

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

# An Improved Harris Hawks Optimization Algorithm With Simulated Annealing for Feature Selection in the Medical Field

Zenab Mohamed Elgamal<sup>1</sup>, Norizan Mohd Yasin<sup>2\*</sup>, Mohammad Tubishat<sup>3</sup>, Mohammed Alswaitti<sup>4</sup>, Seyedali Mirjalili<sup>5,6</sup>

<sup>1,2</sup> Faculty of Computer Science and Information Technology, University of Malaya 50603, Kuala Lumpur, Malaysia

<sup>3</sup> School of Technology and Computing, Asia Pacific University of Technology and Innovation, Kuala Lumpur, Malaysia

<sup>4</sup> School of Electrical and Computer Engineering (ICT), Xiamen University Malaysia, Bandar Sunsuria, 43900 Sepang, Malaysia

<sup>5</sup> Centre for Artificial Intelligence Research and Optimisation, Torrens University Australia, 90 Bowen Terrace, Fortitude Valley, QLD4006, Australia

<sup>6</sup> Yonsei Frontier Lab, Yonsei University, Seoul, Korea

\* Corresponding author: Norizan Mohd Yasin<sup>2</sup> ([norizan@um.edu.my](mailto:norizan@um.edu.my))

**ABSTRACT** Harris Hawks Optimization (HHO) algorithm is a new metaheuristic algorithm, inspired by the cooperative behavior and chasing style of Harris' Hawks in nature called surprise pounce. HHO demonstrated promising results compared to other optimization methods. However, HHO suffers from local optima and population diversity drawbacks. To overcome these limitations and adapt it to solve feature selection problems, a novel metaheuristic optimizer, namely Chaotic Harris Hawks Optimization (CHHO), is proposed. Two main improvements are suggested to the standard HHO algorithm. The first improvement is to apply the chaotic maps at the initialization phase of HHO to enhance the population diversity in the search space. The second improvement is to use the Simulated Annealing (SA) algorithm to the current best solution to improve HHO exploitation. To validate the performance of the proposed algorithm, CHHO was applied on 14 medical benchmark datasets from the UCI machine learning repository. The proposed CHHO was compared with the original HHO and some famous and recent metaheuristics algorithms, containing Grasshopper Optimization Algorithm (GOA), Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Butterfly Optimization Algorithm (BOA), and Ant Lion Optimizer (ALO). The used evaluation metrics include the number of selected features, classification accuracy, fitness values, Wilcoxon's statistical test ( $P$ -value), and convergence curve. Based on the achieved results, CHHO confirms its superiority over the standard HHO algorithm and the other optimization algorithms on the majority of the medical datasets.

**INDEX TERMS** Harris Hawks Optimization (HHO) algorithm; Feature Selection; Wrapper method; Chaos Theory; Simulated Annealing (SA).

## I. INTRODUCTION

Recently, high dimensional data become an essential source for many Machine Learning (ML) research, such as data mining and pattern recognition. However, increasing data volume and data dimensionality causes many problems, like the appearance of noisy, irrelevant, and redundant data [1], [2]. This problem increases the ML complexity and decreases the classification performance. Also, the majority of ML classifiers cannot relate with all features included in the complex dataset. However, this stated problem can affect data mining performance and pattern recognition since it mainly depends on the ML classifier. Thus, Feature Selection (FS) is a critical process in ML to select relevant features and remove noisy and irrelevant ones. In high-dimensional datasets, choosing the most significant features is a challenging task.

However, many studies have proven that; the FS methods can efficiently select the crucial features and remove irrelevant and redundant ones [3], [4], and [5]. Also, reducing the computational complexity and required storage space are essential tasks of FS, which consequently enhance the classifier [6]. Therefore, the FS process has the potential to improve the classification performance of the ML classifier substantially.

Generally, the FS process consists of four parts: feature subset search, evaluation, search stop criteria, and validity [2]. Based on the evaluation criteria, FS methods are separated into two main types: Filters and Wrappers methods. Filter-Based Methods (FBM) utilize statistical functions to choose and rank the feature subsets. Additionally, FBM, such as Chi-square, Information Gain, Relief, and Gini-Index has no direct contact

with the classifier, and they operated before employing the classifier [7]. On the other hand, the Wrapper-Based Methods (WBM) have direct contact with the used classifier [8]. Many studies have been applied to WBM methods in optimization algorithms for feature selection purposes [9], [10], [8]. WBM is computationally more expensive, but it achieved better results than the FBM. Commonly, WBM is employed in FS problems since it considers the classification performance and feature reduction conditions, and due to its ability to interact directly with the classifier.

In WBM, the fitness function is applied to assess the FS process depending on the accuracy of classification. In order to improve the accuracy in FS, various studies have been conducted using optimization algorithms [1], [10], [11], and [12]. However, the main goal of using optimization algorithms in FS is to determine the optimal features or the features near to the optimal during a reasonable time (i.e., the optimal feature sets). On the other hand, the standard inclusive-search seeks to find all possible combinations of features from the complete set of features. This process is considered a time-consuming search and is regarded as an NP-hard problem [13]. Therefore, optimization algorithms are needed to solve the problem stated above appropriately due to their ability to obtain a solution that might be optimal or near the optimal solution. However, in high-dimensional problems such as the FS problem, the optimization algorithms suffer from local optima and population diversity problem. To find a tuned algorithm and make it appropriate for feature selection problems, we focus on improving the HHO algorithm by proposing two contributions folds. The first is to employ chaotic maps to improve population diversity. The second is to use the Simulated Annealing (SA) to improve the exploitation capability of the algorithm and avoid local optima problem.

In this study, an improvement of the standard HHO algorithm named CHHO is proposed. The chaotic map algorithm is used to initialize the solutions (search-agents) at the initialization phase of HHO. The proposed version is expected to accelerate the convergence rate of the HHO and diversify the generated solutions of HHO. Furthermore, we used Simulating Annealing (SA) algorithm to improve the exploitation ability of HHO and avoid the local optima problem. In the literature, different forms of hybrids optimization algorithms were proposed for feature selection problems. Still, as far as the authors' knowledge, this is the first time that a hybrid model using HHO with chaos theory and SA algorithm to be proposed and applied in feature selection problems. CHHO will be used to improve the classification performance for the feature selection problem. In this work, the main contributions can be summarized as follows:

- 1) CHHO: An improved variant of the HHO algorithm to solve its weaknesses and make it suitable for the feature selection problem.
- 2) Two main improvements were introduced into standard HHO including:

- The use of the Chaotic maps at the initialization phase of HHO to improve its solution diversity.
  - Simulated Annealing (SA) algorithm is combined with HHO to improve its exploitation and avoid local optima problem.
- 3) The development of a wrapper-based feature selection model based on the CHHO algorithm.
  - 4) To evaluate the performance of the CHHO algorithm, CHHO compared with original HHO and other popular optimization algorithms involving Grasshopper Optimization Algorithm (GOA), Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Butterfly Optimization Algorithm (BOA), and Ant Lion Optimizer (ALO). The experiments were conducted on 14 benchmark medical datasets from the UCI machine learning repository. The used evaluation metrics include the number of features, classification accuracy, fitness values,  $P$ -value, and convergence rate.

The rest of the paper is structured as follows. Section II formally introduces the related works in the literature. Section III provides all the details about the HHO algorithm. In Section IV, the proposed CHHO algorithm is presented. Section V presents the performed experiments and achieved results. Finally, Section VI concludes the paper.

## II. RELATED WORK

Recently, optimization algorithms have become very popular due to their demonstrated efficiency in solving feature selection problems. Examples of these algorithms are Butterfly Optimization Algorithm (BOA) [14], Grasshopper Optimization Algorithm (GOA) [15], Ant Lion Optimizer (ALO) [11], The Whale Optimization Algorithm (WOA) [16], Slap Swarm Algorithm (SSA) [7]. Despite the unique structure of meta-heuristic algorithms, there is a common characteristic. The majority of the techniques start with a random population initialization, solution evaluation on each iteration based on a fitness function, solution updating, and eventually determining the best solution based on the termination criterion. The phases, as mentioned earlier, define the search behavior were it is mostly referred to as exploration and exploitation phases. In the former stage, the optimization algorithm attempts to discover the best region of the search space. The optimization algorithms apply its stochastic operations as much as possible to examine all areas and sections of the feature space deeply.

On the contrary, the next phase intends to enhance the search process for local regions rather than all feature spaces. Usually, exploitation is performed after the exploration phase [17]. In most complex applications, optimization algorithms are trapped in local optima due to the incorrect balance between the exploitation and exploration and the randomness nature of the initialization process. One of the methods used in literature to solve the population diversity problem is chaos theory. Chaos Optimization Algorithm (COA) [18] is one of the chaos implementations that takes advantage of the nature of chaotic structures. It has proven

that changing the random parameter values with a chaotic system can enhance classification performance [19]. As a result, several optimization algorithms studies combined the chaos theory to improve the performance and to adjust specific parameters. Examples of these implementations are Chaotic Crow Search Optimization (CCSA) [19], where a chaotic system was used to overcome the low convergence rate and local optima entrapment. Chaotic Whale Optimization Algorithm (CWOA) [20] applied a chaos system to improve the global convergence rate and obtain improved performance. Chaotic Genetic Algorithm (CGA) [21] examined a chaotic system to improve GA performance. Chaotic Gray Wolf Optimization (CGWO) [22] applied a chaotic system to accelerate the global convergence rate. Chaotic Grasshopper Optimization Algorithm (CGOA) [23] utilized the chaos system to balance exploration and exploitation more effectively. These algorithms and more were applied in different fields and applications. The noticeable improvements of adding chaotic maps in these algorithms, which was confirmed by the reported results, motivated the current study to combine chaos theory with HHO to improve population diversity.

On the other side of the search behavior, Simulated Annealing (SA) is proposed to solve the HHO local optima problem in high dimensional FS. SA was presented in 1983 by Kirkpatrick. It is considered to be a hill-climbing method that attempts to enhance the candidate solution for the objective function [24]. SA algorithm was used to improve the exploitative capability of the algorithm and prevent local search problems. Many optimization algorithms used SA to enhance the local search strategy. For instance, it was used to evaluate the performance of feature selection [9], to improve the best solution after each iteration [25], to enhance the exploitation search capability [26], to evaluate PSO performance as a wrapper-based method [8]. The performance obtained by employing the SA in these previous studies inspired this study to include the SA algorithm into the iteration process to enhance the local search in the FS problem.

Optimization algorithms have been applied successfully for FS in many applications such as data mining [27] using Particle Swarm Optimization, pattern recognition [28] using Binary Genetic Swarm Optimization, Medical applications [5] using Crow Search Optimization, and image analysis [29] using Genetic Algorithm Optimization, image processing [30], [31], [32] using Optimized Deep Neural Network, and there are many more. Nowadays, FS is an essential step to preprocess high-dimensional datasets. It must be pointed that there are representative computational intelligence algorithms that have been applied to improve the FS in different studies such as [7], [9], [33], [34], [27], [46], and [47]. The optimization methods aim to obtain the optimal solution for FS (i.e., significant feature subset) within an appropriate time and cost. The successful application of these algorithms, which confirmed by performance results in different fields, inspired this study to apply the improved HHO to the FS problem.

