET), Mansoura 35111, Egypt
[2]Faculty of Engineering, Delta University for Science and Technology, Mansoura 11152, Egypt
[3]Department of Computer Engineering and Control Systems, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt

Corresponding authors: El-Sayed M. El-Kenawy (skenawy@ieee.org) and Abdelhameed Ibrahim (afai79@mans.edu.eg)

ABSTRACT Grey Wolf Optimizer (GWO) simulates the grey wolves' nature in leadership and hunting manners. GWO showed a good performance in the literature as a meta-heuristic algorithm for feature selection problems, however, it shows low precision and slow convergence. This paper proposes a Modified Binary GWO (MbGWO) based on Stochastic Fractal Search (SFS) to identify the main features by achieving the exploration and exploitation balance. First, the modified GWO is developed by applying an exponential form for the number of iterations of the original GWO to increase the search space accordingly exploitation and the crossover/mutation operations to increase the diversity of the population to enhance exploitation capability. Then, the diffusion procedure of SFS is applied for the best solution of the modified GWO by using the Gaussian distribution method for random walk in a growth process. The continuous values of the proposed algorithm are then converted into binary values so that it can be used for the problem of feature selection. To ensure the stability and robustness of the proposed MbGWO-SFS algorithm, nineteen datasets from the UCI machine learning repository are tested. The K-Nearest Neighbor (KNN) is used for classification tasks to measure the quality of the selected subset of features. The results, compared to binary versions of the-state-of-the-art optimization techniques such as the original GWO, SFS, Particle Swarm Optimization (PSO), hybrid of PSO and GWO, Satin Bowerbird Optimizer (SBO), Whale Optimization Algorithm (WOA), Multiverse Optimization (MVO), Firefly Algorithm (FA), and Genetic Algorithm (GA), show the superiority of the proposed algorithm. The statistical analysis by Wilcoxon's rank-sum test is done at the 0.05 significance level to verify that the proposed algorithm can work significantly better than its competitors in a statistical way.

INDEX TERMS Feature selection, meta-heuristics, stochastic fractal search, binary optimizer, K-Nearest Neighbor, Wilcoxon's rank-sum test.

I. INTRODUCTION

The optimization process is existing in several research areas such as engineering, medical, agriculture, computer science, and feature selection. In optimization, the main target is to select the optimal solution of a given problem from the available solutions concerning the problem description. Moreover, in optimization algorithms, there is a target that should be minimized or maximized according to the problem to be solved [1], [2]. Filter, wrapper, and hybrid-based are the main categorize of feature selection techniques [3]. The filter-based feature selection techniques or traditional feature selection techniques have an advantage that it is speed and ability to scale to a large dataset. The process of feature selection is often most useful in situations in which wrappers may over-fit such as Information Gain (IG). IG measures how much information a feature can give us about the class and it is useful in reducing the number of features that can give more accuracy in classification model [4].

The search space for selecting features is reduced in the wrapper technique which is accurate but needs much time to include learning algorithms as a part of the select function.

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

Genetic algorithms (GA) are randomly based algorithms on the process of natural selection underlying biological evolution. They can be applied to many challenges, optimization, machine learning problems, and feature selection [5]. To do wrapper feature selection, one needs to utilize an optimization algorithm, however, the classical optimization techniques are somehow restricted in solving the problems. Thus, the evolutionary computation (EC) algorithms are considered as an alternative in searching for the problems' optimum solution and solving the mentioned limitations. Swarm-based algorithms are inspired by nature, biological behavior, and social behavior of animals, birds, whales, bat, grasshopper, firefly, salp, fish, wolves, etc. [6]–[9]. Many kinds of research used optimization to solve a given problem such as the Whale Optimization Algorithm (WOA) [10], [11]. WOA can be used to find the optimal weights to train the neural network. A multi-objective version of WOA is evolved and applied to the problem of forecasting the wind speed in [12].

Another algorithm is the Grey Wolf Optimizer (GWO). GWO is an optimization algorithm that simulates the grey wolves in nature [2], [7], [13]. GWO has the advantages of simplicity, flexibility, deprivation-free mechanism, and the ability to avoid the local optima. Because of that, it has been used in many research areas in the last years such as feature subset selection [1], DC motors control [14], [15], solving optimal reactive power dispatch problem [16], financial crisis prediction [13], and in some applications, the GWO algorithm was used to train the Multilayer Perceptron (MLP) network [17]. For the problem of feature selection, the solution can be represented as a vector of features with size n, which is the number of features and the vector items can be binary values with 1 (the feature is included) and 0 (the feature is not included). Hence, GWO starts with an initial random population of vectors holding randomly selected features. Then, using the exploration and exploitation capabilities, GWO can find the optimal subset of features. The wrapped feature selection methods have a learning algorithm to evaluate the selected subset of features quality [7].

Recently, to solve the feature selection problems, a binary GWO algorithm is integrated with a multi-phase mutation in [7] based on the wrapper methods. In [18], a multi-strategy ensemble GWO is proposed. This method overcomes the single search strategy limitation of GWO in solving function optimization problems. Another research proposed a hierarchy strengthened GWO (HSGWO) algorithm in [19] for solving large-scale problems. To improve the accuracy of identification, a chaos-based grey wolf optimization (EGWO) algorithm is proposed in [20] to find the optimal feature sets. Hybrid algorithms are also proposed for improving the GWO performance for different applications. In [21], a fusion between Particle Swarm Optimization (PSO) exploitation ability with the GWO exploration ability is proposed. Their algorithm was evaluated based on benchmark functions and real-world problems. Another research proposed a hybrid of GWO with a Crow Search Algorithm (CSA) (GWOCSA) in [22]. This hybrid algorithm combines both algorithms' strengths to generate a promising solution for achieving global optima efficiently.

In this paper, a Modified Binary GWO based on Stochastic Fractal Search (SFS) is proposed. The proposed algorithm achieves the exploration and exploitation balance in the identification of the main features. First, a modified GWO is developed by applying an exponential form for parameter a of the original GWO to increase the search space and crossover/mutation operations to increase the diversity of the population. Then, the SFS diffusion process is applied for the modified GWO, the best solution, by using the Gaussian distribution method for random walk in the growth process. The continuous values of the proposed algorithm are then converted into binary values so that it can be used for the problem of feature selection. To ensure the stability and robustness of the proposed MbGWO-SFS algorithm, nineteen datasets from the repository of the UCI machine learning are tested including two datasets with more than 500 attributes. As a preprocessing step, the class imbalance of the datasets is solved using the LSH-SMOTE algorithm [5] to improve the processing time. Compared to the binary versions of the-state-of-the-art optimization techniques of the original GWO [1], SFS [23], PSO [24], hybrid of PSO and GWO [21], Satin Bowerbird Optimizer (SBO) [25], WOA [26], Multiverse Optimization (MVO) [27], and Firefly Algorithm (FA) [28], in addition to, GA [29] and hybrid of GA and GWO, the results show the superiority of the proposed algorithm. In the experiments, the K-Nearest Neighbor (KNN) [30] is used for classification tasks to measure the quality of the selected subset of features. The statistical test of Wilcoxon's rank-sum is done at the 0.05 significance level to determine the significant difference between the results of the proposed algorithm and the other comparison algorithms in a statistically way.

This paper is organized into seven sections. The related work is presented in Section II. Section III shows the background of the basic mechanisms used in this work. The proposed algorithm (MbGWO-SFS) is described in detail in Section IV. Sections V and section VI show the evaluation metrics and the experimental results. Lastly, conclusions are stated in Section VII.

II. RELATED WORK

The optimizer of the grey wolf has been applied in the literature for different research directions such as face recognition, gene selection, electromyography classification, diagnoses of diseases, interference detection systems, and feature selection. The binary form of GWO can be used for feature selection and classification problems efficiently [36]–[38]. Table 1 shows a summary of some binary GWO algorithm in the literature. Binary GWO algorithms have been introduced in [31], [32] to select the subset of features for wrapper feature selection and classification. In these algorithms, a KNN classifier was used as a fitness function to evaluate the selected features subsets. Eight benchmark datasets were applied from the machine learning repository for evaluation. The methods

TABLE 1. Feature selection based on binary GWO.

Algorithm	Classifier	Method	Application
BGWOPSO [1]	kNN	Wrapper	UCI benchmark dataset
BGWO [31]	kNN	Wrapper	UCI benchmark dataset
bGWO1 [32]	kNN	Wrapper	UCI benchmark dataset
bGWO2 [32]	kNN	Wrapper	UCI benchmark dataset
BGWO [33]	C4.5 Decision Tree	Wrapper	Microarray cancer dataset
BoKF-BGWO [34]	kNN	Wrapper	Publicly available ICPR 2016 database
Improved BGWO [35]	kNN	Wrapper	UCI benchmark dataset

were compared with PSO and GA algorithms to show the effectiveness of their proposed methods in the experiments in terms of accuracy and reduction in the number of features. Another binary GWO wrapper method was presented in [33] to classify cancer on gene expression data. They used classifiers with cross-validation based on a decision tree C4.5. Ten microarray cancer datasets were used to evaluate their method and a comparison with Self-Organizing Map (SOM), MLP, and Support Vector Machine (SVM) was provided.

Recently, authors in [1] proposed a binary GWO based on PSO and they used the KNN classifier. They have assessed the performance of their method by using eighteen standard benchmark datasets from the repository of machine learning and compared their proposed method with different optimization approaches such as PSO, GA, and GWO to prove the enhancement in computational time, classification accuracy, and the number of selected features. In [34], a method based on Bag-of-Keypoint Features (BoKF) model and Binary GWO (BGWO) is proposed to distinguish nucleolar and centromere staining patterns. Authors in [35] introduced five transfer functions to get the binary values from the continuous values. They proposed an updating equation for the *a* parameter to balance between the local and global search.

Stochastic Fractal Search (SFS) was proposed firstly in [23] based on the fractal concept, which is a self-similarity property of objects. A chaotic SFS (CSFS) algorithm was introduced in [39] to improve SFS performance. This method integrated ten chaotic maps into the original SFS algorithm. The algorithm random scheme is replaced by the chaotic maps to enhance the accuracy of the solution and convergence speed of the original SFS. Recently, a modified SFS (MSFS) algorithm was proposed in [40] to solve the problem of economic load dispatch. In this method, the power system constraints are taken into consideration. A Multi-Objective SFS (MOSFS) algorithm was proposed to solve complex multi-objective optimization problems for the first time in [41].

The binary GWO still suffers from achieving a high exploration capability. By creating new particles based on the diffusion procedure of SFS, which employed the Gaussian distribution method for random walk in the Diffusion Limited Aggregation (DLA) growth process, a high exploration capability can be achieved. A series of Gaussian walks participating in the diffusion process around the best solution $\overrightarrow{G_\alpha}$ can be listed and checked to get the best solution. This increases the capability of exploration in the proposed MbGWO based on the diffusion process of the SFS algorithm to get the best solution.

III. BACKGROUND

A. GREY WOLF OPTIMIZER

Grey wolf optimizer simulates the wolves' movements in the process of searching for prey. Wolves usually live in packs where a pack consists of from 5 to 12 wolves. One pack has four different kinds of wolves named alpha, beta, delta, and omega wolves [42]. The alpha wolves are making decisions in each pack. The beta wolves help the alpha wolves in making decisions. The delta wolves submit to alpha and beta. The omega wolves submit to other wolves. The GWO algorithm is shown in Algorithm 1 step by step.