Harris Hawks Optimization (HHO) is a recent metaheuristic algorithm developed by [35]. Inspired by the cooperative behavior and chasing style of Harris' Hawks in nature called surprise pounce. HHO is capable of solving unconstructed benchmark problems compared to other popular optimization algorithms, as reported by the author. Also, HHO is a population-based and gradient-free optimization technique to be applied to any optimization problem subject to an engineering formulation. However, HHO is considered a random optimization algorithm and suffers from various issues as population diversity and local optima when dealing with high-dimensional datasets. The reasons above and HHO characteristics motivated this study to improve the performance of the HHO in the feature selection problem. In addition to that, to investigate the performance of the improved HHO (CHHO) in obtaining the optimal subset of the feature selection while achieving better classification performance. The importance of this work comes from the fact that the Harris Hawks Optimization algorithm has been applied in many fields such as image processing [36], Optimal Power Flow Problem [37], drug design, and discovery [38]. Also, HHO applied in feature selection by using the Elite Opposition-Based Learning method [39] and provide good results. In the following section, basics and background about the HHO algorithm.

### III. BASICS AND BACKGROUND

#### A. HARRIS HAWKS OPTIMIZATION ALGORITHM

Harris Hawks Optimization Algorithm (HHO) is a novel optimization algorithm developed by Heidari and Mirjalili et al. in 2019 [35]. The algorithm simulates the collaborative behavior and hunting technique of Harris Hawks in nature named surprise pounce. In this smart strategy, the Hawks collaboratively attack from many directions to surprise the prey. Harris Hawks exposes several chase styles based on the nature of the plots and the escaping patterns of the victim. The standard HHO algorithm proposes the exploration and exploitation strategies, motivated by exploring prey, surprise pounce, and the unique attacking technique of Harris Hawks. HHO algorithm. The HHO algorithm is a population-based and slope optimization method. Therefore, it was utilized to many optimization problems subject to an appropriate formulation. In the next steps, the HHO mathematically simulates these useful techniques and behavior to develop an optimization algorithm.

##### 1) INITIALIZATION PHASE

In this phase, the objective function and the search-space are defined. Also, the initial population-based chaotic maps are initiated. In addition, all parameter values are set.

##### 2) EXPLORATION PHASE

In this phase, all Harris hawks are considered as candidate solutions. In each iteration, the fitness value is computed for all these possible solutions based on the intended prey. Two approaches are applied to mimic the exploration

performances of Harris Hawks in the search-space specified in (1)

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1 |X_{rand}(t) - 2r_2 X(t)| & q \geq 0.5 \\ (X_{rabbit}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)) & q < 0.5 \end{cases} \quad (1)$$

where  $X(t+1)$  is the position of Hawks in second iteration  $t$ .  $X_{rabbit}(t)$  is the prey position and the  $X_{rand}(t)$  stands for the random solution chosen in the current population.  $X(t)$  is the position vector of Hawks in the current iteration  $t$ , the  $r_1, r_2, r_3, r_4$  and  $q$  are random scaled factor within  $[0,1]$ , which are updated in each iteration,  $LB$  and  $UB$  are the upper and lower bounds of variables, and the  $X_m$  is the average number of the solutions.

This intended approach generates the positions of Hawks within  $(UB - LB)$  bounds based on two rules; 1) create the solutions based on randomly selected hawk from the current population and the other hawks. 2) create the solutions based on the prey location, the average position of Hawks, and random scaled factors. While  $r_3$  is a scaling factor, once the value of  $r_4$  is close to 1, it will help increase the randomness of the rule. In this rule, a randomly scaled movement length is added to  $LB$ . A random scaled component is considered to provide more diversification techniques to explore different areas of the feature space. The average position of hawks (solutions) is formulated in (2).

$$X_m(t) = \frac{1}{N} \sum_{i=1}^N X_i(t) \quad (2)$$

where  $X_m(t)$  is the average number of the solutions in the current iteration.  $N$  indicates all possible solutions.  $X_i(t)$  implies the location of each solution in iteration  $t$ , which created based chaos theory. Usually, in Eq. (1), rule one is applied when the hawk uses the information from the random hawks to catch the prey. While rule two is applied when all hawks share the best solution and the best hawk employed.

### 3) TRANSITION FROM EXPLORATION TO EXPLOITATION

This phase explains the movement of HHO from exploration to exploitation, based on the energy of the prey ( $E$ ). HHO assumes that the energy of prey is reduced gradually through the escaping actions.  $E_0$  is the initial energy decreases from  $[1, -1]$ , modeled in (3).

$$E = 2E_0 \left(1 - \frac{t}{T}\right), \quad E_0 \in [-1,1] \quad (3)$$

where  $T$  indicates the maximum number of iterations, and  $t$  is the current iteration.

### 4) EXPLOITATION PHASE

In this phase, the exploitation phase is accomplished using four approaches at parameter sets. These approaches are based on the position identified in the exploration phase. However, the prey tries to escape frequently, while the hawks tracing and try to catch it. HHO exploitation is mimic the attacking strategy of the Hawks by using four possible approaches. These approaches are the soft besiege, hard besiege, soft besiege with progressive rapid dives, and hard besiege with progressive rapid dives. These approaches based on two variables  $r$  and  $|E|$ , which specify the executed approach. Where  $|E|$  is the escaping energy of the prey,  $r$  refers to the probability of escaping, where  $r < 0.5$  indicates the higher possibility for the prey to escape successfully and  $r \geq 0.5$  for unsuccessfully escape. A summary of these approaches are presented as follows:

In the soft besiege approach, where  $r \geq 0.5$  and  $|E| \geq 0.5$ , the rabbit still has some energy to escape, while the hawks are softly encircling the prey make it lose more energy before performing the surprise pounce. Soft besiege mathematically formulated in (4), (5), and (6).

$$X(t+1) = \Delta X(t) - E |J X_{rabbit} - X(t)| \quad (4)$$

$$\Delta X(t) = X_{rabbit} - X(t) \quad (5)$$

$$J = 2(1 - r_5), \quad r_5 \in [0,1] \quad (6)$$

where  $\Delta x(t)$  is the difference among the position vector of the prey and the current location in iteration  $t$ , and  $J$  presents the jump power of the prey and  $r_5$  is a random variable.

In the hard besiege strategy, where  $r \geq 0.5$  and  $|E| < 0.5$ , the prey is tired with a weak escaping chance. In this condition, the hawk hardly encircles the prey to perform the final surprise pounce. Thus, the solution is updated using (7).

$$X(t+1) = X_{rabbit}(t) - E |\Delta X(t)| \quad (7)$$

Eq. (8) shows the soft besiege with progressive rapid dives approach. In this condition  $r < 0.5$  and  $|E| \geq 0.5$ , the prey still has the energy to escape. The hawk moves smartly around the prey and patiently dives before the surprised pounce. This action is considered as intelligent soft besiege, where the position of the hawks is updated in two steps. In the first step, the hawks move toward the prey by estimating the next move of the prey as formula (8)

$$Y = X_{rabbit}(t) - E |X_{rabbit}(t) - X(t)| \quad (8)$$

In the second step, the hawk decided whether to dive or not, based on the comparison between the previous dive and the possible result. If it is not, the hawks producing irregular dive, based on the Levy Flight ( $LF$ ) concept, as formulated in (9)



$$Z = Y + S \times LF(Dim) \quad (9)$$

where the dimension of solutions is defined as  $Dim$ ,  $S$  is a random vector of size  $1 \times dim$ .  $LF$  is the Levy Flight function calculated using (10).

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \quad \sigma = \left( \frac{\Gamma(1+\beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{\frac{\beta-1}{2}}} \right)^{\frac{1}{\beta}} \quad (10)$$

where  $\beta$  is a default constant set automatically to 1.5, and  $u, v$  are random values within  $[0,1]$ . Therefore, updating the Harris hawks positions in with progressive rapid dives can be formulated in (11)

$$X(t+1) = \begin{cases} Y & \text{if } F(Y) < F(X(t)) \\ Z & \text{if } F(Z) < F(X(t)) \end{cases} \quad (11)$$

where  $Y$  and  $Z$  are performed using (8) and (9), and both refer to the new iteration's next location.

The last approach is called hard besiege with progressive rapid dives, where  $r < 0.5$  and  $|E| < 0.5$ . In this condition, the prey has no energy to escape, and the Harris hawks attempt to reach the prey by rapid dives before performing a surprise pounce to catch the prey. The movement of the hawks in the condition is formulated in (12)

$$X(t+1) = \begin{cases} Y & \text{if } F(Y) < F(X(t)) \\ Z & \text{if } F(Z) < F(X(t)) \end{cases} \quad (12)$$

where  $Y$  is set as in (13), and  $Z$  is updated as in (14)

$$Y = X_{rabbit}(t) - E|JX_{rabbit}(t) - X_m(t)| \quad (13)$$

$$Z = Y + S \times LF(Dim) \quad (14)$$

Finally, the classification accuracy computed using the fitness function set in Eq. (15). The fitness function includes the computation of classification error, as mathematically formulated in (15)

$$Fitness = \alpha \gamma_R(D) + \beta \frac{|R|}{|N|} \quad (15)$$

where  $\alpha \gamma_R(D)$  refer to the classification error rate of the used classifier KNN. Besides,  $|R|$  is a cardinal number of the selected subset and  $|N|$  is the total number of features in the dataset,  $\alpha$ , and  $\beta$  are two parameters corresponding to the importance of classification quality and subset length,  $\alpha \in [0, 1]$  and  $\beta = (1 - \alpha)$  approved in [25].

## B. CHAOTIC MAPS

Chaos optimization is a dynamic system. This system is one of the most modern methods to search for the global optimum solutions in a search space. In this study, we have implemented ten chaotic maps to replace the Harris Hawks position's random variables as listed in Table I. The main idea of it is to replace the random initialization variables with chaotic maps variables. The initial value set to all chaotic maps  $x_0$  is 0.7. While  $y$  refers to the symbol of chaotic sequence  $x$ . Also  $x_y$  is the  $y$ th number on the chaotic sequence. The remaining variables  $d, c$  and  $\mu$  are the control variables that help to define the chaotic performance of the algorithm.