Mathematically, the best solution is named the alpha ($\overrightarrow{G_\alpha}$), while beta ($\overrightarrow{G_\beta}$) and delta ($\overrightarrow{G_\delta}$) are the second and third best solutions. Other solutions are indicated as omega ($\overrightarrow{G_\omega}$). During the process of catching the prey as shown in Fig. 1, alpha, beta, and delta wolves guide other wolves as denoted in Equations (1, 2, 3, and 4).

$$\vec{G}(t+1) = \vec{G}_p(t) - \vec{A}.\vec{D} \tag{1}$$

$$\vec{D} = |\vec{C}.\vec{G}_p(t) - \vec{G}(t)| \tag{2}$$

where t is the current iteration, $\vec{A}$, $\vec{C}$ are coefficient vectors, $\vec{G}_p$ is the position of prey and $\vec{G}$ represents the wolf position. The $\vec{A}$, $\vec{C}$ vectors are computed as

$$\vec{A} = 2\vec{a}.\vec{r_1} - \vec{a} \tag{3}$$

$$\vec{C} = 2\vec{r_2} \tag{4}$$

where the components of $\vec{a}$ are decreasing linearly from 2 to 0 throughout iterations, and vectors $\vec{r_1}$, $\vec{r_2}$ are random values $\in [0, 1]$. The parameter $\vec{a}$ is updated and controls the balance of the exploration and exploitation processes [42]. The $\vec{a}$ values are computed as in the following equation [42]:

$$\vec{a} = 2 - t.\frac{2}{M_t} \tag{5}$$

where M_t is the available number of iterations for the optimizer.

The three best solutions, $\overrightarrow{G_\alpha}$, $\overrightarrow{G_\beta}$, and $\overrightarrow{G_\delta}$, guide other individuals ($\overrightarrow{G_\omega}$) to change their positions toward the estimated position of the prey as shown in Figure 1.

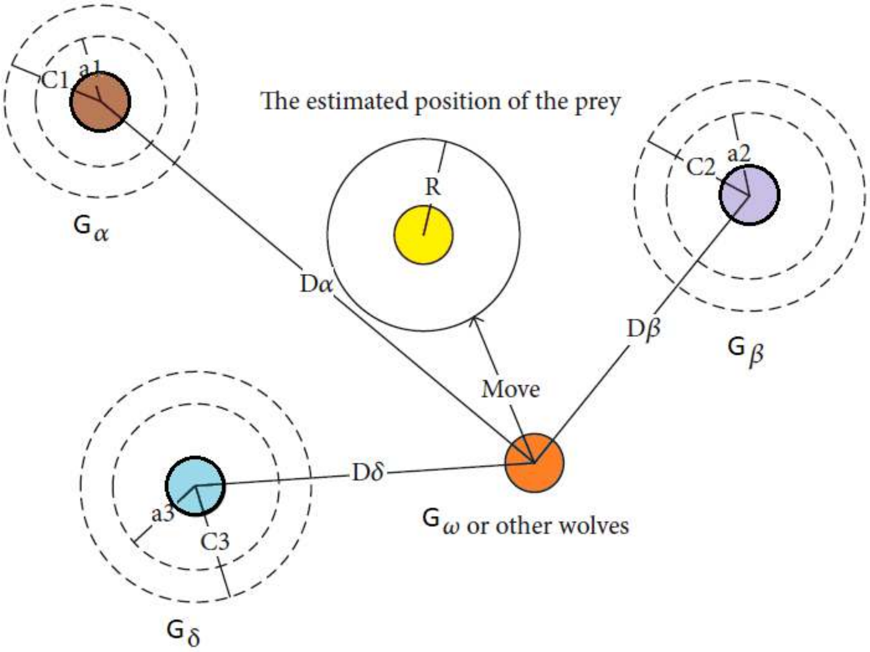


FIGURE 1. Position updating in the GWO algorithm.

Equations (6, 7, and 8) show the process of positions updating.

$$\begin{aligned}
\vec{D}_\alpha &= |\vec{C}_1.\vec{G}_\alpha - \vec{G}|,\\
\vec{D}_\beta &= |\vec{C}_2.\vec{G}_\beta - \vec{G}|,\\
\vec{D}_\delta &= |\vec{C}_3.\vec{G}_\delta - \vec{G}| \qquad (6)\\
\vec{G}_1 &= \vec{G}_\alpha - \vec{A}_1.\vec{D}_\alpha,\\
\vec{G}_2 &= \vec{G}_\beta - \vec{A}_2.\vec{D}_\beta,\\
\vec{G}_3 &= \vec{G}_\delta - \vec{A}_3.\vec{D}_\delta \qquad (7)
\end{aligned}$$

where $\vec{A}_1$, $\vec{A}_2$, $\vec{A}_3$ are calculated as in Eq. 3 and $\vec{C}_1$, $\vec{C}_2$, $\vec{C}_3$ are calculated as in Eq. 4. The updated positions for the population, $\vec{G}(t+1)$, can be expressed as an average of the three solutions of $\vec{G}_1$, $\vec{G}_2$, and $\vec{G}_3$ from Eq. 7 as follows

$$\vec{G}(t+1) = \frac{\vec{G}_1 + \vec{G}_2 + \vec{G}_3}{3} \qquad (8)$$

B. GENETIC ALGORITHM

Genetic algorithm (GA) is based on some techniques such as inheritance, mutation, crossover, and selection which are inspired by evolutionary biology. The algorithm uses the chromosomes/genes representation of living organisms [43]. In GA, a solution $x \in \zeta$ is an individual for ζ as the search space. Each chromosome x consists of discrete units or genes as, $x = [x_1; x_2; \ldots x_N]$, where x_i is the ith gene in chromosome x and N is the total number of genes or the dimension of the search space. The genes are usually represented by binary numbers and each chromosome is corresponding to a solution in the search space. The population of the GA is started randomly and the individuals are then generated. Crossover and mutation operators, as shown in Fig. 2, are used to get new generations and then all the individuals are evaluated to select the best individuals for the next iteration.

Algorithm 1 Pseudo Code of the Grey Wolf Optimizer

```
Initialize GWO population $\vec{G}_i (i = 1, 2, \ldots, n)$ with size $n$, maximum iterations number $M_t$, and fitness function $F_n$.
Initialize GWO parameters ($\vec{a}$, $\vec{A}$, $\vec{C}$)
Set t = 1. (initialize counter).
Calculate the fitness function $F_n$ for each $\vec{G}_i$
Find best, second best and third best individuals as $\vec{G}_\alpha, \vec{G}_\beta, \vec{G}_\delta$
while $t < M_t$ (Termination condition) do
    for ($i = 1 : i \le n + 1$) do
        Calculate $\vec{G}_1, \vec{G}_2, \vec{G}_3$ by Eq. 7
        Update individual positions based on Eq. 8
    end for
    Update ($\vec{a}$) by Eq. 5
    Update parameters ($\vec{A}$, $\vec{C}$)
    Calculate the fitness function $F_n$ for each $\vec{G}_i$
    Update $\vec{G}_\alpha, \vec{G}_\beta, \vec{G}_\delta$
    Set t = t + 1. (increase counter).
end while
return $\vec{G}_\alpha$
```

The GA has the following challenges:

- The agents are moved randomly in the entire search space, thus the algorithm may select sub-optimal solutions.

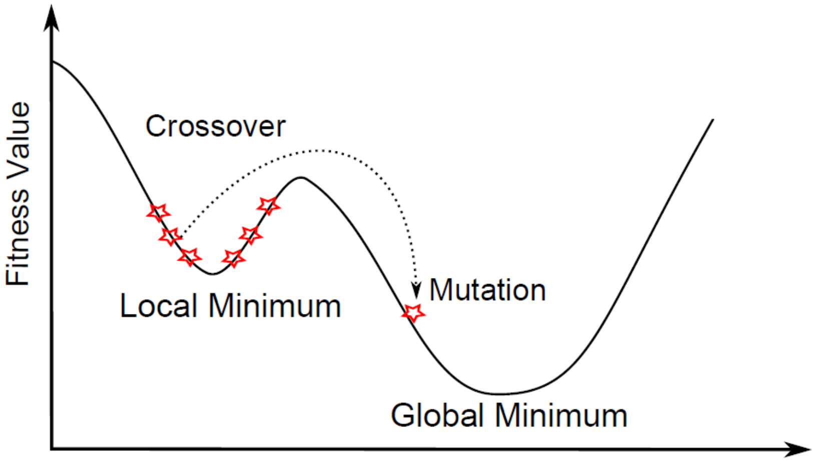


FIGURE 2. Crossover and mutation processes of genetic algorithm [43].

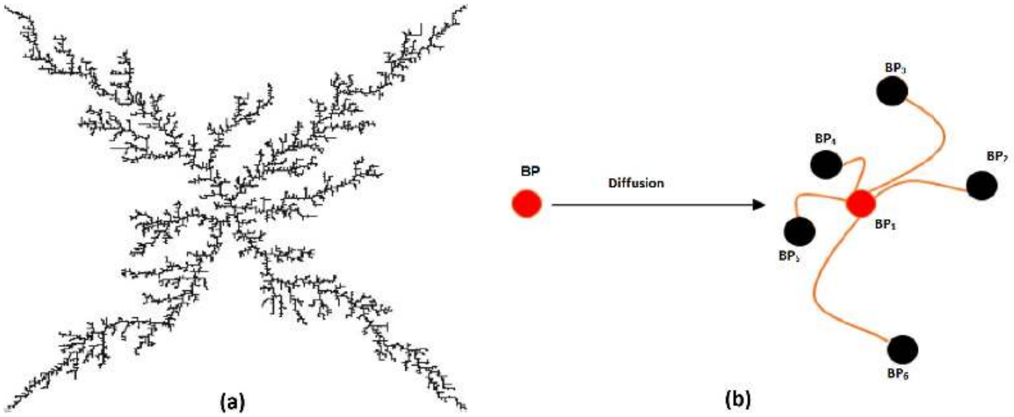


FIGURE 3. SFS fractal and diffusion processes; (a) Generate random fractal by DLA, (b) Diffusing the best particle.

- The exploration capability of the GA algorithm is very limited and it may trap into local minimum which is not the best solution (global minimum).
- The algorithm has slow convergence due to the encoding and decoding steps and more recent optimization algorithms are easier to be implemented than GA.

C. STOCHASTIC FRACTAL SEARCH

Using the characteristics of the original fractal method, a meta-heuristic algorithm can be inspire based on the random fractals in time consumption and accuracy [23]. To find a solution for a given problem, the basic Fractal Search (FS) method uses the following three simple rules

1) A particle can have electrical potential energy.
2) Each particle can diffuse and other random particles can be created. The original particle energy is distributed among the new particles.
3) In each generation, a few best particles are remaining and other particles are discarded.

Stochastic Fractal Search (SFS) was proposed based on the mathematical model of the fractal [23]. The author proposed a Fractal Search (FS) algorithm using the DLA method, which is employed to generate fractal-shaped objects. Figure 3 (a) shows a sample of random fractal generated by the DLA method. The main SFS structure consists of three processes of diffusion, first and second update processes to overcome the disadvantages of the FS algorithm. Figure 3 (b) presents the diffusion process in the SFS algorithm. A series of Gaussian walks participating in the diffusion process around the best solution (best particle) BP which can be listed around this best solution as BP_1, BP_2, BP_3, BP_4, BP_5.

D. K-NEAREST NEIGHBOR

In this work, a wrapper approach based on the K-Nearest Neighbor (KNN) classifier, a supervised learning algorithm, is used for feature selection [30]. In KNN, each sample is classified into a specific class label based on the majority of its K neighbors. To decide the class of the unknown instance, KNN uses training instances instead of building models.

In our experiments, KNN is used for classification tasks to measure the quality of the selected subset of features. The Euclidean distance, Euc_D, between features of the training data and features of the testing data is calculated to determine the nearest K neighbors to a sample as follows

$$Euc_D = \sqrt{\sum_{i=1}^{k} (Train\_F_i - Test\_F_i)^2} \tag{9}$$

where $Train\_F_i$ is a feature in the training data, $Test\_F_i$ is a feature in the testing data, and k is the number of features.

IV. MbGWO-SFS: MODIFIED BINARY GREY WOLF OPTIMIZER WITH STOCHASTIC FRACTAL SEARCH

This section shows the Modified binary Grey Wolf Optimizer (MbGWO) with the Stochastic Fractal Search (SFS) in detail. Also, the fitness function that is used to measure the quality of the original GWO solutions and the proposed algorithm solutions is presented. The proposed MbGWO-SFS algorithm is explained in Algorithm 2 step by step.