There are some studies available in the literature were they utilized chaotic maps to improve HHO. Examples of these studies, in [40], they replaced the random parameters in HHO with a chaotic logistic map. Moreover, in [41] they replaced the random parameters in Multi-Verse Optimizer (MVO) using chaotic maps. Also, they used HHO as a local search operator within MVO to solve its local optima problem. Furthermore, in [42], they replaced the random

TABLE I  
THE TEN APPLIED CHAOTIC MAPS

S.No	Chaotic Map Name	Definition	Range
CHH01	Singer	$x_{y+1} = \mu(7.86x_y - 23.31x_y^2 + 28.75x_y^3 - 13.302875x_y^4)$ , $\mu = 1.07$	(0,1)
CHH02	Sinusoidal	$x_{y+1} = c x_y^2 \sin(\pi x_y)$ , $c = 2.3$	(0,1)
CHH03	Chebyshev	$x_{y+1} = \cos(a \cdot \cos^{-1} x_y)$	(-1,1)
CHH04	Circle	$x_{y+1} = x_y + b - \left(\frac{a}{2\pi}\right) \sin(2\pi x_y) \bmod(1)$	(0,1)
CHH05	Tent	$x_{y+1} = \begin{cases} \frac{x_y}{.7}, & x_y < 0.7 \\ \frac{10}{3}(1 - x_y), & x_y \geq 0.7 \end{cases}$	(0,1)
CHH06	Sine	$x_{y+1} = \frac{c}{4} \sin(\pi x_y)$ , $c = 4$	(0,1)
CHH07	Piecewise	$x_{y+1} = \begin{cases} \frac{x_y}{1}, & 0 \leq x_y < 1 \\ \frac{x_y - 1}{.5 - 1}, & 1 \leq x_y < .5 \\ \frac{1 - l - x_y}{.5 - 1}, & .5 \leq x_y < 1 - l, l = . \\ \frac{1 - x_y}{l}, & 1 - l \leq x_y < 1 \end{cases}$	(0,1)
CHH08	Logistic	$x_{y+1} = cx_y(1 - x_y)$ , $c = 4$	(0,1)
CHH09	Iterative	$x_{y+1} = \sin \frac{c\pi}{x_y}$ , $c = 0.7$	(-1,1)
CHH010	Gauss/mouse	$x_{y+1} = \begin{cases} 1, & x_y = 0 \\ \frac{1}{\bmod(x_{y,1})}, & \text{otherwise} \end{cases}$	(0,1)

parameter in HHO with chaotic map value. However, the main differences in our work compared to these previous improvements on HHO include the following: 1) chaotic maps used to initialize the solutions (search\_agents) positions at the initialization phase of HHO instead of using the standard random numbers for initializing the HHO solutions positions. 2) we utilized SA as a local search operator within HHO to solve its local optima problem.

### C. SIMULATED ANNEALING

Simulated Annealing (SA) was proposed by Kirkpatrick et al. in 1983. It is a local search algorithm based on a single solution. It is considered a hill-climbing method that repeatedly tries to improve the available solutions for the objective function [24]. The improved solution will be accepted, while the worst solution will be taken with a well-defined probability of the algorithm to avoid the local optima. The probability of choosing a worse solution is determined by Boltzmann probability function  $P = e^{-\theta T}$ , where  $\theta$  is the difference of evaluation of the objective function between the best solution ( $Sol_{trial}$ ) and the trial solution ( $Sol_{best}$ ). In the same time,  $T$  is a parameter (named temperature) that periodically decreasing throughout the search process [24].

## IV. THE PROPOSED CHAOTIC HARRIS HAWKS ALGORITHM (CHHO)

In this study, feature selection is regarded as a multi-objective optimization problem, in which two contradictory goals must be achieved. These goals are to minimize the number of selected features and maximize the classification accuracy. In other words, to reach a minimum number of selected features in the solution that leads to higher classification accuracy. Every solution is calculated according to the proposed fitness function, which depends on the KNN classifier [43], to obtain the classification accuracy of the solution as well as the number of selected features. To balance the number of selected features in each solution (to be minimum) with the classification accuracy (to be maximum), we have chosen the fitness function in the equation (15) is applied for evaluating the search agents in the algorithm.

Based on the previous studies, which utilized HHO for solving different problems and confirmed its outperformance in comparison to other recent and well-known optimization algorithms, we have been motivated to apply HHO on feature selection problem. However, the standard HHO algorithm suffers from two significant problems when applied to high-dimensional problems such as the feature selection problem. These problems are including 1) problem of solutions diversity; 2) problem of local optima. Therefore, to improve the HHO algorithm and make it suitable for the feature selection problem, two main improvements are introduced in this study to solve the weakness of the HHO algorithm. The first improvement includes the use of chaotic maps at the initialization phase to improve the diversity of the solutions. The second improvement consists of using the SA algorithm with the HHO algorithm to enhance its exploitation and avoid being stuck in local optima. The details of these contributions into HHO are detailed as follows:

In the CHHO algorithm, the chaotic map value replaced the randomly generated values for initializing the Harris Hawks population positions at the initialization phase. The chaotic values are generated from chaotic maps. In this work, ten chaotic maps were applied to the algorithm to contrast the effect of employing different chaotic maps. These maps are Singer, Sinusoidal, Chebyshev, Circle, Tent,

Sine, Piecewise, Logistic, Iterative, and Gauss/mouse. The maps with its statistical equations are listed in Table I. These maps significantly increase the convergence rate and the fitness performance of the HHO, as will be demonstrated later in the experimental discussion section.

Moreover, the second improvement is to embed the SA in the CHHO algorithm to enhance its local searchability. This embedding will improve the exploitation capability of the algorithm. After implementing chaotic maps and obtaining the best solution, SA is used to improve the current best solution at the end of each HHO iteration. The pseudocode of the proposed CHHO algorithm is illustrated in Algorithm 1.

---

### Algorithm 1: Pseudo-code of CHHO algorithm

---

**Inputs:** The population size  $N$  and maximum number of iterations  $T$

**Outputs:** The location of the rabbit and its fitness value

**Initialize the chaotic population**  $X_i$  ( $i = 1, 2, \dots, N$ )

**while** (fitness value  $\neq$  stopping criteria) **do**

Compute the fitness values of hawks

Set  $X_{rabbit}$  as the location of rabbit (best location)

**for** (each hawk ( $X_i$ )) **do**

Update the initial energy  $E_0$  and jump strength  $J$

$E_0 = 2rand() - 1$ ,  $J = 2(1 - rand())$

Update the  $E$  using Eq. (3) ▷ Exploration phase

**if** ( $|E| \geq 1$ ) **then**

Update the location vector using Eq. (1)

**if** ( $|E| < 1$ ) **then** ▷ Exploitation phase

**if** ( $r \geq .5$  and  $|E| \geq .5$ ) **then** ▷ Soft besiege

Update the location vector using Eq. (4)

**else if** ( $r \geq .5$  and  $|E| < .5$ ) **then** ▷ Hard besiege

Update the location vector using Eq. (7)

**else if** ( $r < .5$  and  $|E| \geq .5$ ) **then** ▷ Soft besiege

with progressive rapid dives

Update the location vector using Eq. (11)

**else if** ( $r < .5$  and  $|E| < .5$ ) **then** ▷ Hard besiege

with progressive rapid dives

Update the location vector using Eq. (12)

**Apply SA**

$T = t +$

Return  $X_{rabbit}$

---

To explain the computational complexity of the CHHO algorithm. The computational complexity stands on initialization, fitness evaluation, and updating of candidate solutions processes. Considering all the possible solutions  $N$ , the computational complexity of the initialization process is  $O(N)$ . Also, the computational complexity of the updating process  $O(T * N) + O(T * N * Dim)$ , which is contained the updating the search-agents positions and finding the best solution,  $T$  indicates the maximum number of iterations and  $Dim$  is the dimension of the search space. However, the computational complexity of utilizing the SA search strategy can be identified as  $O(T * I * S)$ , where the  $I$  is the number of SA iterations, and  $S$  is the SA search strategy. Consequently, the computational complexity of CHHO is  $O(N * (T + TDim + 1) + TIS)$ .

The proposed CHHO is also presented in the form of a flowchart in Figure. 1. The starts of the CHHO process by

initializing the Harris Hawks (search-agents) population using chaotic maps. Then, compute the fitness value of the candidate solution. After that, SA applied in each iteration. Then the evaluation of fitness value using wrapper FS based KNN classifier. All earlier process will be reiterated until the stopping condition is satisfied.

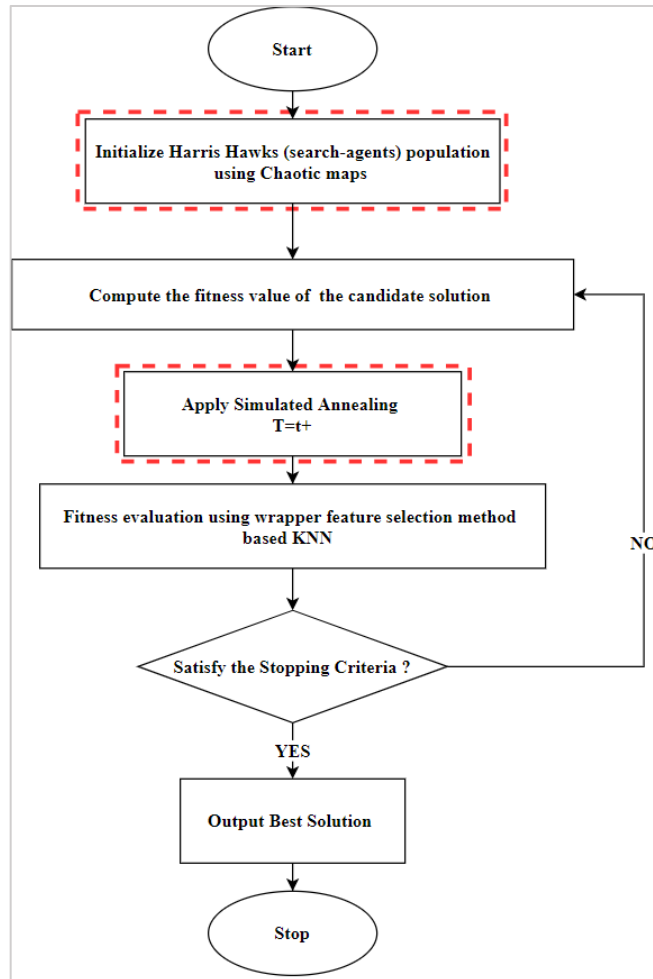


FIGURE 1. The flowchart of the proposed CHHO algorithm using chaotic maps and SA techniques.

## V. EXPERIMENTAL RESULTS AND DISCUSSION

To validate and evaluate the performance of the proposed CHHO algorithm. CHHO was compared with some famous and recent optimization algorithms, including GOA, GA, PSO, BOA, and ALO algorithms. All experiments were conducted on 14 benchmark datasets from the UCI repository. The used datasets and all experiment details presented in the following steps:

### A. DATASETS DETAILS

In this experiment, fourteen medical benchmark datasets were used from the UCI machine learning repository. The details of these datasets are presented in Table II. All experiments were conducted using the settings stated in Table III.

TABLE II  
UCI MEDICAL DATASETS DESCRIPTION

ID	Dataset	No. of Features	No. of Instances
Ds1	Lung	57	32
Ds2	Heart Disease	14	270
Ds3	Lymphography	19	148
Ds4	Primary Tumor	18	339
Ds5	Parkinson	29	1040
Ds6	Breast Cancer	10	699
Ds7	CTG	23	2126
Ds8	Cervical Cancer	36	858
Ds9	Dermatology	35	366
Ds10	HCV	29	1385
Ds11	ILPD	11	583
Ds12	SPECT	23	267
Ds13	WDBC	31	569
Ds14	Fertility dataset	10	100

TABLE III  
PC SPECIFICATIONS

Name	Detailed settings
CPU	Core(TM) i7 1.80GHz
RAM	16.0GB
OS	Windows10
Language	MATLAB R2020a

### B. ALGORITHMS AND EXPERIMENTS PARAMETER SETTING

In all experiments, the wrapper method based KNN classifier (10-fold cross-validation) was utilized to validate the fitness performance of the proposed algorithm. This validation technique uses  $k - 1$  folds for training and one-fold for testing. Also, the parameter settings of other baseline optimization algorithms GOA, GA, PSO, BOA, and ALO, are shown in Table IV. Furthermore, for all algorithms, the population size was set to 10, and the maximum number of iterations was 50. The classification accuracy was chosen as a critical metric for evaluating and validating the optimization algorithms performance. The results are presented in Table V, were the results performed based on the average number of 20 runs, in each run 50 iteration modified by the SA algorithm.

TABLE IV  
PARAMETER SET OF OPTIMIZATION ALGORITHMS

Algorithm	Parameter	Ref.
GOA	cMax=1; cMin=0.00004	[44]
GA	Crossover Ratio 0.9 Mutation Rate 0.2	[45]
PSO	Inertia Weight value 0.9 Inertia Weight Damping Ratio 0.4 Accelerating-constant values are C1=2, C2=2	[45]
BOA	Probability-switch 0.8 Power exponent=0.1 Sensory modality=0.01	[14]

C. RESULTS AND DISCUSSION

In this section, we present the summary and results of all experiments. Two main experiments were conducted using the CHHO algorithm to customize the algorithm to solve the feature selection problem. The first experiment includes evaluating the performance of the CHHO with the original HHO algorithm, where ten different variants of CHHO were tested using different previously mentioned chaotic maps. The second experiment includes the comparison of the CHHO algorithm with other state-of-the-art and recent optimization algorithms GOA, GA, PSO, BOA, and ALO. Four metrics were used in all experiments, including Average Number of selected features (AFS), Average Fitness Value (AFV), Average of Classification Accuracy (AAC), and the *P*-value statistical test to distinguish the performance of the competing algorithms in each phase. These metrics were calculated within 20 runs, while the SA algorithm chose the best solution among 50 iterations on each run.

1) THE EFFECT OF DIFFERENT CHAOTIC MAPS WITH SA ON THE STANDARD HHO ALGORITHM PERFORMANCE

The first aim of this experiment is to evaluate the performance of CHHO with ten chaotic maps and determine the best chaotic map while including the SA algorithm into the iterations process. In Table V, the evaluations of CHHO

with ten chaotic maps and the standard HHO are reported. The *P*-value of Wilcoxon’s statistical test was used to evaluate the HHO with ten CHHO variants to highlight the significance of the improvement. In Table V, *P*-value is underlined were the significant *P*-value < 0.5. It must be indicated that CHHO1, CHHO2, ..., CHHO10 in Table V refer to the ten implemented chaotic maps as presented in Table I. Additionally, it is worth mentioning that Ds1, Ds2, ..., Ds14 in Table V refer to the 14 benchmark datasets as shown in Table II. It can be seen from Table V. that CHHO with chaotic maps outperformed the standard HHO. Also, for the statistical comparison between chaotic maps among all datasets, the best results of the *P*-value are underlined as presented in Table VI. *P*-value results in Table VI show the outperforming expansion of Sine chaotic map in comparison with all other maps. As observed in Table VI, the CHHO6 (Sine map) variant obtained significant statistical results compared to the others in most cases. Therefore, this result concludes that CHHO6 is a significant improvement over the original HHO algorithm. As shown in Table V, the CHHO2 variant (Sinusoidal) provided the best number of selected features followed by the CHHO6 variant. In addition, the CHHO6 variant obtained the best classification accuracies while the CHHO2 variant obtained the best fitness values.

TABLE V  
THE PERFORMANCE RESULTS OF CHHO’S WITH THE STANDARD HHO ALGORITHM IN TERMS OF AVERAGE SELECTED FEATURES (AFS), AVERAGE FITNESS VALUE (AFV), AVERAGE CLASSIFICATION ACCURACY (AAC) AND *P*-VALUE.

Ds1	AFS	AFV	AAC	<i>P</i> -Value	Ds2	AFS	AFV	AAC	<i>P</i> -Value
HHO	4.764	0.0003571	0.6		HHO	3.7082	0.10524	0.8963	
CHHO1	3.4607	0.012643	<b>0.8375</b>	3.79E-04	CHHO1	<b>3.712</b>	0.06518	0.93704	1.43E-03
CHHO2	1.7913	<b>0.0001696</b>	0.65	1.96E-02	CHHO2	3.9146	0.06717	0.93519	8.11E-05
CHHO3	2.1691	0.016812	0.48333	<b>2.36E-01</b>	CHHO3	3.9	0.07083	0.93148	1.44E-06
CHHO4	1.7749	0.0001875	0.6	4.49E-02	CHHO4	4.1686	0.07282	0.92963	1.89E-06
CHHO5	2.0136	0.012571	0.6375	4.29E-02	CHHO5	4.0289	<b>0.05983</b>	<b>0.94259</b>	1.08E-03
CHHO6	2.5185	0.0002054	0.6	4.52E-02	CHHO6	4.1806	0.07278	0.92963	3.90E-03
CHHO7	2.1462	0.029152	0.67083	2.91E-02	CHHO7	4.0294	0.07458	0.92778	3.57E-03
CHHO8	2.5442	0.0001875	0.7	7.20E-03	CHHO8	3.8673	0.09096	0.91111	9.81E-05
CHHO9	2.9034	0.016795	0.78333	1.51E-03	CHHO9	4.25	0.0766	0.92593	3.18E-03
CHHO10	<b>1.7136</b>	0.0001786	0.6	4.47E-02	CHHO10	4.05	0.07462	0.92778	4.63E-05
Ds3	AFS	AFV	AAC	<i>P</i> -Value	Ds4	AFS	AFV	AAC	<i>P</i> -Value
HHO	6.6012	0.32224	0.67787		HHO	7.2008	0.13649	0.86618	
CHHO1	5.7234	0.23306	0.76778	4.90E-03	CHHO1	6.8	0.09453	0.90856	1.14E-02
CHHO2	<b>5.5168</b>	0.18202	0.8192	1.55E-03	CHHO2	7.7632	<b>0.08925</b>	<b>0.91448</b>	1.22E-03
CHHO3	6.5	0.17649	0.82538	2.75E-05	CHHO3	6.7	0.1149	0.88792	4.38E-05
CHHO4	6.2662	0.24451	0.75655	5.62E-04	CHHO4	6.75	0.09	0.9131	2.71E-04
CHHO5	7.1	0.21697	0.78483	2.34E-03	CHHO5	7.65	0.10271	0.9008	2.43E-03
CHHO6	6.3332	0.21427	0.78716	3.96E-03	CHHO6	7.1	0.09299	0.91029	1.68E-04
CHHO7	6.4237	<b>0.16456</b>	<b>0.83729</b>	1.60E-04	CHHO7	6.7724	0.09878	0.90423	3.33E-04
CHHO8	6.65	0.2067	0.79494	2.92E-03	CHHO8	6.75	0.10474	0.89822	6.19E-04
CHHO9	6.5	0.20172	0.79989	3.64E-03	CHHO9	<b>6.55</b>	0.11635	0.88636	2.03E-03
CHHO10	6.4	0.18115	0.82061	1.17E-03	CHHO10	6.7098	0.09627	0.90677	1.03E-04