A. MODIFIED GREY WOLF OPTIMIZER

The process of finding the global minimum is a challenging task. GWO uses exploration and exploitation to do its job. GWO achieves the balance between exploration and exploitation, to avoid stagnation in local optimum and to converge on the global minimum, using the two parameters of $\overrightarrow{A}$ and $\overrightarrow{a}$. The value of $\overrightarrow{a}$ decreases linearly from 2 to 0 during iterations according to Eq. 5. Thus, part of the iterations are associated to exploration ($|\overrightarrow{A}| > 1$) and the remaining part is associated to exploitation ($|\overrightarrow{A}| < 1$).

1) EXPONENTIAL FORM

To achieve the balancing between exploration and exploitation, Eq. 5 is changed so that the value is decreasing exponentially throughout iteration as shown in Eq. 10. By apply this exponential change, the number of iterations that can be used for exploration is increased and hence the proposed modified GWO achieves higher exploration of the search space for more iterations. Figure 4 illustrates the difference between a linear and exponential change of the value of $\overrightarrow{a}$ which indicates that the exploration is achieved for a greater number of iterations.

$$\overrightarrow{a} = 2\left(1 - \left(\frac{t}{M_t}\right)^2\right) \tag{10}$$

where iteration number in denoted as t and the optimizer total number of iterations are denoted as M_t.

2) CROSSOVER AND MUTATION

The crossover is the operation that combines information of the different solutions to generate a new offspring, which is the way to generate new solutions from an existing population. The crossover operation increases the diversity of the population and enhances exploitation capability. A single-point crossover, cp_i, $i = 0$ to $N - 1$, is chosen randomly for a number with N bits. The offspring of the three suggested solutions of $(\overrightarrow{G_1}, \overrightarrow{G_2}, \overrightarrow{G_3})$ consists of the pre-cp_i section from first solution followed by the post-cp_i section of the next one as shown in Fig. 5. The following equation represents the crossover process

$$\begin{aligned} Offspring = [&\overrightarrow{G_1}(section < cp_i) + \overrightarrow{G_2}(cp_i > section),\\ &\overrightarrow{G_2}(section < cp_i) + \overrightarrow{G_3}(cp_i > section),\\ &\overrightarrow{G_3}(section < cp_i) + \overrightarrow{G_1}(cp_i > section)] \end{aligned} \tag{11}$$

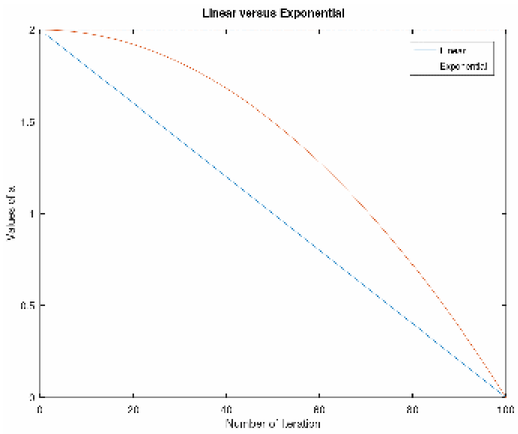


FIGURE 4. Linear change of Eq. 5 versus the exponential change of Eq. 10.

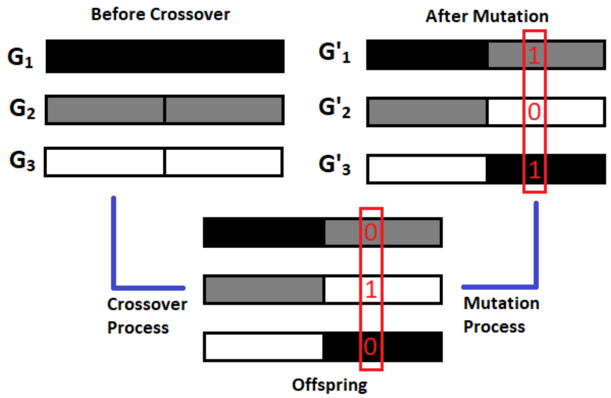


FIGURE 5. One point crossover and random mutation processes as in Equations 11 and 12.

The mutation operator changes one or more components of the offspring randomly. This is used to prevent premature convergence. The mutation operation is employed to enhance the position of a specific solution around randomly selected leaders. The positions are then updated as shown in Fig. 5 based on a random point mp_i, $i = 0$ to $N - 1$, which is chosen randomly for the offspring number with N bits. the following equation represents the crossover process

$$(\overrightarrow{G'_1}, \overrightarrow{G'_2}, \overrightarrow{G'_3}) = Mutation(Offspring) \tag{12}$$

where $\overrightarrow{G'_1}, \overrightarrow{G'_2}, \overrightarrow{G'_3}$ represent the updated position after the crossover and mutation processes.

To summarize, two different modifications are presented in this subsection to the original GWO. The first modification enforces the parameter $\vec{a}$ to change exponentially and hence increases the number of iterations for exploration. The second modification is based on applying the crossover and mutation processes to the solutions of $\vec{G_1}, \vec{G_2}, \vec{G_3}$ to get the updated position of $\vec{G'_1}, \vec{G'_2}, \vec{G'_3}$. The crossover operator enhances the exploitation process while the mutation operator enhances the exploration process. By merging these modifications, the proposed modified GWO has a higher exploration and exploration capabilities than the original GWO.

B. SFS DIFFUSION PROCESS

To create new particles based on the diffusion procedure of SFS, the Gaussian distribution method is employed for random walk in the DLA growth process. A list of generated walks in the diffusion process according to the best solution $\vec{G_\alpha}$ can be calculated as:

$$\vec{G'_{\alpha_i}} = Gaussian(\mu_{\vec{G_\alpha}}, \sigma) + (\eta \times \vec{G_\alpha} - \eta' \times \vec{P_i}) \tag{13}$$

where $\vec{G'_{\alpha_i}}$ is the updated best solution. Parameters of η and η' are random numbers $\in [0, 1]$. $\vec{G_\alpha}$ and $\vec{P_i}$ are the position of the best point and the ith point in the surrounding group. $\mu_{\vec{G_\alpha}}$ is equal to $\left|\vec{G_\alpha}\right|$ and σ is equal to $\left|\vec{P_i} - \vec{G_\alpha}\right|$ since the number of generation around the best solution decreases. This increases the capability of exploration in the proposed MbGWO based on the diffusion process of the SFS algorithm to get the best solution.

C. BINARY OPTIMIZER

The problem of feature selection is so special because the search space is limited to two binary values 0 and 1. Hence, the traditional continuous version of an optimizer should be modified to work properly for this problem. Here a technique is presented to convert the continuous values of the proposed optimizer (MbGWO-SFS) to binary values, so that it can be used for the feature selection problem. To convert the standard the continuous values to binary values, the following form will be applied as shown in the proposed Algorithm 2.

$$\begin{aligned} \vec{G}_d^{(t+1)} &= \begin{cases} 1 & \text{if } Sigmoid(x) \geq 0.5 \\ 0 & otherwise, \end{cases} \\ Sigmoid(x) &= \frac{1}{1 + \exp^{-10(x-0.5)}}, \\ x &= \frac{\vec{G'_\alpha}\vec{G'_1} + \vec{G_\beta}\vec{G'_2} + \vec{G_\delta}\vec{G'_3}}{\vec{G'_\alpha} + \vec{G_\beta} + \vec{G_\delta}} \end{aligned} \tag{14}$$

where $\vec{G}_d^{(t+1)}$ is the updated binary position of the dimension d at iteration t and $\vec{G'_\alpha}$ is the updated best solution from Eq. 13, $\vec{G_\beta}$ and $\vec{G_\delta}$ are the second and third best individuals. $\vec{G'_1}$, $\vec{G'_2}$, and $\vec{G'_3}$ are the updated positions from Eq. 12. The role of the *Sigmoid* function is to scale the continuous values to be 0 or 1. As shown in Fig. 6, the condition of $Sigmoid(x) \geq$ 0.5 is used to decide whether the value of the dimension will be zero or one.

Algorithm 2 Pseudo Code of the Proposed MbGWO-SFS

```
Initialize MbGWO-SFS population $\vec{G}_i (i = 1, 2, \dots, n)$ with size $n$, maximum iterations number $M_t$, and fitness function $F_n$.
Initialize MbGWO-SFS parameters ($\vec{a}$, $\vec{A}$, $\vec{C}$)
Set t = 1. (initialize counter).
Convert solution to binary [0 or 1].
Calculate the fitness function $F_n$ for each $\vec{G}_i$
Find best, second best and third best individuals as $\vec{G_\alpha}, \vec{G_\beta}, \vec{G_\delta}$
while $t < M_t$ (Termination condition) do
    for ($i = 1 : i < n + 1$) do
        Calculate $\vec{D_\alpha} = |\vec{C_1}.\vec{G_\alpha} - \vec{G}|$
        Calculate $\vec{D_\beta} = |\vec{C_2}.\vec{G_\beta} - \vec{G}|$
        Calculate $\vec{D_\delta} = |\vec{C_3}.\vec{G_\delta} - \vec{G}|$
        Calculate $\vec{G_1} = \vec{G_\alpha} - \vec{A_1}.\vec{D_\alpha}$
        Calculate $\vec{G_2} = \vec{G_\beta} - \vec{A_2}.\vec{D_\beta}$
        Calculate $\vec{G_3} = \vec{G_\delta} - \vec{A_3}.\vec{D_\delta}$
        Apply Crossover Process from Eq. 11 using $\vec{G_1}, \vec{G_2}, \vec{G_3}$
        Apply Mutation Process from Eq. 12 to get updated positions $\vec{G'_1}, \vec{G'_2}, \vec{G'_3}$
    end for
    for ($i = 1 : i < n + 1$) do
        Apply Diffusion Process from Eq. 13 to get $\vec{G'_{\alpha_i}} = Gaussian(\mu_{\vec{G_\alpha}}, \sigma) + (\eta \times \vec{G_\alpha} - \eta' \times \vec{P_i})$
    end for
    Update ($\vec{a}$) by the exponential form of $\vec{a} = 2\left(1 - \left(\frac{t}{M_t}\right)^2\right)$
    Update parameters ($\vec{A}$, $\vec{C}$)
    Convert updated solution to binary using Eq. 14.
    Calculate the fitness function $F_n$ for each $\vec{G}_i$
    Update $\vec{G_\alpha}, \vec{G_\beta}, \vec{G_\delta}$
    Set t = t + 1. (increase counter).
end while
return $\vec{G_\alpha}$
```

D. FITNESS FUNCTION

Fitness function is used to measure the quality of the optimizer solutions. The fitness function depends on two factors: the number of selected features and the classification error rate. The solution is considered to be good if it selected a subset of features that give a lower classification error rate and a lower number of selected features. To evaluate the quality of each solution, the following equation will be used

$$F_n = h_1 E(D) + h_2 \frac{|s|}{|f|} \tag{15}$$

where $E(D)$ is the error rate for the classier, s is the number of selected features, f is the total number of features and

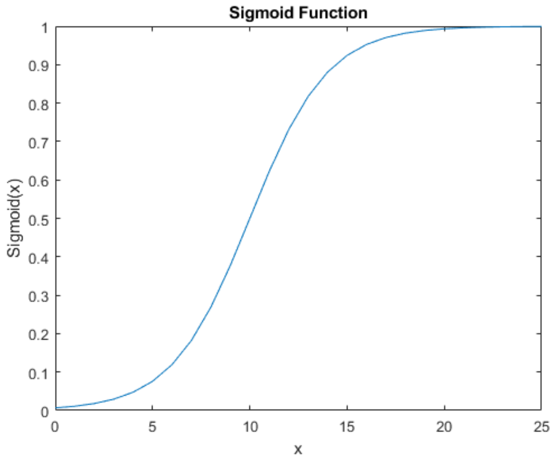


FIGURE 6. Sigmoid function of Eq. 14.

$h_1 \in [0,1], h_2 = 1 - h_1$ manage the importance of the number of the selected feature for population with size n and the classification error rate.