Ds	AFS	AFV	AAC	P-Value	Ds	AFS	AFV	AAC	P-Value
HHO	4.1812	0.0010295	0.99952		HHO	3.1847	0.01207	0.9914	
CHHO1	1.3207	<b>0.0003571</b>	<u>1</u>	8E-09	CHHO1	3.2	0.01913	0.98427	<b>2.81E-01</b>
CHHO2	<b>1.068</b>	<b>0.0003571</b>	<u>1</u>	8E-09	CHHO2	2.9704	0.01956	0.98355	<b>7.55E-02</b>
CHHO3	1.6571	<b>0.0003571</b>	<u>1</u>	8E-09	CHHO3	2.9098	0.01807	0.985	4.30E-02
CHHO4	1.4191	<b>0.0003571</b>	<u>1</u>	8E-09	CHHO4	3.4466	0.01733	0.98637	<b>3.42E-01</b>
CHHO5	1.5753	<b>0.0003571</b>	<u>1</u>	7.99E-09	CHHO5	3.1	0.01618	0.98713	2.40E-02
CHHO6	1.1926	<b>0.0003571</b>	<u>1</u>	7.98E-09	CHHO6	3.3748	<b>0.00934</b>	<b>0.99427</b>	<b>9.89E-02</b>
CHHO7	1.1539	<b>0.0003571</b>	<u>1</u>	7.95E-09	CHHO7	3.2341	0.01488	0.98856	<b>2.62E-01</b>
CHHO8	1.1138	<b>0.0003571</b>	<u>1</u>	2.96E-08	CHHO8	3.102	0.01123	0.99213	<b>1.81E-01</b>
CHHO9	1.3018	<b>0.0003571</b>	<u>1</u>	8.01E-09	CHHO9	<b>2.8838</b>	0.01313	0.98999	<b>1.12E-01</b>
CHHO10	1.5811	<b>0.0003571</b>	<u>1</u>	8.01E-09	CHHO10	3.1043	0.01474	0.98853	<b>1.18E-01</b>
Ds7	AFS	AFV	AAC	P-Value	Ds8	AFS	AFV	AAC	P-Value
HHO	5.1684	0.017239	0.98401		HHO	8.1129	0.02699	0.97438	
CHHO1	2.6697	0.010757	0.99035	1.32E-07	CHHO1	4.0124	0.02017	0.98073	9.05E-07
CHHO2	2.9573	0.0099321	0.9913	9.07E-08	CHHO2	4.0024	0.01377	0.98719	1.06E-03
CHHO3	2.9723	0.010685	0.99058	3.15E-07	CHHO3	4.4384	0.01679	0.98428	2.31E-04
CHHO4	2.9976	0.010185	0.99107	6.69E-08	CHHO4	4.6151	0.01629	0.9848	1.22E-04
CHHO5	3.1431	0.0090956	<b>0.99224</b>	1.35E-06	CHHO5	4.8071	0.01679	0.98428	1.28E-03
CHHO6	<b>2.2474</b>	<b>0.0086811</b>	<b>0.99224</b>	5.12E-07	CHHO6	4.4496	0.01797	0.98312	1.23E-02
CHHO7	3.2293	0.012643	0.9887	6.64E-08	CHHO7	<b>3.9546</b>	<b>0.01259</b>	<b>0.98835</b>	2.24E-05
CHHO8	3.1054	0.0097613	0.99154	1.26E-06	CHHO8	4.6576	0.01498	0.98603	1.31E-05
CHHO9	2.7568	0.011028	0.99012	5.96E-07	CHHO9	4.1235	0.01378	0.98719	9.21E-05
CHHO10	2.4241	0.01018	0.99082	1.04E-06	CHHO10	4.6047	0.01506	0.98602	1.22E-03
Ds9	AFS	AFV	AAC	P-Value	Ds10	AFS	AFV	AAC	P-Value
HHO	13.172	0.0076908	0.99587		HHO	8.6004	0.65838	0.33768	
CHHO1	6.0566	0.0018235	<u>1</u>	1.83E-07	CHHO1	7.8372	0.63337	0.36294	4.54E-06
CHHO2	<b>5.6234</b>	<b>0.0016912</b>	<u>1</u>	5.00E-08	CHHO2	<b>5.8621</b>	0.63074	0.365	1.44E-04
CHHO3	6.6319	0.0019706	<u>1</u>	6.48E-07	CHHO3	7.5283	0.62924	0.36709	2.69E-06
CHHO4	6.3803	0.0019412	<u>1</u>	9.91E-08	CHHO4	8.3032	<b>0.62909</b>	<b>0.36744</b>	3.50E-06
CHHO5	7.4578	0.0035143	0.99865	4.58E-05	CHHO5	7.6576	0.62988	0.36627	3.29E-05
CHHO6	6.9455	0.0020441	<u>1</u>	3.35E-07	CHHO6	8.064	0.63956	0.35679	2.05E-06
CHHO7	6.6672	0.0019853	<u>1</u>	1.34E-07	CHHO7	8.4373	0.63002	0.36653	3.10E-05
CHHO8	5.7319	<b>0.0016912</b>	<u>1</u>	2.79E-07	CHHO8	7.1586	0.62991	0.36632	2.35E-06
CHHO9	6.1003	0.0018088	<u>1</u>	2.21E-07	CHHO9	6.5711	0.63857	0.35718	2.59E-04
CHHO10	5.7329	0.0017353	<u>1</u>	4.68E-07	CHHO10	7.2055	0.63207	0.36414	9.25E-05
Ds11	AFS	AFV	AAC	P-Value	Ds12	AFS	AFV	AAC	P-Value
HHO	3.6571	0.20995	0.79136		HHO	8.0716	0.17972	0.82187	
CHHO1	3.2797	<b>0.19004</b>	0.81132	2.06E-02	CHHO1	8.676	0.08721	0.91588	5.16E-05
CHHO2	3.2373	0.19736	0.80383	2.55E-02	CHHO2	8.1229	0.08167	0.92123	9.61E-04
CHHO3	3.85	0.19051	<b>0.81146</b>	<b>1.23E-01</b>	CHHO3	8.3	0.07041	0.93269	8.70E-05
CHHO4	3.1076	0.19427	0.8069	3.31E-03	CHHO4	7.8	0.07406	0.92877	9.17E-04
CHHO5	3.0484	0.20188	0.79906	1.11E-03	CHHO5	7.6372	0.07961	0.92308	2.88E-04
CHHO6	<b>2.9312</b>	0.19675	0.80419	<b>6.37E-02</b>	CHHO6	7.75	0.10443	0.89808	8.66E-03
CHHO7	3.2705	0.19012	0.81119	3.34E-02	CHHO7	<b>7.4839</b>	<b>0.05735</b>	<b>0.94551</b>	8.73E-05
CHHO8	3.2	0.20835	0.79278	9.77E-03	CHHO8	8.4185	0.06886	0.93426	1.43E-04
CHHO9	2.9511	0.19292	0.80811	<b>8.81E-02</b>	CHHO9	8.7596	0.08709	0.91603	6.81E-04
CHHO10	3.3273	0.1972	0.80419	1.14E-02	CHHO10	8.8207	0.07639	0.92685	3.18E-03
Ds13	AFS	AFV	AAC	P-Value	Ds14	AFS	AFV	AAC	P-Value
HHO	6.6633	0.032007	0.96924		HHO	3.1587	0.09683	0.855	
CHHO1	4.023	0.017629	0.98327	2.85E-04	CHHO1	2.4689	0.05206	0.75	<b>2.93E-01</b>
CHHO2	<b>3.626</b>	0.01937	0.98158	8.68E-04	CHHO2	2.4992	0.04221	0.81	<b>2.01E-01</b>
CHHO3	4.5233	0.023808	0.97713	1.28E-03	CHHO3	2.0329	0.03192	0.67	<b>4.81E-01</b>
CHHO4	4.2427	0.020486	0.98067	1.19E-05	CHHO4	1.946	0.0416	0.61	<b>4.54E-01</b>
CHHO5	4.0011	<b>0.013357</b>	<b>0.9877</b>	1.67E-04	CHHO5	1.5816	<b>0.03131</b>	0.52	<b>6.59E-01</b>
CHHO6	4.5712	0.020485	0.98065	2.12E-03	CHHO6	2.6191	0.08203	<b>0.87</b>	<b>1.24E-01</b>
CHHO7	4.0708	0.01858	0.98239	7.83E-04	CHHO7	1.4541	0.04616	0.605	<b>5.98E-01</b>

CHHO8	4.2592	0.017682	0.98332	1.27E-02	CHHO8	<b>1.25</b>	0.03604	0.515	<b>6.18E-01</b>
CHHO9	4.1104	0.018585	0.98246	9.57E-04	CHHO9	2.3432	0.06685	0.735	<b>2.76E-01</b>
CHHO10	3.9305	0.021306	0.97981	1.08E-05	CHHO10	2.3605	0.05194	0.7	<b>2.23E-01</b>

TABLE VI  
P-VALUE OF CHHO'S AMONG THE DATASETS

	CHHO1	CHHO2	CHHO3	CHHO4	CHHO5	CHHO6	CHHO7	CHHO8	CHHO9	CHHO10
Ds1	2.19E-01	1.14E-01	9.08E-01	3.17E-01	7.25E-01	6.31E-01	4.57E-01	1.25E-01	1.66E-01	5.12E-01
Ds2	7.81E-02	<u>1.04E-02</u>	8.29E-02	1.10E-01	<u>3.10E-02</u>	7.31E-02	1.54E-01	2.00E-01	9.19E-02	<u>5.28E-02</u>
Ds3	<u>3.97E-02</u>	<u>4.10E-02</u>	<u>1.95E-03</u>	<u>4.31E-03</u>	<u>1.19E-02</u>	<u>1.43E-02</u>	<u>1.48E-03</u>	<u>7.40E-03</u>	<u>1.93E-02</u>	<u>1.79E-02</u>
Ds4	<u>1.60E-02</u>	<u>1.69E-03</u>	<u>1.67E-03</u>	<u>1.50E-04</u>	<u>1.60E-02</u>	<u>2.53E-03</u>	<u>1.09E-02</u>	<u>3.04E-03</u>	<u>2.30E-02</u>	<u>6.56E-05</u>
Ds5	<u>9.11E-04</u>	1.62E-01	<u>9.54E-03</u>	<u>1.97E-02</u>	<u>4.51E-03</u>	<u>4.49E-03</u>	8.06E-02	<u>1.96E-02</u>	<u>2.07E-03</u>	8.06E-02
Ds6	4.94E-01	6.74E-01	3.07E-01	4.71E-01	2.82E-01	5.47E-01	6.21E-01	3.82E-01	3.91E-01	5.31E-01
Ds7	0.13286	<u>2.21E-02</u>	<u>1.09E-02</u>	<u>1.48E-02</u>	1.19E-01	<u>4.01E-02</u>	6.16E-02	1.01E-01	1.51E-01	8.54E-02
Ds8	6.57E-02	<u>6.26E-03</u>	<u>2.55E-02</u>	8.79E-02	1.37E-01	<u>4.97E-02</u>	<u>3.54E-02</u>	6.55E-02	<u>5.99E-02</u>	<u>5.12E-02</u>
Ds9	<u>8.31E-07</u>	<u>5.00E-08</u>	<u>5.68E-06</u>	<u>1.95E-06</u>	<u>5.58E-05</u>	<u>5.48E-07</u>	<u>2.15E-07</u>	<u>3.64E-07</u>	<u>6.42E-07</u>	<u>1.14E-06</u>
Ds10	<u>2.30E-02</u>	7.42E-02	<u>8.35E-04</u>	<u>2.67E-03</u>	<u>1.95E-03</u>	<u>6.23E-05</u>	<u>5.11E-03</u>	<u>2.67E-03</u>	<u>5.55E-03</u>	<u>4.90E-03</u>
Ds11	3.30E-01	1.26E-01	4.56E-01	4.89E-01	2.03E-01	3.04E-01	4.09E-01	1.89E-01	3.57E-01	5.42E-01
Ds12	<u>1.60E-04</u>	<u>5.83E-04</u>	<u>6.20E-05</u>	<u>2.72E-04</u>	<u>3.20E-04</u>	<u>3.33E-03</u>	<u>1.41E-05</u>	<u>2.73E-05</u>	<u>4.13E-04</u>	<u>7.94E-04</u>
Ds13	<u>4.05E-02</u>	<u>1.37E-02</u>	6.34E-02	<u>2.35E-02</u>	<u>9.95E-04</u>	<u>2.73E-02</u>	<u>2.28E-02</u>	<u>3.95E-02</u>	<u>4.49E-02</u>	<u>4.35E-02</u>
Ds14	6.42E-01	8.05E-01	8.69E-01	9.22E-01	8.42E-01	5.18E-01	8.89E-01	7.43E-01	8.38E-01	7.94E-01

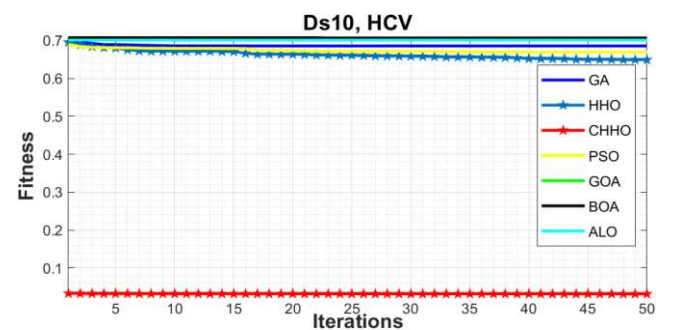
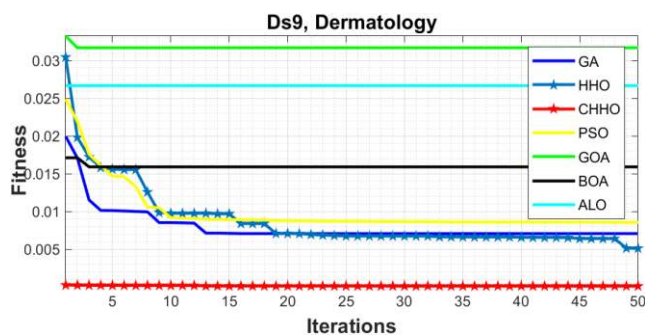
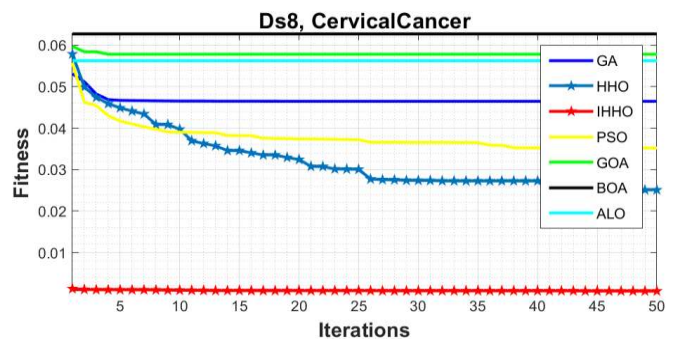
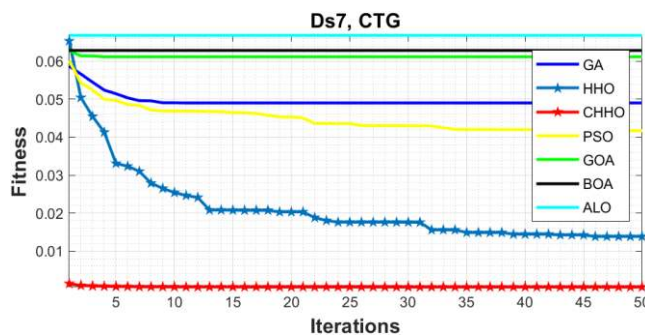
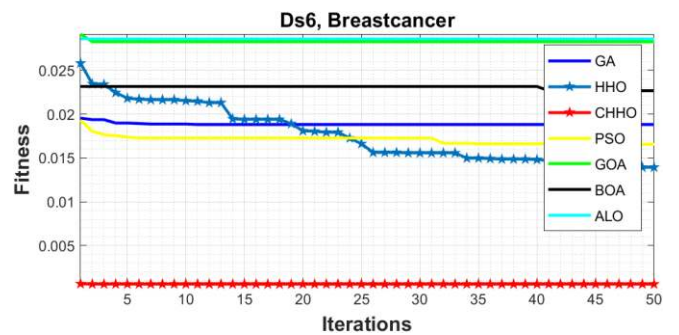
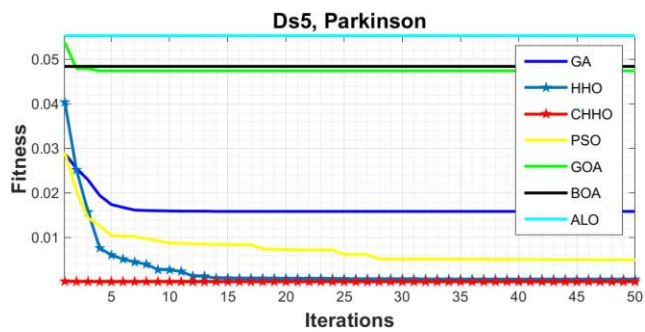
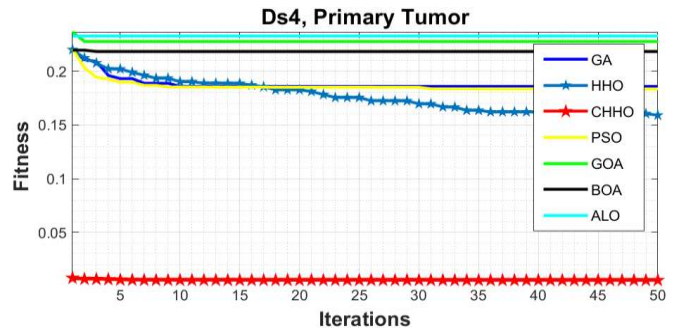
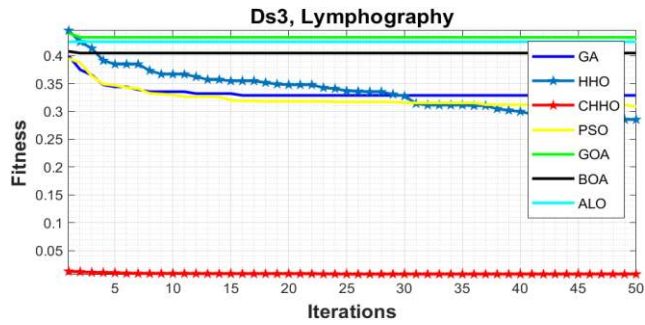
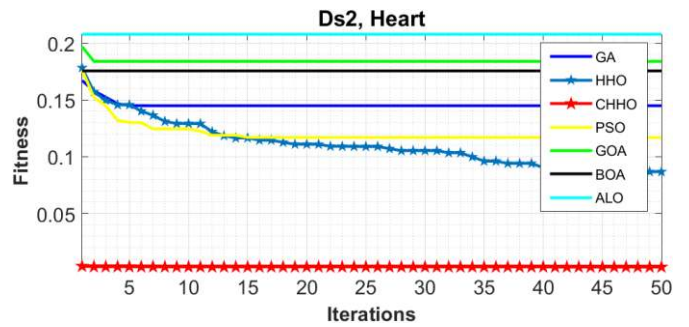
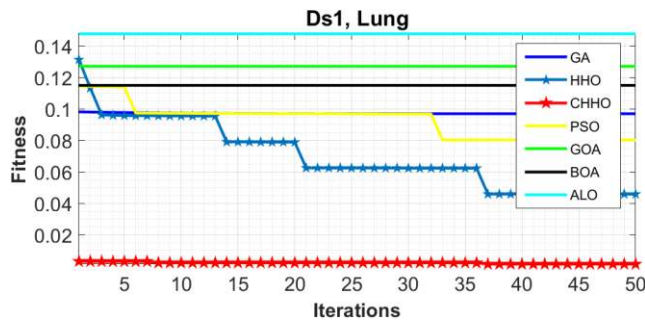
Generally, most of the CHHO variants with the chaotic maps produced better solutions than the standard HHO in all metrics. This shows the importance and effect of employing a chaotic map and SA to improve the population diversity and enhance the local search. However, in most cases, the worst results were found by the CHHO7 (Piecewise map) variant, which makes it incompatible with the search mechanism the standard HHO algorithm. Based on the reported results in Tables V and VI, it can be concluded that the Sine chaotic map seems to be a proper choice to enhance the performance of the standard HHO algorithm and customize it for the feature selection problem. It is worth mentioning that the highest classification performance results were obtained on Ds 5. This is due to the large number of experiments required from a small number of patients to identify Parkinson's disease. In the following section, Sine chaotic map (CHHO6) has been selected for further investigation along with other states of the art algorithms to verify the performance of the CHHO in the feature selection problem.

## 2) COMPARISON OF THE PROPOSED CHHO WITH STATE OF THE ART ALGORITHMS

The second experiment in this study includes the comparison of CHHO performance with other optimization algorithms. The baseline algorithms are GOA, GA, PSO, BOA, ALO, and HHO. The parameter settings for all algorithms are shown in Table IV were the maximum iterations and the search-agents set to 50 and 10, respectively, for all algorithms. The performance score was calculated based on 20 runs. Table VII. shows the number of selected features in all evaluated algorithms. It is observed that the CHHO

achieved the best results of selected features in 11 datasets, while HHO succeeded in two datasets and GA in one dataset. In terms of classification accuracy presented in Table VIII, it is observed that the CHHO obtained the best results in most of the cases. Still, it gave similar classification accuracy to the HHO in four datasets. Also, PSO provided similar classification accuracy in two datasets. In the second place, PSO and HHO obtained similar classification accuracy in most cases, and GA comes in third place. However, in Table IX CHHO algorithm outperformed all other algorithms in terms of fitness value, considering it attained the minimum classification error among all the algorithms.

Graphical representation of the convergence-curves was also considered to evaluate the convergence speed of CHHO on 14 benchmark datasets as displayed in Figure 2. From Figure 2, it is observed that the CHHO algorithm achieved higher performance results on 13 datasets while it is comparable with standard HHO in Ds5. Also, it is observed that the performance of HHO is comparable with the PSO in most cases where the ALO achieved the worst convergence speed. PSO algorithm is considered to be the second efficient method after HHO in all benchmark datasets. In other words, CHHO has a higher converged rate and lower classification error than the different competing algorithms. This superiority came from the improvement stated in the initialization and exploitation phases. The enhanced population diversity in the initialization phase leads to accelerate the convergence speed. Also, the enhanced in the exploitation phase provided high fitness value. These superiority results are a clue of the higher algorithm capability to avoid the local optima problem and solve the problem of feature selection.





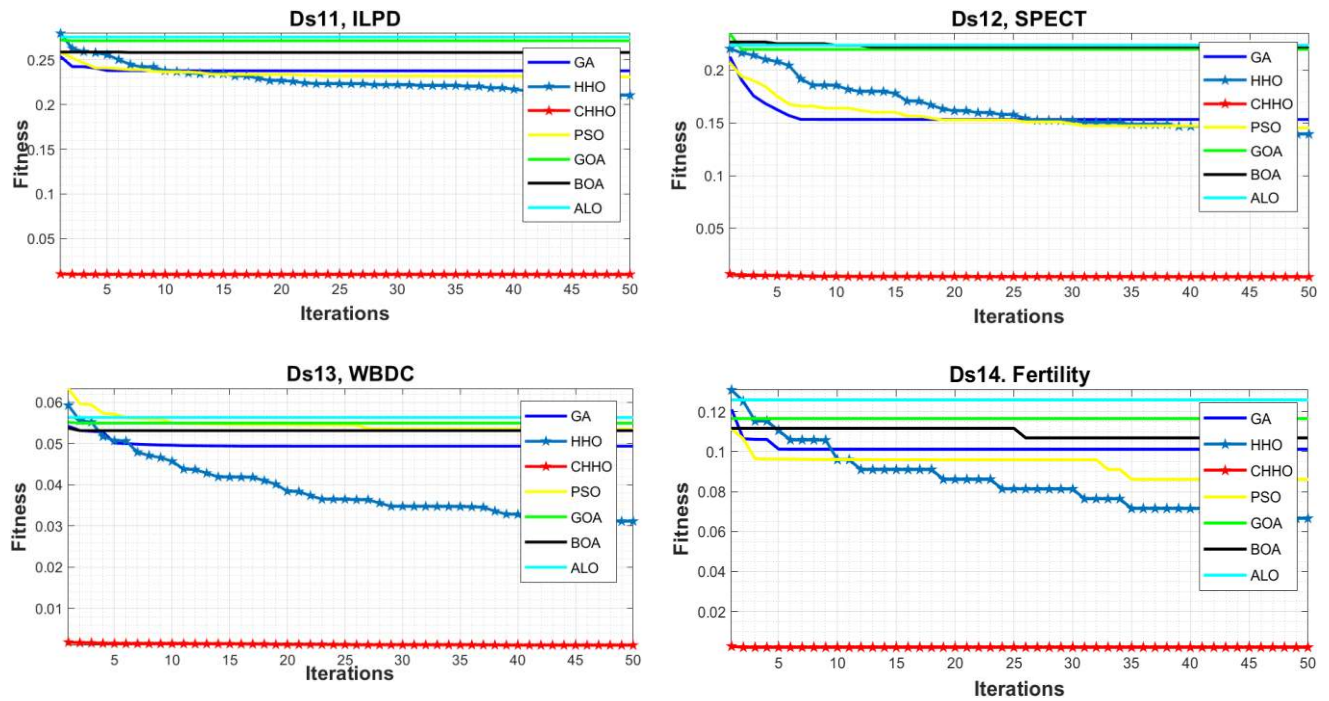


FIGURE 2. Convergence-curves of the competing algorithms on all datasets.