E. COMPUTATIONAL COMPLEXITY ANALYSIS

In this subsection, the proposed MbGWO-SFS algorithm computational complexity will be introduced according to Algorithm 2. Let n be the number of population; M_t be the maximum number of iterations. For each part of the MbGWO-SFS optimizer, the time complexity is defined as follows:

- Initialization of MbGWO-SFS population: $O(1)$.
- Initialization of MbGWO-SFS parameters $\vec{a}$, $\vec{A}$, and $\vec{C}$: $O(1)$.
- Iteration number Initialization: $O(1)$.
- Converting solution to binary: $O(n)$.
- Fitness function calculation for each wolf: $O(n)$.
- Finding first, second, and third best individual: $O(n)$.
- Updating positions for each individual: $O(M_t \times n)$.
- Calculating the diffusion process: $O(M_t \times n)$.
- Updating $\vec{a}$ by the exponential form: $O(M_t)$.
- Updating parameters $\vec{A}$ and $\vec{C}$: $O(M_t)$.
- Converting updated solution to binary: $O(M_t \times n)$.
- Fitness function calculation for each wolf: $O(M_t \times n)$.
- Updating the first, second, and third best individual: $O(M_t \times n)$.
- Increasing the iteration number: $O(M_t)$.
- Producing the best individual: $O(1)$

Based on the previous analysis, the computational complexity for the proposed algorithm is $O(M_t \times n)$. For a problem with m dimension, the proposed algorithm computational complexity will be $O(M_t \times n \times m)$.

V. EVALUATION METRICS

The following metrics are used to evaluate the effectiveness of the proposed MbGWO-SFS algorithm. Assume that: M is the number repetitions of runs of an optimizer for the feature selection problem; g_j^* is the best solution at the run number j; N is the number of tested points.

TABLE 2. Datasets description.

No.	Dataset	# Attributes	# Instances	# Classes
1	Hepatitis	19	155	2
2	Ionosphere	34	351	2
3	Vertebral	6	310	2
4	Seeds	7	210	3
5	Parkinsons	23	197	2
6	Australian	14	690	2
7	Blood	5	748	2
8	Breast_Cancer	10	699	2
9	Diabetes	8	768	2
10	Lymphography	18	148	4
11	Zoo	17	101	7
12	Ring	20	7400	2
13	Titanic	3	2201	2
14	Towonorm	20	7400	2
15	Waveform	21	5000	3
16	Tic-Tac-Toe	9	949	2
17	Mofn	10	1324	2
18	HAR Using Smartphones	561	10299	6
19	ISOLET	617	7797	26

TABLE 3. Proposed algorithm configuration.

Parameter	Value
No of search agents	10
No of iterations	80
Problem dimension	Number of features in the data
Search domain	[0,1]
No. repetitions of runs	20
Mutation ratio	0.1
Crossover	0.9
Maximum diffusion level	1
h_1 Parameter in the fitness function	0.99
h_2 Parameter in the fitness function	0.01

TABLE 4. Compared algorithms configuration.

Algorithm	Parameter (s)	Value (s)
GWO	a	2 to 0
SFS	Maximum diffusion level	1
PSO	Inertia W_{max}, W_{min}	[0.9,0.6]
	Acceleration constants C_1, C_2	[2,2]
GA	Mutation ratio	0.1
	Crossover	0.9
	Selection mechanism	Roulette wheel
SBO	Step size	0.94
	Mutation probability	0.05
	Difference between the upper and lower limit	0.02
WOA	a	2 to 0
	r	[0,1]
MVO	Wormhole existence probability	[0.2,1]
FA	Number of fireflies	10

- **Average Error** is calculated to show the accuracy of the classifier in giving the selected feature set. Average Error can be calculated as

$$AvgError = 1 - \frac{1}{M}\sum_{j=1}^{M}\frac{1}{N}\sum_{i=1}^{N} Match(C_i, L_i) \tag{16}$$

  where C_i is the label of the classifier output for point i, and L_i is the label of the class for point i, and *Match* calculates the matching between two inputs.
- **Average Fitness** is the selected features average size to the total number of features in the dataset (D). Average

TABLE 5. Average error, average select size, and average fitness (Mean) of different optimization techniques in the experiments.

Average Error

Dataset	MbGWO-SFS	MbGWO	bGWO-GA	bGWO-PSO	bSFS	bGWO	bPSO	bGA	bSBO	bWOA	bMVO	bFA
Hepatitis	0.1971	0.2075	0.2196	0.2039	0.2098	0.2108	0.2011	0.1980	0.2157	0.1980	0.1980	0.2049
Ionosphere	0.1344	0.1360	0.1949	0.1485	0.1546	0.1474	0.1752	0.1551	0.1556	0.1603	0.1692	0.1658
Vertebral	0.2063	0.2081	0.2252	0.2175	0.2214	0.2194	0.2248	0.2204	0.2388	0.2165	0.2325	0.2311
Seeds	0.2871	0.2571	0.2586	0.2505	0.2504	0.2971	0.2807	0.2800	0.2586	0.2921	0.2793	0.2900
Parkinsons	0.1355	0.1515	0.1631	0.1538	0.1405	0.1534	0.1515	0.1469	0.1292	0.1423	0.1539	0.1415
Australian	0.1502	0.1817	0.1574	0.1530	0.1506	0.1652	0.1570	0.1552	0.1643	0.1594	0.1607	0.1676
Blood	0.2401	0.2494	0.2386	0.2546	0.2672	0.2492	0.2564	0.2552	0.2490	0.2496	0.2486	0.2432
Breast_Cancer	0.0426	0.0498	0.0489	0.0498	0.0472	0.0485	0.0446	0.0468	0.0472	0.0436	0.0442	0.0464
Diabetes	0.2711	0.2758	0.2586	0.2563	0.3664	0.2684	0.2756	0.2830	0.2589	0.2750	0.2711	0.2744
Lymphography	3.2656	3.2558	3.4188	3.3154	3.3412	3.4286	3.4969	3.1908	3.5564	3.5980	3.7622	3.4908
Zoo	0.1339	0.1462	0.1692	0.1385	0.1538	0.1462	0.1469	0.1523	0.1569	0.1669	0.1454	0.1577
Ring	0.1572	0.1603	0.1673	0.1635	0.1676	0.1591	0.1648	0.1651	0.1657	0.1633	0.1605	0.1630
Titanic	0.2222	0.2301	0.2213	0.2317	0.2220	0.2456	0.2319	0.2261	0.2147	0.2295	0.2244	0.2316
Towonorm	0.0322	0.0332	0.0696	0.0428	0.0602	0.0493	0.0664	0.0690	0.0524	0.0344	0.0460	0.0569
Waveform	0.3926	0.4366	0.4365	0.4286	0.3938	0.4032	0.4262	0.4429	0.4148	0.3960	0.4160	0.4231
Tic-Tac-Toe	0.2571	0.2628	0.2897	0.2746	0.2637	0.2724	0.2694	0.2730	0.3003	0.2564	0.2600	0.2756
Mofn	0.0612	0.0970	0.1324	0.1376	0.1374	0.1163	0.1329	0.1416	0.1079	0.1263	0.1180	0.1353
HAR Using Smartphones	0.4136	0.6652	0.7769	0.8874	0.9978	0.9874	0.8191	0.8879	1.8440	1.5740	1.7454	1.3240
ISOLET	0.6789	0.7541	0.8159	0.6896	0.9988	0.8856	0.9123	0.6895	0.9965	0.9998	0.9681	0.9651

Average Select Size

Dataset	MbGWO-SFS	MbGWO	bGWO-GA	bGWO-PSO	bSFS	bGWO	bPSO	bGA	bSBO	bWOA	bMVO	bFA
Hepatitis	0.3410	0.4512	0.4800	0.4000	0.3599	0.4050	0.5350	0.5000	0.5200	0.6300	0.5050	0.5400
Ionosphere	0.1364	0.2541	0.4242	0.2909	0.3140	0.2667	0.4909	0.3909	0.4303	0.3318	0.4712	0.4879
Vertebral	0.4066	0.4887	0.7000	0.5000	0.5031	0.5000	0.5083	0.5083	0.4330	0.5083	0.5167	0.5083
Seeds	0.5286	0.5612	0.5140	0.5420	0.5140	0.5071	0.7000	0.5500	0.6000	0.6857	0.5286	0.5857
Parkinsons	0.2562	0.3090	0.5000	0.3909	0.2918	0.4114	0.4682	0.4477	0.4900	0.4841	0.5364	0.4705
Australian	0.2721	0.3714	0.5286	0.4571	0.4857	0.4286	0.5321	0.5071	0.5714	0.6786	0.5143	0.5250
Blood	0.6425	0.5436	0.6500	0.6000	0.6107	0.7000	0.6500	0.7750	0.6500	0.7750	0.7375	0.7625
Breast_Cancer	0.4765	0.5214	0.6500	0.5000	0.5250	0.5250	0.5938	0.5813	0.6500	0.6375	0.5938	0.6375
Diabetes	0.3433	0.4250	0.5250	0.5250	0.4751	0.5125	0.6063	0.5625	0.5250	0.6375	0.5875	0.5750
Lymphography	0.5200	0.5211	0.5333	0.2778	0.5333	0.3750	0.4833	0.4306	0.5333	0.5083	0.4778	0.4583
Zoo	0.2591	0.3909	0.4818	0.3818	0.4545	0.3614	0.4932	0.4591	0.4909	0.5000	0.4750	0.4909
Ring	0.3000	0.3211	0.3200	0.3400	0.3321	0.3200	0.3600	0.3350	0.3400	0.3250	0.3350	0.3550
Titanic	0.8000	0.8311	0.8000	0.8667	0.8567	0.8000	0.8167	0.8500	0.8140	0.8500	0.8833	0.8833
Towonorm	0.6400	0.7300	0.6700	0.8500	0.8100	0.8525	0.6900	0.8700	0.7900	0.9775	0.8500	0.7475
Waveform	0.4452	0.6000	0.6857	0.5143	0.5195	0.5286	0.5810	0.6524	0.6000	0.8952	0.6405	0.6071
Tic-Tac-Toe	0.4611	0.4734	0.5897	0.4746	0.4677	0.5333	0.6111	0.6111	0.4703	0.7500	0.6111	0.6333
Mofn	0.1817	0.2070	0.4324	0.1976	0.2383	0.6000	0.6650	0.6850	0.4079	0.8600	0.6450	0.6750
HAR Using Smartphones	0.5671	0.6987	0.7121	0.7752	0.9012	0.8845	0.8794	0.7714	0.9550	0.9035	0.9165	0.9561
ISOLET	0.6987	0.7713	0.7865	0.7785	0.9952	0.8771	0.8141	0.7952	0.9814	0.9651	0.9356	0.9332

Average Fitness (Mean)