TABLE VII  
AVERAGE NUMBER OF SELECTED FEATURE OF CHHO IN THE COMPARISON WITH ALL OPTIMIZATION ALGORITHMS

	GOA	GA	PSO	BOA	ALO	HHO	CHHO
DS1	16.13	11.25	14.49	15.70	15.31	4.764	<u>2.5185</u>
DS2	5.87	5.50	5.50	6.30	5.92	<u>3.7082</u>	4.1806
DS3	9.11	8.95	8.48	10.65	8.88	6.6012	<u>6.3332</u>
DS4	8.26	7.90	8.53	10.75	8.19	7.2008	<u>7.1</u>
DS5	13.57	9.30	12.43	16.00	13.30	4.1812	<u>1.1926</u>
DS6	4.34	3.35	4.17	5.60	4.49	<u>3.1847</u>	3.3748
DS7	10.93	8.65	10.10	12.60	10.83	5.1684	<u>2.2474</u>
DS8	16.45	13.25	15.45	16.20	16.20	8.1129	<u>4.4496</u>
DS9	16.78	15.15	16.06	23.05	17.38	13.1718	<u>6.9455</u>
DS10	13.23	10.35	12.91	17.90	13.86	8.6004	<u>8.064</u>
DS11	4.92	4.00	4.38	5.05	5.08	3.6571	<u>2.9312</u>
DS12	10.72	9.80	10.40	13.95	10.88	8.0716	<u>7.75</u>
DS13	13.62	9.55	13.29	15.65	14.73	6.6633	<u>4.5712</u>
DS14	3.64	<u>2.10</u>	3.43	3.25	3.28	3.1587	2.6191

TABLE VIII  
AVERAGE CLASSIFICATION ACCURACY OF CHHO IN THE COMPARISON WITH ALL OPTIMIZATION ALGORITHMS

	GOA	GA	PSO	BOA	ALO	HHO	CHHO
DS1	0.55	0.58	0.57	0.55	0.55	<u>0.60</u>	<u>0.60</u>
DS2	0.83	0.85	0.88	0.80	0.81	0.90	<u>0.93</u>
DS3	0.56	0.63	0.66	0.57	0.52	0.68	<u>0.79</u>
DS4	0.79	0.84	0.85	0.81	0.80	0.87	<u>0.91</u>
DS5	0.95	0.99	<u>1.00</u>	0.96	0.95	<u>1.00</u>	<u>1.00</u>
DS6	0.98	<u>0.99</u>	<u>0.99</u>	<u>0.99</u>	0.98	<u>0.99</u>	<u>0.99</u>
DS7	0.94	0.95	0.96	0.94	0.93	0.98	<u>0.99</u>



DS8	0.95	0.97	0.97	0.95	0.94	0.97	<u>0.98</u>
DS9	0.97	0.99	0.99	0.97	0.95	<u>1.00</u>	<u>1.00</u>
DS10	0.30	0.32	0.33	0.30	0.30	0.34	<u>0.36</u>
DS11	0.73	0.76	0.78	0.75	0.73	0.79	<u>0.80</u>
DS12	0.75	0.80	0.83	0.77	0.74	0.82	<u>0.90</u>
DS13	0.95	0.96	0.96	0.95	0.94	0.97	<u>0.98</u>
DS14	0.80	0.82	0.82	0.82	0.79	0.86	<u>0.87</u>

TABLE IX  
AVERAGE FITNESS VALUE OF CHHO IN THE COMPARISON WITH ALL OPTIMIZATION ALGORITHMS

	GOA	GA	PSO	BOA	ALO	HHO	CHHO
DS1	0.0521	0.0185	0.0343	0.0523	0.0520	0.0004	<u>0.0002</u>
DS2	0.1761	0.1491	0.1221	0.2047	0.1966	0.1052	<u>0.0728</u>
DS3	0.4369	0.3680	0.3410	0.4334	0.4766	0.3222	<u>0.2143</u>
DS4	0.2130	0.1604	0.1545	0.1927	0.2043	0.1365	<u>0.0930</u>
DS5	0.0492	0.0095	0.0062	0.0409	0.0558	0.0010	<u>0.0004</u>
DS6	0.0215	0.0179	0.0155	0.0197	0.0226	0.0121	<u>0.0093</u>
DS7	0.0659	0.0547	0.0427	0.0625	0.0698	0.0172	<u>0.0087</u>
DS8	0.0567	0.0372	0.0284	0.0553	0.0626	0.0270	<u>0.0180</u>
DS9	0.0359	0.0193	0.0103	0.0368	0.0579	0.0077	<u>0.0020</u>
DS10	0.6983	0.6780	0.6712	0.6960	0.6966	0.6584	<u>0.6396</u>
DS11	0.2689	0.2395	0.2217	0.2567	0.2763	0.2100	<u>0.1968</u>
DS12	0.2563	0.2012	0.1675	0.2323	0.2604	0.1797	<u>0.1044</u>
DS13	0.0526	0.0423	0.0434	0.0513	0.0594	0.0320	<u>0.0205</u>
DS14	0.1568	0.1310	0.1359	0.1323	0.1658	0.0968	<u>0.0820</u>

Precisely, the proposed CHHO framework succeeded in balancing the search process among the exploration and exploitation over the search iterations.

### 3) THE LIMITATIONS OF CHHO ALGORITHM

The proposed CHHO is a promising algorithm that can solve high dimensional and complex optimization problems. CHHO improved the standard HHO in different aspects, such as the reduction of selected features, increasing classification accuracy, and fitness values. However, similar to other optimization algorithms, CHHO also has some limitations. The primary limitation is that it is comparatively time-consuming in comparison to different algorithms. However, the reason for the time-consumption is the computational complexity of the standard HHO, not because of the proposed improvements. Also, we believe that the time-consumption could be decreased if we reduced ten iterations of SA.

## VI. CONCLUSION AND FUTURE DIRECTIONS

In this study, an improved CHHO algorithm is proposed by including chaotic maps to the HHO algorithm at the initialization phase and including the SA algorithm to the exploitation phase. Ten different chaotic maps were tested to determine the best compatible choice with the HHO algorithm to enhance the population diversity and improve the convergence speed. Furthermore, the SA algorithm was

employed to improve the exploitation phase, which avoids the local optima problem. The proposed framework CHHO was applied for the feature selection problem. Fourteen medical benchmark datasets from the UCI machine learning repository were selected for the experiments along with Five evaluation criteria. These criteria are the number of selected features, classification accuracy, fitness value, *P*-value, and convergence speed.

Additionally, the performance of the CHHO was compared with other recent and famous optimization algorithms. These algorithms are GOA, GA, PSO, BOA, ALO, and original HHO. The experimental and evaluation results demonstrated the superiority of the CHHO in comparison with other optimization algorithms in all metrics. Moreover, the results showed that the CHHO with the Sine map could significantly improve the performance of the standard HHO in terms of classification performance, the number of selected features, and convergence rates. Also, the results showed that applying the SA algorithm in the exploitation phase enhanced local search. The modifications achieved a balanced search behavior and suggested that the proposed framework is convenient for medical applications. For future research, it could be attractive to investigate the performance of the proposed CHHO algorithm on more sophisticated science and engineering problems and further enhance its complexity without affecting the current performance.

## ACKNOWLEDGMENT

The contributors would like to acknowledge the editors, reviewers, and Dr. Norizan Mohd Yasin for their valuable knowledge and comments. By the University of Malaya (UM).

## REFERENCES

- [1] M. Mafarja and S. Mirjalili, "Whale optimization approaches for wrapper feature selection," *Appl. Soft Comput. J.*, vol. 62, pp. 441–453, 2018, doi: 10.1016/j.asoc.2017.11.006.
- [2] Q. Wu, Z. Ma, J. Fan, G. Xu, and Y. Shen, "A Feature Selection Method Based on Hybrid Improved Binary Quantum Particle Swarm Optimization," *IEEE Access*, vol. 7, pp. 80588–80601, 2019, doi: 10.1109/ACCESS.2019.2919956.
- [3] Z. M. Hira and D. F. Gillies, "A review of feature selection and feature extraction methods applied on microarray data," *Adv. Bioinformatics*, vol. 2015, no. 1, 2015, doi: 10.1155/2015/198363.
- [4] N. Almgren and H. Alshamlan, "A survey on hybrid feature selection methods in microarray gene expression data for cancer classification," *IEEE Access*, vol. 7, pp. 78533–78548, 2019, doi: 10.1109/ACCESS.2019.2922987.
- [5] A. M. Anter and M. Ali, "Feature selection strategy based on hybrid crow search optimization algorithm integrated with chaos theory and fuzzy c-means algorithm for medical diagnosis problems," *Soft Comput.*, vol. 24, no. 3, pp. 1565–1584, 2020, doi: 10.1007/s00500-019-03988-3.
- [6] Z. Tao, L. Huiling, W. Wenwen, and Y. Xia, "GA-SVM based feature selection and parameter optimization in hospitalization expense modeling," *Appl. Soft Comput. J.*, vol. 75, pp. 323–332, 2019, doi: 10.1016/j.asoc.2018.11.001.
- [7] M. Tubishat, N. Idris, L. Shuib, M. A. M. Abushariah, and S. Mirjalili, "Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection," *Expert Syst. Appl.*, vol. 145, p. 113122, 2020, doi: 10.1016/j.eswa.2019.113122.
- [8] P. Moradi and M. Gholampour, "A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy," *Appl. Soft Comput.*, vol. 43, pp. 117–130, Jun. 2016, doi: 10.1016/J.ASOC.2016.01.044.
- [9] Q. Al-Tashi, S. J. Abdul 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, doi: 10.1109/ACCESS.2019.2906757.
- [10] S. Arora and P. Anand, "Binary butterfly optimization approaches for feature selection," *Expert Syst. Appl.*, vol. 116, pp. 147–160, 2019, doi: 10.1016/j.eswa.2018.08.051.
- [11] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary ant lion approaches for feature selection," *Neurocomputing*, vol. 213, pp. 54–65, 2016, doi: 10.1016/j.neucom.2016.03.101.
- [12] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, 2016, doi: 10.1016/j.neucom.2015.06.083.
- [13] M. Taradeh *et al.*, "An evolutionary gravitational search-based feature selection," *Inf. Sci. (Nij.)*, vol. 497, pp. 219–239, Sep. 2019, doi: 10.1016/J.INS.2019.05.038.
- [14] S. Arora and S. Singh, "Butterfly optimization algorithm: a novel approach for global optimization," *Soft Comput.*, vol. 23, no. 3, pp. 715–734, 2019, doi: 10.1007/s00500-018-3102-4.
- [15] S. Z. Mirjalili, S. Mirjalili, S. Saremi, H. Faris, and I. Aljarah, "Grasshopper optimization algorithm for multi-objective optimization problems," *Appl. Intell.*, vol. 48, no. 4, pp. 805–820, 2018, doi: 10.1007/s10489-017-1019-8.
- [16] M. Tubishat, M. A. M. 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, 2019, doi: 10.1007/s10489-018-1334-8.
- [17] M. Alswaitti, M. Albughdadi, and N. A. Mat Isa, "Variance-based differential evolution algorithm with an optional crossover for data clustering," *Appl. Soft Comput. J.*, vol. 80, pp. 1–17, 2019, doi: 10.1016/j.asoc.2019.03.013.
- [18] H. J. Lu, H. M. Zhang, and L. H. Ma, "New optimization algorithm based on chaos," *J. Zhejiang Univ. Sci.*, vol. 7, no. 4, pp. 539–542, 2006, doi: 10.1631/jzus.2006.A0539.
- [19] G. I. Sayed, A. E. Hassanien, and A. T. Azar, "Feature selection via a novel chaotic crow search algorithm," *Neural Comput. Appl.*, vol. 31, no. 1, pp. 171–188, 2019, doi: 10.1007/s00521-017-2988-6.
- [20] G. Kaur and S. Arora, "Chaotic whale optimization algorithm," *J. Comput. Des. Eng.*, vol. 5, no. 3, pp. 275–284, Jul. 2018, doi: 10.1016/J.JCDE.2017.12.006.
- [21] M. Yousefi, M. Yousefi, R. P. M. Ferreira, J. H. Kim, and F. S. Fogliatto, "Chaotic genetic algorithm and Adaboost ensemble metamodeling approach for optimum resource planning in emergency departments," *Artif. Intell. Med.*, vol. 84, pp. 23–33, 2018, doi: 10.1016/j.artmed.2017.10.002.
- [22] M. Kohli and S. Arora, "Chaotic grey wolf optimization algorithm for constrained optimization problems," *J. Comput. Des. Eng.*, vol. 5, no. 4, pp. 458–472, 2018, doi: 10.1016/j.jcde.2017.02.005.
- [23] S. Arora and P. Anand, "Chaotic grasshopper optimization algorithm for global optimization,"

- Neural Comput. Appl.*, vol. 31, no. 8, pp. 4385–4405, 2019, doi: 10.1007/s00521-018-3343-2.
- [24] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science* (80-. ), vol. 220, no. 4598, pp. 671–680, 1983, doi: 10.1126/science.220.4598.671.
- [25] M. M. Mafarja and S. Mirjalili, “Hybrid Whale Optimization Algorithm with simulated annealing for feature selection,” *Neurocomputing*, vol. 260, pp. 302–312, 2017, doi: 10.1016/j.neucom.2017.04.053.
- [26] J. Wang, B. Zhou, and S. Zhou, “An Improved Cuckoo Search Optimization Algorithm for the Problem of Chaotic Systems Parameter Estimation,” *Comput. Intell. Neurosci.*, vol. 2016, no. 1, 2016, doi: 10.1155/2016/2959370.
- [27] B. Wei, W. Zhang, X. Xia, Y. Zhang, F. Yu, and Z. Zhu, “Efficient Feature Selection Algorithm Based on Particle Swarm Optimization with Learning Memory,” *IEEE Access*, vol. 7, pp. 166066–166078, 2019, doi: 10.1109/ACCESS.2019.2953298.
- [28] 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, 2019, doi: 10.1515/jisys-2019-0062.
- [29] P. Ghamisi and J. A. Benediktsson, “Feature selection based on hybridization of genetic algorithm and particle swarm optimization,” *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 2, pp. 309–313, 2015, doi: 10.1109/LGRS.2014.2337320.
- [30] L. Wang, H. Zhen, X. Fang, S. Wan, W. Ding, and Y. Guo, “A unified two-parallel-branch deep neural network for joint gland contour and segmentation learning,” *Futur. Gener. Comput. Syst.*, vol. 100, pp. 316–324, 2019, doi: 10.1016/j.future.2019.05.035.
- [31] R. Hussein, H. Palangi, R. K. Ward, and Z. J. Wang, “Optimized deep neural network architecture for robust detection of epileptic seizures using EEG signals,” *Clin. Neurophysiol.*, vol. 130, no. 1, pp. 25–37, 2019, doi: 10.1016/j.clinph.2018.10.010.
- [32] H. Guo, J. Zhou, M. Koopialipoor, D. Jahed Armaghani, and M. M. Tahir, “Deep neural network and whale optimization algorithm to assess flyrock induced by blasting,” *Eng. Comput.*, no. 0123456789, 2019, doi: 10.1007/s00366-019-00816-y.
- [33] F. Pourpanah, Y. Shi, C. P. Lim, Q. Hao, and C. J. Tan, “Feature selection based on brain storm optimization for data classification,” *Appl. Soft Comput. J.*, vol. 80, pp. 761–775, 2019, doi: 10.1016/j.asoc.2019.04.037.
- [34] A. E. Hassanien, T. Gaber, U. Mokhtar, and H. Hefny, “An improved moth flame optimization algorithm based on rough sets for tomato diseases detection,” *Comput. Electron. Agric.*, vol. 136, pp. 86–96, 2017, doi: 10.1016/j.compag.2017.02.026.
- [35] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, “Harris hawks optimization: Algorithm and applications,” *Futur. Gener. Comput. Syst.*, vol. 97, pp. 849–872, 2019, doi: 10.1016/j.future.2019.02.028.
- [36] X. Bao, H. Jia, and C. Lang, “A Novel Hybrid Harris Hawks Optimization for Color Image Multilevel Thresholding Segmentation,” *IEEE Access*, vol. 7, pp. 76529–76546, 2019, doi: 10.1109/ACCESS.2019.2921545.
- [37] H. R. E. H. Bouchekara, A. E. Chaib, M. A. Abido, and R. A. El-Sehiemy, “Optimal power flow using an Improved Colliding Bodies Optimization algorithm,” *Appl. Soft Comput. J.*, vol. 42, pp. 119–131, 2016, doi: 10.1016/j.asoc.2016.01.041.
- [38] 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, p. 106656, 2020, doi: 10.1016/j.compchemeng.2019.106656.
- [39] R. Sihwail, K. Omar, K. A. Z. Ariffin, and M. Tubishat, “Improved Harris Hawks Optimization Using Elite Opposition-Based Learning and Novel Search Mechanism for Feature Selection,” *IEEE Access*, vol. 8, pp. 121127–121145, 2020, doi: 10.1109/ACCESS.2020.3006473.
- [40] T. Singh, “A chaotic sequence-guided Harris hawks optimizer for data clustering,” *Neural Comput. Appl.*, vol. 2, 2020, doi: 10.1007/s00521-020-04951-2.
- [41] A. A. Ewees and M. A. Elaziz, “Performance analysis of Chaotic Multi-Verse Harris Hawks Optimization: A case study on solving engineering problems,” *Eng. Appl. Artif. Intell.*, vol. 88, p. 103370, Feb. 2020, doi: 10.1016/J.ENGAPAI.2019.103370.
- [42] A. S. Menesy, H. M. Sultan, A. Selim, M. G. Ashmawy, and S. Kamel, “Developing and Applying Chaotic Harris Hawks Optimization Technique for Extracting Parameters of Several Proton Exchange Membrane Fuel Cell Stacks,” *IEEE Access*, vol. 8, pp. 1146–1159, 2020, doi: 10.1109/ACCESS.2019.2961811.
- [43] N. S. Altman, “An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression Author ( s ): N . S . Altman Source : The American Statistician , Aug . , 1992 , Vol . 46 , No . 3 ( Aug . , 1992 ) , pp . 175-185 Published by : Taylor & Francis , Ltd . on behalf of th,” vol. 46, no. 3, pp. 175–185, 1992.
- [44] S. Saremi, S. Mirjalili, and A. Lewis, “Grasshopper Optimisation Algorithm: Theory and application,” *Adv. Eng. Softw.*, vol. 105, pp. 30–47, 2017, doi:

- 10.1016/j.advengsoft.2017.01.004.
- [45] S. Mirjalili, J. S. Dong, A. Lewis, and L. Reviews, *Nature-Inspired Optimizers*.
- [46] M. Alweshah, S. Al Khalailah, B. B. Gupta, A. Almomani, A. I. Hammouri, and M. A. Al-Betar, "The monarch butterfly optimization algorithm for solving feature selection problems," *Neural Comput. Appl.*, vol. 0, 2020, doi: 10.1007/s00521-020-05210-0.
- [47] M. Hosseini Rad and M. Abdolrazzagah-Nezhad, "A new hybridization of DBSCAN and fuzzy earthworm optimization algorithm for data cube clustering," *Soft Comput.*, vol. 0, no. Pujari 2001, 2020, doi: 10.1007/s00500-020-04881-0.



ZENAB MOHAMED ELGAMAL born in Cairo, Egypt. Received the M.Sc. degrees in information system from the Faculty of Computer Science and Information Technology in the Arab Academy for Science & Technology. Currently pursuing the Ph.D. degree in Information System & Technology, in the Faculty of Computer Science and

Information Technology, University of Malaya. Kuala Lumpur, Malaysia. From 2011 to 2018, was working as Lecturer in several institutions. The current research interests include Optimization Algorithm., Machine Learning, Feature Selection, and Bioinformatics.



NORIZAN BINTI MOHD YASIN Senior Lecturer Department of Information System, Faculty of Computer Science & Information Technology, University of Malaya, Kuala Lumpur, Malaysia.



MOHAMMAD TUBISHAT received his PhD degree in Computer Science (Artificial Intelligence – Natural Language Processing) from the University of Malaya in 2019. In addition, MSc in Computer and Information Sciences from Yarmouk

University in 2004. Furthermore, BSc degree in Computer Science from Yarmouk University in 2002. His research interests include natural language processing, data mining, artificial intelligence, machine learning, optimization algorithms, data science, and sentiment analysis. He is now working as a lecturer at Asia Pacific University of Technology and innovation.



MOHAMMED ALSWAITTI received the B.Eng. degree in computer engineering from The Islamic University of Gaza (IUG), Palestine, in 2010, and the M.Sc. degree in electronic systems design engineering from Universiti Sains Malaysia (USM), Malaysia, in 2011. He worked as a Lecturer at the University of Palestine

and Instructor at the educational technology center at IUG. He pursued his Ph.D. degree under the Malaysian International Scholarship (MIS) scheme and worked as a Graduate Assistant at the Electrical and Electronic Engineering Department, USM, Malaysia. Currently, he is an assistant professor at the Electrical and Computer Engineering Department, XMUM. His research has been focused on nature-inspired optimization-based clustering techniques, machine learning, and pattern recognition.



SEYEDALI MIRJALILI is an Associate Professor and the Centre for Artificial Intelligence Research and Optimization at Torrens University Australia. He is internationally recognized for his advances in Swarm Intelligence and Optimization,

including the first set of algorithms from a synthetic intelligence standpoint - a radical departure from how natural systems are typically understood - and a systematic design framework to reliably benchmark, evaluate, and propose computationally cheap robust optimization algorithms. Seyedali has published over 200 publications with over 20,000 citations and an H-index of 50. As one of the most cited researcher in Artificial Intelligence, he is in the list of 1% highly-cited researchers and named as one of the most influential researchers in the world by Web of Science. Seyedali is a senior member of IEEE and an associate editor of several journals including Neurocomputing, Applied Soft Computing, Advances in Engineering Software, Applied Intelligence, PLOS One, and IEEE Access. His research interests include Robust Optimization, Machine Learning, Multi-objective Optimization, Swarm Intelligence, Evolutionary Algorithms, Artificial Neural Networks, and applied optimization.