Dataset	MbGWO-SFS	MbGWO	bGWO-GA	bGWO-PSO	bSFS	bGWO	bPSO	bGA	bSBO	bWOA	bMVO	bFA
Hepatitis	0.2248	0.2265	0.2190	0.2430	0.2610	0.2607	0.2471	0.2481	0.2550	0.2481	0.2481	0.2549
Ionosphere	0.1465	0.1781	0.1940	0.1480	0.1543	0.1814	0.2089	0.1890	0.1550	0.1941	0.2030	0.1996
Vertebral	0.3426	0.2230	0.3250	0.2170	0.2443	0.3906	0.3958	0.3915	0.3380	0.3877	0.4035	0.4021
Seeds	0.3643	0.3877	0.4480	0.4540	0.4510	0.3942	0.3779	0.3772	0.4280	0.3892	0.3765	0.3871
Parkinsons	0.1704	0.1713	0.1630	0.1530	0.1633	0.1628	0.1796	0.1750	0.1890	0.1704	0.1819	0.1697
Australian	0.3130	0.1817	0.1574	0.1530	0.1406	0.3279	0.3197	0.3180	0.1643	0.3220	0.3233	0.3302
Blood	0.8519	0.2494	0.2386	0.2546	0.2677	0.8717	0.8789	0.8777	0.2490	0.8721	0.8711	0.8638
Breast_Cancer	0.3234	0.3468	0.3489	0.3498	0.3466	0.3393	0.3354	0.3376	0.5472	0.3344	0.3350	0.3371
Diabetes	0.5767	0.5858	0.5859	0.5863	0.5877	0.5857	0.5928	0.6002	0.5934	0.5923	0.5884	0.5917
Lymphography	3.0736	3.4898	5.2490	4.3878	4.6776	3.4221	3.4898	3.1867	4.7388	3.5898	3.7524	3.4837
Zoo	0.1621	0.1662	0.1692	0.1785	0.1838	0.1742	0.1750	0.1803	0.1769	0.1948	0.1735	0.1857
Ring	1.3892	0.1403	0.1673	0.1635	0.1676	1.3911	1.3967	1.3970	0.1657	1.3952	1.3924	1.3949
Titanic	2.6666	2.7192	2.7213	2.7173	2.7902	2.6898	2.6763	2.6705	2.9473	2.6738	2.6688	2.6759
Towonorm	1.1758	1.3483	1.6696	1.4428	1.6101	1.2823	1.2992	1.3018	1.5524	1.2676	1.2791	1.2899
Waveform	1.1825	1.3236	0.4365	1.4286	0.4353	1.1930	1.2158	1.2323	1.4148	1.1858	1.2056	1.2127
Tic-Tac-Toe	0.6100	0.6438	0.6897	0.7900	0.2677	0.6253	0.6223	0.6259	0.6203	0.6094	0.6130	0.6284
Mofn	0.5036	0.5214	0.5324	0.5179	0.5361	0.5572	0.5736	0.5822	0.5794	0.5670	0.5589	0.5759
HAR Using Smartphones	0.5443	0.7654	0.7894	0.7891	0.9112	0.8567	0.8556	0.7741	0.9745	0.9135	0.9348	0.9225
ISOLET	0.6771	0.7765	0.8001	0.7756	0.9321	0.8443	0.8812	0.8012	0.9856	0.9456	0.9312	0.9148

Fitness is calculated from the following equation

$$AvgSelectSize = \frac{1}{M} \sum_{j=1}^{M} \frac{size(g_j^*)}{D} \tag{17}$$

where $size(g_j^*)$ is the size of the vector g_j^*.

- **Mean** is the average of the solutions output from running an optimizer for several times M. It can be calculated as

$$Mean = \frac{1}{M} \sum_{j=1}^{M} g_j^* \tag{18}$$

- **Best Fitness** is the minimum fitness function of an optimizer running for several times M. Best Fitness can be calculated as

$$BestF_n = Min_{j=1}^{M} g_j^* \tag{19}$$

- **Worst Fitness** is the worst solution found by an optimizer running for several times M. Worst Fitness can be calculated as

$$WorstF_n = Max_{j=1}^{M} g_j^* \tag{20}$$

TABLE 6. Best fitness, worst fitness, and standard deviation fitness of different optimization techniques in the experiments.

Best Fitness

Dataset	MbGWO-SFS	MbGWO	bGWO-GA	bGWO-PSO	bSFS	bGWO	bPSO	bGA	bSBO	bWOA	bMVO	bFA
Hepatitis	0.1297	0.1773	0.2073	0.2267	0.2103	0.1491	0.1491	0.1491	0.2073	0.1879	0.2073	0.1879
Ionosphere	0.0947	0.1448	0.1793	0.1370	0.1014	0.1116	0.1455	0.1285	0.1285	0.1201	0.1032	0.0947
Vertebral	0.3079	0.3163	0.3463	0.3752	0.3656	0.3271	0.3175	0.3175	0.3752	0.3175	0.3367	0.3367
Seeds	0.1000	0.1667	0.2697	0.2414	0.2354	0.1849	0.1566	0.1849	0.2273	0.1566	0.2273	0.2414
Parkinsons	0.0605	0.1666	0.1514	0.1057	0.0752	0.0752	0.1209	0.0905	0.0905	0.0905	0.0905	0.0752
Australian	0.2762	0.2934	0.3063	0.2977	0.3010	0.2891	0.2848	0.2891	0.3063	0.2805	0.2977	0.2934
Blood	0.8318	0.8567	0.8278	0.8357	0.8676	0.8318	0.8397	0.8278	0.8378	0.8318	0.8397	0.8278
Breast_Cancer	0.3087	0.3150	0.3295	0.3295	0.3172	0.3125	0.3167	0.3167	0.3252	0.3083	0.3125	0.3040
Diabetes	0.5289	0.5542	0.5520	0.5636	0.5559	0.5482	0.5559	0.5443	0.5482	0.5366	0.5559	0.5443
Lymphography	1.9674	1.4219	3.8868	2.8563	3.1221	1.9068	1.9068	1.7451	3.8261	1.4825	1.6643	1.9674
Zoo	0.0905	0.1342	0.1514	0.1209	0.1514	0.0905	0.0905	0.1057	0.1057	0.1057	0.0905	0.0905
Ring	1.3692	1.3849	1.3909	1.3849	1.3917	1.3704	1.3841	1.3857	1.3844	1.3820	1.3804	1.3704
Titanic	2.6439	2.6366	2.6358	2.6358	2.6520	2.6439	2.6439	2.6439	2.6458	2.6439	2.6371	2.6439
Towonorm	1.1932	1.2118	1.2813	1.2672	1.2849	1.2708	1.2732	1.2837	1.2805	1.2616	1.2640	1.2785
Waveform	1.1171	1.2079	1.1890	1.2015	1.1718	1.1545	1.1486	1.1735	1.1640	1.1278	1.1676	1.1408
Tic-Tac-Toe	0.5635	0.5887	0.5821	0.5821	0.5897	0.5635	0.5666	0.5759	0.5852	0.5790	0.5635	0.5728
Mofn	0.4510	0.4679	0.5632	0.4779	0.5165	0.4847	0.4981	0.5138	0.4622	0.5228	0.4847	0.5385
HAR Using Smartphones	0.5001	0.7066	0.7055	0.7623	0.9011	0.8332	0.8143	0.7619	0.9007	0.9104	0.8807	0.9115
ISOLET	0.6103	0.7334	0.7779	0.7111	0.8896	0.8045	0.8456	0.7895	0.7996	0.9123	0.9127	0.8893

Worst Fitness

Dataset	MbGWO-SFS	MbGWO	bGWO-GA	bGWO-PSO	bSFS	bGWO	bPSO	bGA	bSBO	bWOA	bMVO	bFA
Hepatitis	0.3242	0.3432	0.3238	0.3044	0.2849	0.3432	0.3626	0.3238	0.3238	0.3626	0.3238	0.3820
Ionosphere	0.2459	0.2639	0.2893	0.2047	0.2876	0.2555	0.2639	0.3062	0.2301	0.2808	0.2470	0.2978
Vertebral	0.4497	0.4713	0.4425	0.3944	0.4236	0.4329	0.4617	0.5001	0.5001	0.4617	0.5001	0.6059
Seeds	0.5099	0.4194	0.4394	0.4394	0.4394	0.6516	0.6091	0.6233	0.4394	0.6091	0.5384	0.5526
Parkinsons	0.2412	0.2580	0.2732	0.2428	0.2580	0.2276	0.2732	0.2428	0.2428	0.2580	0.2580	0.2428
Australian	0.3469	0.4355	0.3365	0.3408	0.3192	0.3967	0.3795	0.3537	0.3365	0.3623	0.3451	0.4957
Blood	0.9351	0.9192	0.9192	0.9192	0.9630	0.9351	0.9351	0.9351	0.9192	0.9152	0.9351	0.8954
Breast_Cancer	0.3601	0.3677	0.3465	0.3592	0.3592	0.3762	0.3550	0.3677	0.3550	0.3507	0.3550	0.3635
Diabetes	0.6279	0.6255	0.5946	0.5791	0.5984	0.6565	0.6216	0.7067	0.5868	0.6410	0.6448	0.6332
Lymphography	5.2825	5.6274	7.3012	5.0384	6.4404	5.2404	5.9274	5.1192	6.6547	5.4627	5.4829	5.0990
Zoo	0.2289	0.2680	0.2785	0.2275	0.2428	0.2732	0.2885	0.2885	0.2580	0.3189	0.2428	0.2428
Ring	1.4017	1.3879	1.4057	1.4105	1.4158	1.4037	1.4150	1.4122	1.4130	1.4110	1.4081	1.4162
Titanic	2.4857	2.6519	2.6898	2.6776	2.7614	3.1085	2.7330	2.7317	2.6871	2.7803	2.7181	2.8613
Towonorm	1.1657	1.2957	1.3130	1.2829	1.3042	1.2985	1.3403	1.3403	1.2925	1.2877	1.2957	1.3038
Waveform	1.1886	1.2448	1.2448	1.2502	1.2769	1.2906	1.2686	1.2918	1.2597	1.2110	1.2555	1.2769
Tic-Tac-Toe	0.6590	0.6876	0.6907	0.6845	0.6597	0.6938	0.7497	0.6690	0.7062	0.6597	0.6814	0.6814
Mofn	0.5802	0.6008	0.5812	0.5722	0.6528	0.6283	0.6396	0.6283	0.5834	0.6059	0.6283	0.6014
HAR Using Smartphones	0.5676	0.7881	0.8004	0.8678	0.9315	0.8664	0.8776	0.8100	0.8456	0.9877	0.9455	0.9324
ISOLET	0.6799	0.8146	0.8423	0.8133	0.9467	0.8611	0.8899	0.8333	1.1240	0.9645	0.9544	0.9371

Standard Deviation Fitness

Dataset	MbGWO-SFS	MbGWO	bGWO-GA	bGWO-PSO	bSFS	bGWO	bPSO	bGA	bSBO	bWOA	bMVO	bFA
Hepatitis	0.0329	0.0424	0.0443	0.0355	0.0519	0.0519	0.0596	0.0503	0.0475	0.0462	0.0398	0.0506
Ionosphere	0.0330	0.0377	0.0465	0.0383	0.0746	0.0425	0.0333	0.0475	0.0376	0.0423	0.0397	0.0480
Vertebral	0.0260	0.0376	0.0356	0.0286	0.0485	0.0288	0.0400	0.0450	0.0511	0.0377	0.0404	0.0640
Seeds	0.0930	0.0951	0.0859	0.0815	0.0891	0.1162	0.1038	0.1057	0.0994	0.1072	0.0794	0.0831
Parkinsons	0.0319	0.0354	0.0477	0.0517	0.0765	0.0413	0.0429	0.0375	0.0576	0.0525	0.0528	0.0444
Australian	0.0154	0.0341	0.0193	0.0189	0.0268	0.0320	0.0219	0.0179	0.0167	0.0236	0.0154	0.0455
Blood	0.0203	0.0226	0.0361	0.0338	0.0446	0.0251	0.0285	0.0314	0.0403	0.0248	0.0250	0.0209
Breast_Cancer	0.0127	0.0107	0.0064	0.0126	0.0331	0.0142	0.0093	0.0133	0.0116	0.0111	0.0115	0.0132
Diabetes	0.0158	0.0316	0.0190	0.0175	0.0368	0.0302	0.0170	0.0351	0.0165	0.0225	0.0267	0.0255
Lymphography	0.8408	1.7625	1.3669	0.8804	1.0943	0.9014	1.1364	0.9380	1.1322	0.8472	0.9721	0.9014
Zoo	0.0407	0.0446	0.0560	0.0444	0.0588	0.0454	0.0495	0.0476	0.0632	0.0558	0.0465	0.0454
Ring	0.0067	0.0067	0.0059	0.0099	0.0102	0.0084	0.0078	0.0063	0.0108	0.0094	0.0078	0.0117
Titanic	0.0148	0.0448	0.0229	0.0154	0.0446	0.0997	0.0270	0.0215	0.0187	0.0321	0.0193	0.0461
Towonorm	0.0060	0.0107	0.0126	0.0065	0.0118	0.0087	0.0163	0.0147	0.0084	0.0066	0.0083	0.0075
Waveform	0.0212	0.0314	0.0230	0.0206	0.0418	0.0316	0.0294	0.0337	0.0425	0.0255	0.0313	0.0334
Tic-Tac-Toe	0.0204	0.0221	0.0419	0.0396	0.0445	0.0345	0.0237	0.0299	0.0434	0.0294	0.0204	0.0296
Mofn	0.0299	0.0691	0.0068	0.0408	0.0311	0.0384	0.0462	0.0166	0.0491	0.0335	0.0211	0.0396
HAR Using Smartphones	0.0169	0.0222	0.3011	0.0299	0.0512	0.0334	0.0312	0.0301	0.0501	0.0544	0.0456	0.0445
ISOLET	0.0233	0.0301	0.3240	0.0378	0.0601	0.0414	0.0399	0.0302	0.0523	0.0623	0.0512	0.0524

- **Standard Deviation** (SD) is the obtained best solutions variation which can be found by running an optimizer several times M. SD is an important indicator of the stability and robustness of an optimizer. An optimizer's ability to converge to the same solution is indicated by a smaller SD. SD can be calculated as

$$SD = \sqrt{\frac{1}{M-1}\sum(g_j^* - Mean)^2} \tag{21}$$

where $Mean$ is the average defined in equation 18.

VI. EXPERIMENTAL RESULTS AND DISCUSSION

To evaluate the quality and effectiveness of the proposed MbGWO-SFS algorithm, nineteen datasets from the repository of the UCI machine learning are tested. The datasets are selected with various number of attributes, instances, and classed to represent different kind of issues that the proposed algorithm can be tested on, with two datasets have more than 500 attributes. Table 2 shows the description of the UCI datasets that are used in the experiments. Each dataset is divided into three randomly equal-size parts of training, validation, and testing. The training part is used to train the

TABLE 7. Processing time for different optimization techniques in the experiments.

	Processing Time										
Dataset	MbGWO-SFS	MbGWO	bGWO-GA	bGWO-PSO	bSFS	bGWO	bPSO	bGA	bSBO	bWOA	bMVO	bFA
Hepatitis	2.734	3.150	3.510	2.970	4.460	3.378	3.060	3.334	3.790	2.867	3.321	4.088
Ionosphere	3.152	3.690	4.890	4.220	4.410	5.083	4.655	4.608	4.620	3.988	4.595	4.672
Vertebral	2.404	3.101	3.880	3.840	3.770	3.600	3.679	3.079	3.560	2.415	3.372	3.928
Seeds	2.405	3.020	3.050	3.200	3.430	3.071	3.916	3.166	3.150	2.507	3.684	3.008
Parkinsons	2.408	3.141	3.410	2.800	3.640	3.351	3.130	2.602	4.250	3.083	3.254	3.342
Australian	4.473	4.800	5.900	5.845	6.990	5.742	5.656	5.436	6.380	5.828	5.64	5.847
Blood	4.483	4.900	5.640	4.789	5.310	4.206	2.684	5.618	4.560	4.602	5.738	6.052
Breast_Cancer	4.287	5.110	5.670	4.789	5.230	5.594	5.906	5.728	5.970	5.278	5.950	5.490
Diabetes	4.448	5.330	5.440	5.130	5.860	4.769	5.454	5.790	6.350	6.327	6.657	7.312
Lymphography	3.067	3.910	4.570	3.990	5.789	3.612	4.759	3.599	4.320	3.269	3.785	4.335
Zoo	3.098	4.880	4.890	4.550	5.650	4.344	3.145	3.709	5.880	4.253	3.601	3.599
Ring	47.112	55.120	78.690	111.470	131.350	101.007	75.948	49.956	83.590	68.716	76.058	76.924
Titanic	7.047	11.600	27.365	9.850	14.800	11.821	10.383	7.558	89.360	11.236	13.286	12.849
Towonorm	95.321	99.870	135.698	132.330	145.780	127.760	137.247	95.268	145.740	722.644	157.921	164.873
Waveform	28.402	66.600	45.980	58.630	79.890	48.051	54.455	32.995	54.780	64.347	49.838	52.057
Tic-Tac-Toe	5.454	7.690	7.660	7.120	5.990	6.946	6.934	6.642	8.150	6.437	6.700	6.967
Mofn	7.151	8.150	8.660	8.780	9.470	8.161	7.643	7.169	8.880	7.782	8.132	7.717
HAR Using Smartphones	319.23	415.990	459.330	435.610	599.360	455.800	466.580	488.440	607.450	623.590	612.880	599.880
ISOLET	425.550	445.610	488.700	476.900	613.8	488.990	455.300	489.100	666.350	729.750	635.770	691.870

KNN classifier during the learning phase. The validation is used to test when calculating the fitness function for a specific solution and the testing part is used to evaluate the proposed model efficiency. Table 3 shows the configuration of the proposed algorithm in the experiments. Each optimizer is run 20 times for 80 iterations and the number of search agents is set to 10. For the KNN classifier, the number of k-neighbors is 5 and the value of the k-fold cross-validation is set to 10. The parameters of h_1 and h_2 in the fitness function are assigned to 0.99 and 0.01, respectively. Table 4 shows the configuration of the compared algorithms in the experiments.

The proposed (MbGWO-SFS) algorithm is compared in the experiments to different optimization algorithms with single and combined mechanisms. The single mechanisms are the binary versions of the techniques of GWO [1] (bGWO), SFS [23] (bSFS), PSO [24] (bPSO), SBO [25] (bSBO), WOA [26] (bWOA), MVO [27] (bMVO), FA [28] (bFA), and GA [29] (bGA), where b indicated binary output of the algorithm. The binary version uses the Sigmoid function with x represents the algorithm output. The combined mechanisms such as a hybrid of PSO and GWO (bGWO-PSO) [21], a hybrid of GA and GWO (bGWO-GA), and the MbGWO algorithm without applying the diffusion processes of the SFS algorithm are also applied to the tested datasets to clarify the effectiveness of the proposed algorithm these three mechanisms are introduced. Seven different experiments are conducted to evaluate the performance of the proposed MbGWO-SFS optimizer. The performance metrics of average error, average select size, average fitness (Mean), best fitness, worst fitness, standard deviation fitness, and the processing time are evaluated for different optimization techniques during the experiments.

The results of the average error, the average select size, and the average fitness (Mean) for the optimization techniques are shown in Table 5. The lower error indicates that the optimizer has selected the proper set of features that can train the classifier and produce a lower error on the hidden test data. Note that, the lowest error is achieved by the proposed (MbGWO-SFS) algorithm for the Hepatitis, Ionosphere, Vertebral, Australian, Breast-Cancer, Zoo, Ring, Towonorm, Waveform, Mofn, HAR Using Smartphones, and ISOLET datasets which indicates the high exploration of the search space. The bGWO-PSO algorithm achieved lower error for Seeds and Diabetes datasets, however, bSBO achieved lower error for Parkinsons and Titanic datasets. The bGWO-GA,

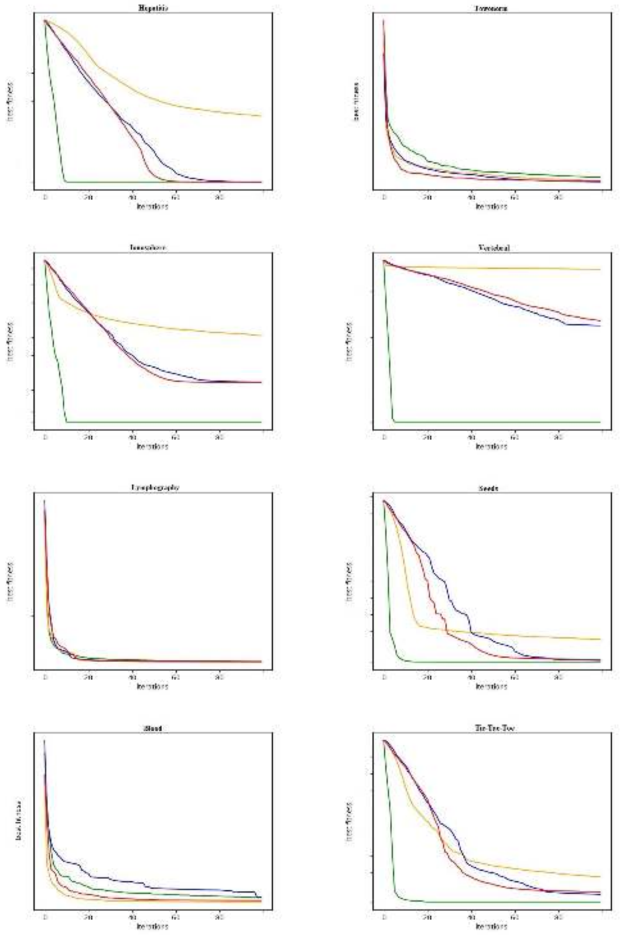
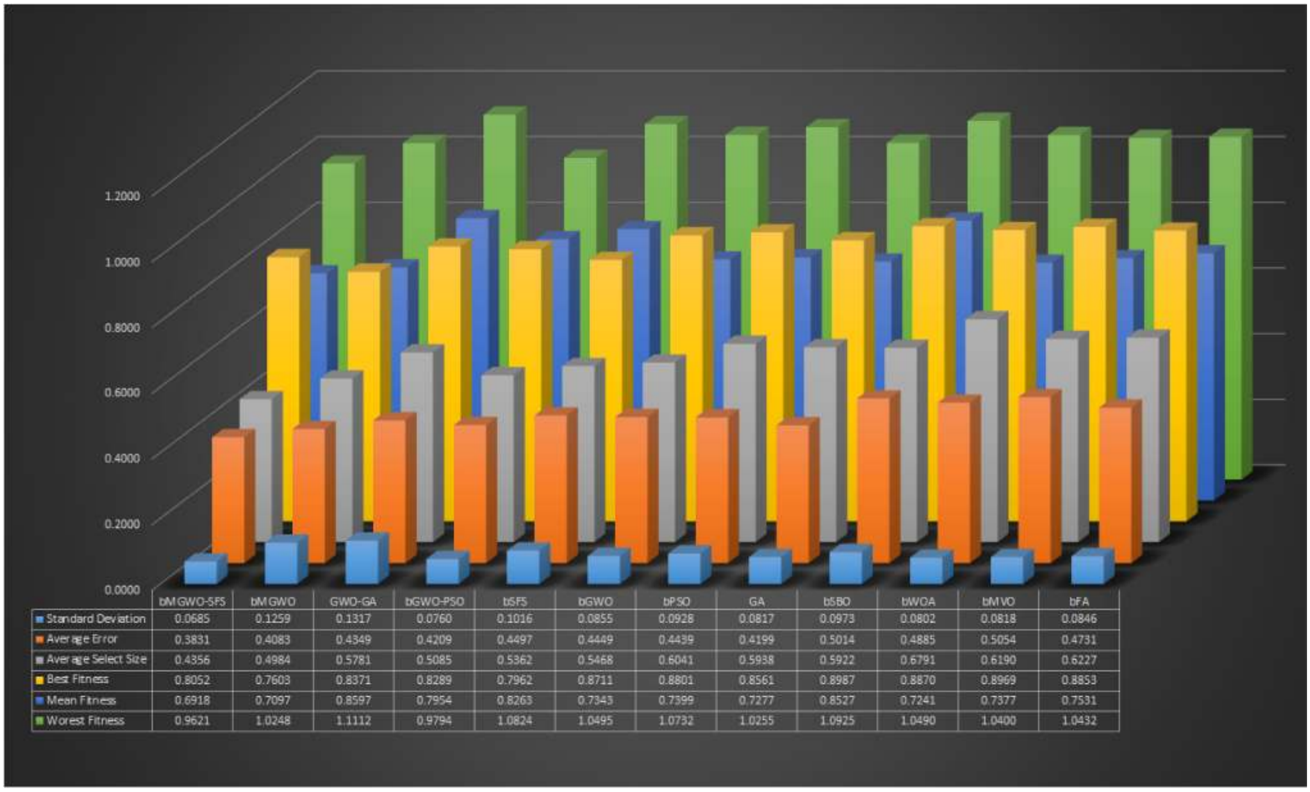


FIGURE 7. Proposed (MbGWO-SFS) convergence curves compared to other techniques; green, yellow, blue, and red lines indicates MbGWO-SFS, bPSO, bGWO, bGA algorithms, respectively.



	bMGWO-SFS	bMGWO	GWO-GA	bGWO-PSO	bSFS	bGWO	bPSO	GA	bSBO	bWOA	bMVO	bFA
Standard Deviation	0.0685	0.1259	0.1317	0.0760	0.1016	0.0855	0.0928	0.0817	0.0973	0.0802	0.0818	0.0846
Average Error	0.3831	0.4083	0.4349	0.4209	0.4497	0.4449	0.4439	0.4199	0.5014	0.4885	0.5054	0.4731
Average Select Size	0.4356	0.4984	0.5781	0.5085	0.5362	0.5468	0.6041	0.5938	0.5922	0.6791	0.6190	0.6227
Best Fitness	0.8052	0.7603	0.8371	0.8289	0.7962	0.8711	0.8801	0.8561	0.8987	0.8870	0.8969	0.8853
Mean Fitness	0.6918	0.7097	0.8597	0.7954	0.8263	0.7343	0.7399	0.7277	0.8527	0.7241	0.7377	0.7531
Worest Fitness	0.9621	1.0248	1.1112	0.9794	1.0824	1.0495	1.0732	1.0255	1.0925	1.0490	1.0400	1.0432

FIGURE 8. **Averaged error, select size, fitness (mean), best fitness, worst fitness, and standard deviation fitness acquired over all the datasets.**

bGA, and bWOA showed lower error for Blood, Lymphography, and Tic-Tac-Toe datasets. The proposed algorithm uses the crossover operator to move toward the optimal solution, which contains the optimal subset of features, that minimizes the error.

The average selected features from Table 5 shows the effectiveness of the proposed algorithm. Although, choosing a lower number of features indicates that the optimizer performs feature selection, maintaining lower error is important. Thus, the fitness function assigns a higher weight for the classification error and encourages the optimizer to choose the lower number of features. The MbGWO-SFS algorithm can find the least number of channels for most of the datasets and can get the lower classification for them. However, MbGWO-SFS chooses a higher number of features for (Seeds and Lymphography) datasets and it maintains the smallest error for these datasets. The bGWO and bGWO-PSO algorithms show better results for Seeds and Lymphography datasets.

Table 5 also shows that the proposed algorithm can find the lowest fitness value for all datasets except for Vertebral, Parkinsons, Blood, and Tic-Tac-Toe datasets which are better achieved by bGWO-PSO, bGWO-GA, and bWOA. This means that MbGWO-SFS can select the optimal subset of features that give the lowest classification error. The reason for this high performance is the cooperative nature of the individuals of the GWO which utilizes the proposed modification of $\overrightarrow{a}$ parameter and the mutation operator to highly explore the search space for different solutions. Moreover, the proposed crossover and the diffusion procedure of the SFS algorithm enhances the exploitation process.

The results of the best fitness, the worst fitness, and the standard deviation fitness of different optimization techniques are shown in Table 6. From the table, the proposed MbGWO-SFS algorithm can find the best fitness compared to other optimization techniques throughout runs. However, bGWO-GA, MbGWO, and bGWO-PSO algorithms achieved better results for Blood, Lymphography, and Titanic datasets. On the other hand, MbGWO-SFS can not find the worst fitness that proves the capability of the proposed algorithm to find the optimal subset of features compared to other techniques in any of the tested datasets even in the higher dimensions datasets of HAR Using Smartphones and ISOLET. Table 6 also outlines the standard deviation for statistical results. The proposed MbGWO-SFS algorithm has the lowest standard deviation compared to other algorithms that prove the stability and robustness of the proposed algorithm in most of the datasets. The Seeds, Breast-Cancer, Ring, Waveform, Mofn datasets get better standard deviation by other optimizations techniques including bMVO, bGWO-GA, and bGWO-PSO algorithms.

The last experiment investigates the processing time that is required by different optimization techniques as shown

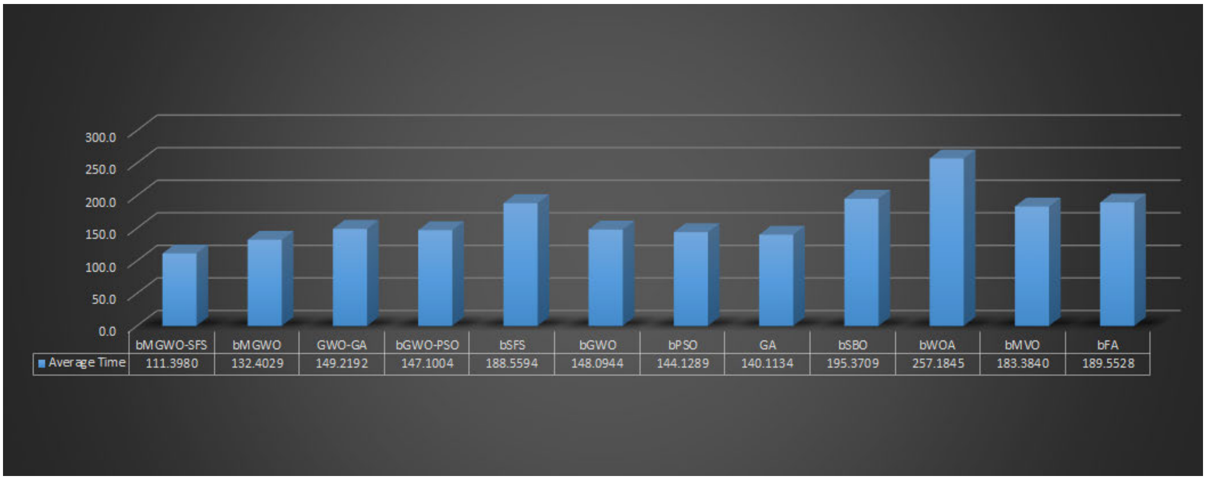


FIGURE 9. Averaged processing time over all the datasets using the selected features from the different optimization techniques.

TABLE 8. p-values of MbGWO-SFS in comparison to other algorithms using Wilcoxon's rank-sum ($p > 0.05$ are underlined).

Dataset	MbGWO	bGWO-GA	bGWO-PSO	bSFS	bGWO	bPSO	bGA	bSBO	bWOA	bMVO	bFA
Hepatitis	1.21E-05	1.11E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	7.88E-02	1.21E-05	1.80E-01	6.26E-02	1.21E-05
Ionosphere	7.61E-02	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05
Vertebral	7.81E-02	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05
Seeds	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05
Parkinsons	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05
Australian	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.11E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05
Blood	1.21E-05	7.84E-02	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05
Breast_Cancer	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.11E-05	1.21E-05	6.33E-02	1.21E-05	1.21E-05
Diabetes	1.21E-05	1.11E-05	1.31E-05	1.21E-05	1.21E-05	1.31E-05	1.21E-05	1.21E-05	1.21E-05	6.61E-02	1.21E-05
Lymphography	1.21E-05	1.11E-05	1.31E-05	1.21E-05	1.21E-05	1.31E-05	6.05E-02	1.21E-05	1.21E-05	1.21E-05	1.21E-05
Zoo	1.21E-05	1.11E-05	1.21E-05	1.21E-05	1.21E-05	1.31E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05
Ring	1.21E-05	1.11E-05	1.21E-05	1.21E-05	1.31E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05
Titanic	1.21E-05	9.91E-02	1.21E-05	8.89E-02	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05
Towonorm	5.87E-02	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05
Waveform	1.21E-05	1.21E-05	1.21E-05	8.72E-02	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05
Tic-Tac-Toe	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	8.86E-02	1.21E-05	1.21E-05
Mofn	1.21E-05	1.11E-05	1.11E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05
HAR Using Smartphones	1.21E-05	1.11E-05	1.11E-05	1.11E-05	1.11E-05	1.11E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05	1.21E-05
ISOLET	1.11E-05	1.11E-05	9.15E-02	1.21E-05	1.21E-05	1.21E-05	7.32E-02	1.21E-05	1.21E-05	1.21E-05	1.21E-05

in Table 7. As a preprocessing step for the proposed algorithm, the problem of class imbalance that may occur in some datasets is solved by applying the LSH-SMOTE [5] algorithm to improve the processing time. The lower processing time in most cases indicates that the optimizer finds the optimal subset of features in less time. The proposed optimizer has competitive results compared to other algorithms for the higher dimensions datasets of HAR Using Smartphones and ISOLET. The bPSO and bGA achieved better processing time for the Blood and Towonorm datasets. The faster convergence time as shown in Fig. 7 proves the high exploitation capability of the proposed optimizer and the ability to avoid local optima. This proves the robustness and reliability of the MbGWO-SFS algorithm in finding the optimal subset of features in a reasonable amount of time.

As average values for all the tested datasets according to different optimization techniques, Figure 8 outlines the averaged error, the average size, the average mean, the best fitness, the worst fitness, and the standard deviation fitness overall the nineteen datasets. This figure shows the stability of the proposed algorithm compared to other algorithms. Figure 9 shows the performance of test data averaged processing time overall the datasets using the selected features from the different optimization techniques. Note from these figures that, the proposed MbGWO-SFS algorithm is performing better than most of the other optimization techniques.

To summarize the results of seven different experiments, the proposed MbGWO-SFS algorithm outperforms other optimization techniques in most datasets. The proposed algorithm achieved the average standard deviation of (0.0685), the average error of (0.3831), the average select size of (0.4356), the best fitness of (0.8052), the mean fitness of (0.6918), the worst fitness of (0.9621), and the average processing time of (111.3980) acquired over all datasets. This is due to the high exploration and exploitation of the MbGWO-SFS which allows it to find the best subset of features. This confirms the robustness and reliability in the classification tasks for various datasets in finding the optimal subset of features.

A. WILCOXON'S RANK-SUM

The test of Wilcoxon's rank-sum is done here to get the p-values of the proposed MbGWO-SFS algorithm in comparison to other meta-heuristic algorithms. This test helps to determine if the results of the proposed algorithm and other algorithms have a significant difference or not. If the p-value < 0.05, it means that the proposed algorithm results are significantly different from the compared algorithms. Otherwise, a p-value > 0.05 means that the results have no significant difference. Table 8 shows the results of p-value where the worst values that are greater than 0.05 are underlined. Note from the table that, the p-values obtained between the proposed algorithm and other algorithms using this test are smaller than 0.05. This shows the superiority of the MbGWO-SFS algorithm and that the algorithm is statistically significant.

VII. CONCLUSION AND FUTURE DIRECTIONS

This paper proposed a modified binary GWO algorithm based on a stochastic fractal search technique (MbGWO-SFS) that is used with the KNN classifier to select the optimal subset of features for different problems by achieving the exploration and exploitation balance. The modified GWO was developed first by applying an exponential form of parameter $\overrightarrow{a}$ of the original GWO to increase the search space for exploitation and the crossover/mutation operations to increase the diversity of the population for exploitation. The SFS technique diffusion process was then applied using the Gaussian distribution method for a random walk for the best solution of the modified GWO. Finally, the continuous values of the proposed algorithm were converted into binary ones by a Sigmoid sunction to use it for the problem of feature selection. The stability and robustness of the proposed MbGWO-SFS algorithm were investigated in the experiments using nineteen datasets from the UCI machine learning repository. The results were compared to the optimization techniques of MbGWO, bGWO, bSFS, bPSO, the hybrid of PSO and GWO (bGWO-PSO), bGA, the hybrid of GA and GWO (bGWO-GA), bSBO, bWOA, bMVO, and bFA. The results showed the superiority of the proposed MbGWO-SFS algorithm. In the future work, the proposed algorithm will be tested for continuous problems, constrained engineering problems, and another binary problem such as EEG problem and also binary problems with more than 1000 attributes. The authors will try to improve continuous MbGWO-SFS and validate the performance of the proposed algorithm at CEC2017 or CEC2019.

REFERENCES

[1] 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.

[2] A. Ibrahim, A. Tharwat, T. Gaber, and A. E. Hassanien, "Optimized superpixel and AdaBoost classifier for human thermal face recognition," *Signal, Image Video Process.*, vol. 12, no. 4, pp. 711–719, May 2018.

[3] M. Tubishat, M. A. Abushariah, N. Idris, and I. Aljarah, "Improved whale optimization algorithm for feature selection in Arabic sentiment analysis," *Appl. Intell.*, vol. 49, no. 5, pp. 1688–1707, May 2019, doi: 10.1007/s10489-018-1334-8.

[4] C.-H. Yang, L.-Y. Chuang, and C. H. Yang, "IG-GA: A hybrid filter/wrapper method for feature selection of microarray data," *J. Med. Biol. Eng.*, vol. 30, no. 1, pp. 23–28, 2010. [Online]. Available: http://www.jmbe.org.tw/files/507/public/507-1737-1-PB.pdf

[5] E. M. Hassib, A. I. El-Desouky, E. M. El-Kenawy, and S. M. El-Ghamrawy, "An imbalanced big data mining framework for improving optimization algorithms performance," *IEEE Access*, vol. 7, pp. 170774–170795, 2019.

[6] A. Ibrahim, A. Ahmed, S. Hussein, and A. E. Hassanien, "Fish image segmentation using salp swarm algorithm," in *Proc. Int. Conf. Adv. Mach. Learn. Technol. Appl. (AMLTA)*, A. E. Hassanien, M. F. Tolba, M. Elhoseny, and M. Mostafa, Eds. Cham, Switzerland: Springer, 2018, pp. 42–51.

[7] M. Abdel-Basset, D. El-Shahat, I. El-henawy, S. Mirjalili, and V. H. C. de Albuquerque, "A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection," *Expert Syst. Appl.*, vol. 139, Jan. 2020, Art. no. 112824. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417419305263

[8] 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. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705118302132

[9] M. Mafarja, I. Aljarah, A. A. Heidari, A. I. Hammouri, H. Faris, A. M. Al-Zoubi, and S. Mirjalili, "Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems," *Knowl.-Based Syst.*, vol. 145, pp. 25–45, Apr. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705117306159

[10] I. Aljarah, H. Faris, and S. Mirjalili, "Optimizing connection weights in neural networks using the whale optimization algorithm," *Soft Comput.*, vol. 22, no. 1, pp. 1–15, Jan. 2018.

[11] E. M. Hassib, A. I. El-Desouky, L. M. Labib, and E.-S.-M. El-kenawy, "WOA + BRNN: An imbalanced big data classification framework using whale optimization and deep neural network," *Soft Comput.*, vol. 24, no. 8, pp. 5573–5592, Apr. 2020.

[12] J. Wang, P. Du, T. Niu, and W. Yang, "A novel hybrid system based on a new proposed algorithm—Multi-objective whale optimization algorithm for wind speed forecasting," *Appl. Energy*, vol. 208, pp. 344–360, Dec. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306261917314307

[13] S. Sankhwar, D. Gupta, K. C. Ramya, S. Sheeba Rani, K. Shankar, and S. K. Lakshmanaprabu, "Improved grey wolf optimization-based feature subset selection with fuzzy neural classifier for financial crisis prediction," *Soft Comput.*, vol. 24, no. 1, pp. 101–110, Jan. 2020.

[14] A. Madadi and M. M. Motlagh, "Optimal control of DC motor using grey wolf optimizer algorithm," *Tech. J. Eng. Appl. Sci.*, vol. 4, no. 4, pp. 373–379, 2014.

[15] K. R. Das, D. Das, and J. Das, "Optimal tuning of PID controller using GWO algorithm for speed control in DC motor," in *Proc. Int. Conf. Soft Comput. Techn. Implement. (ICSCTI)*, Oct. 2015, pp. 108–112.

[16] M. H. Sulaiman, Z. Mustaffa, M. R. Mohamed, and O. Aliman, "Using the gray wolf optimizer for solving optimal reactive power dispatch problem," *Appl. Soft Comput.*, vol. 32, pp. 286–292, Jul. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494615001994

[17] M. R. Mosavi, M. Khishe, and A. Ghamgosar, "Classification of sonar data set using neural network trained by gray wolf optimization," *Neural Netw. World*, vol. 26, no. 4, pp. 393–415, 2016.

[18] Q. Tu, X. Chen, and X. Liu, "Multi-strategy ensemble grey wolf optimizer and its application to feature selection," *Appl. Soft Comput.*, vol. 76, pp. 16–30, Mar. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494618306793

[19] Q. Tu, X. Chen, and X. Liu, "Hierarchy strengthened grey wolf optimizer for numerical optimization and feature selection," *IEEE Access*, vol. 7, pp. 78012–78028, 2019.

[20] X. Zhao, X. Zhang, Z. Cai, X. Tian, X. Wang, Y. Huang, H. Chen, and L. Hu, "Chaos enhanced grey wolf optimization wrapped ELM for diagnosis of paraquat-poisoned patients," *Comput. Biol. Chem.*, vol. 78, pp. 481–490, Feb. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1476927118307965

[21] F. A. enel, F. Gokçe, A. S. Yuksel, and T. Yigit, "A novel hybrid PSO–GWO algorithm for optimization problems," *Eng. Comput.*, vol. 35, no. 4, pp. 1359–1373, Dec. 2019, doi: 10.1007/s00366-018-0668-5.

[22] S. Arora, H. Singh, M. Sharma, S. Sharma, and P. Anand, "A new hybrid algorithm based on grey wolf optimization and crow search algorithm for unconstrained function optimization and feature selection," *IEEE Access*, vol. 7, pp. 26343–26361, 2019.
[23] H. Salimi, "Stochastic fractal search: A powerful metaheuristic algorithm," *Knowl.-Based Syst.*, vol. 75, pp. 1–18, Feb. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705114002822
[24] R. Bello, Y. Gomez, A. Nowe, and M. M. Garcia, "Two-step particle swarm optimization to solve the feature selection problem," in *Proc. 7th Int. Conf. Intell. Syst. Design Appl. (ISDA)*, Oct. 2007, pp. 691–696.
[25] S. H. S. Moosavi and V. K. Bardsiri, "Satin bowerbird optimizer: A new optimization algorithm to optimize ANFIS for software development effort estimation," *Eng. Appl. Artif. Intell.*, vol. 60, pp. 1–15, Apr. 2017, doi: 10.1016/j.engappai.2017.01.006.
[26] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0965997816300163
[27] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 495–513, Feb. 2016, doi: 10.1007/s00521-015-1870-7.
[28] I. Fister, X.-S. Yang, I. Fister, and J. Brest, "Memetic firefly algorithm for combinatorial optimization," Slovenian Res. Agency, Ljubljana, Slovenia, Tech. Rep. arXiv:1204.5165, Apr. 2012. [Online]. Available: https://cds.cern.ch/record/1443422
[29] M. M. Kabir, M. Shahjahan, and K. Murase, "A new local search based hybrid genetic algorithm for feature selection," *Neurocomputing*, vol. 74, no. 17, pp. 2914–2928, Oct. 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231211002748
[30] S. Jang, Y.-E. Jang, Y.-J. Kim, and H. Yu, "Input initialization for inversion of neural networks using K-nearest neighbor approach," *Inf. Sci.*, vol. 519, pp. 229–242, May 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025520300426
[31] E. Emary, H. M. Zawbaa, C. Grosan, and A. E. Hassenian, "Feature subset selection approach by gray-wolf optimization," in *Proc. Afro-Eur. Conf. Ind. Advancement*, A. Abraham, P. Krömer, and V. Snasel, Eds. Cham, Switzerland: Springer, 2015, pp. 1–13.
[32] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, Jan. 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231215010504
[33] M. Vosooghifard and H. Ebrahimpour, "Applying grey wolf optimizer-based decision tree classifer for cancer classification on gene expression data," in *Proc. 5th Int. Conf. Comput. Knowl. Eng. (ICCKE)*, Oct. 2015, pp. 147–151.
[34] K. Devanathan, N. Ganapathy, and R. Swaminathan, "Binary grey wolf optimizer based feature selection for nucleolar and centromere staining pattern classification in indirect immunofluorescence images," in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2019, pp. 7040–7043.
[35] P. Hu, J.-S. Pan, and S.-C. Chu, "Improved binary grey wolf optimizer and its application for feature selection," *Knowl.-Based Syst.*, vol. 195, May 2020, Art. no. 105746. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S095070512030160X
[36] Q. Al-Tashi, H. Md Rais, S. J. Abdulkadir, S. Mirjalili, and H. Alhussian, *A Review of Grey Wolf Optimizer-Based Feature Selection Methods for Classification*. Singapore: Springer, 2020, pp. 273–286, doi: 10.1007/978-981-32-9990-0_13
[37] G. Rebello and E. J. de Oliveira, *Modified Binary Grey Wolf Optimizer*. Singapore: Springer, 2020, pp. 148–179, doi: 10.1007/978-981-15-2133-1_7.
[38] E.-S. El-Kenawy and M. Eid, "Hybrid gray wolf and particle swarm optimization for feature selection," *Int. J. Innov. Comput. Inf. Control*, vol. 16, no. 3, pp. 831–844, 2020.
[39] T. P. Nguyen, T. T. Tran, and D. N. Vo, "Improved stochastic fractal search algorithm with chaos for optimal determination of location, size, and quantity of distributed generators in distribution systems," *Neural Comput. Appl.*, vol. 31, no. 11, pp. 7707–7732, Nov. 2019.
[40] L. H. Pham, M. Q. Duong, V.-D. Phan, T. T. Nguyen, and H.-N. Nguyen, "A high-performance stochastic fractal search algorithm for optimal generation dispatch problem," *Energies*, vol. 12, no. 9, p. 1796, May 2019, doi: 10.3390/en12091796.
[41] S. Khalilpourazari, B. Naderi, and S. Khalilpourazary, "Multi-objective stochastic fractal search: A powerful algorithm for solving complex multi-objective optimization problems," *Soft Comput.*, vol. 24, no. 4, pp. 3037–3066, Feb. 2020.
[42] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
[43] M. Elhoseny, A. Tharwat, and A. E. Hassanien, "Bezier curve based path planning in a dynamic field using modified genetic algorithm," *J. Comput. Sci.*, vol. 25, pp. 339–350, Mar. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877750317308906

EL-SAYED M. EL-KENAWY (Member, IEEE) is an Assistant Professor with the Delta Higher Institute for Engineering and Technology (DHIET), Mansoura, Egypt. He is inspiring and motivating students by providing a thorough understanding of a variety of computer concepts. He has pioneered and launched independent research programs. He is interested in computer science and machine learning field. He is adept at explaining some complex concepts in an easy-to-understand manner.

MARWA METWALLY EID received the Ph.D. degree in electronics and communications engineering from the Faculty of Engineering, Mansoura University, Egypt, in 2015. She has been an Assistant Professor with the Delta Higher Institute for Engineering and Technology, since 2011. Her current research interests are in image processing, encryption, wireless communication systems, and field-programmable gate array (FPGA) applications.

MOHAMED SABER received the B.Sc. degree in communication and electronics and the M.Sc. degree in electrical communications from the Faculty of Engineering, Mansoura University, in 2001 and 2006, respectively, and the Ph.D. degree in informatics and communications from the Graduate School of Information Science and Electrical Engineering, Kyushu University, Japan, in 2012. He works as an Assistant Professor with the Delta University for Science and Technology, Mansoura, Egypt. His research interests are digital signal processing, the design and implementation of digital communication systems on FPGA and DSP circuits, and synchronization (time, frequency, and phase) in digital receivers.

ABDELHAMEED IBRAHIM received the bachelor's and master's degrees in engineering from the Department of Computer Engineering and Control Systems, Mansoura University, Egypt, in 2001 and 2005, respectively, and the Ph.D. degree in engineering from the Faculty of Engineering, Chiba University, Japan, in 2011. He was with the Faculty of Engineering, Mansoura University, from 2001 to 2007, where he is currently an Assistant Professor of computer engineering with the Faculty of Engineering. His research interests include computer vision, pattern recognition, optimization, cloud computing, and virtual machine migration. He serves as a Reviewer for the *Journal of Electronic Imaging*, *Optical Engineering*, the IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, IEEE ACCESS, *Computer Standards and Interfaces*, the *Journal of Healthcare Engineering*, *IET Image Processing*, *Multimedia Tools and Applications*, and other respected journals.

• • •