

# A Chaotic Oppositional Whale Optimisation Algorithm with Firefly Search for Medical Diagnostics

Milan Tair<sup>1</sup>, Nebojsa Bacanin<sup>1</sup>, Miodrag Zivkovic<sup>1</sup> and K. Venkatachalam<sup>2,\*</sup>

<sup>1</sup>Singidunum University, 32 Danijelova Str., 11010, Belgrade, Serbia

<sup>2</sup>Department of Applied Cybernetics, Faculty of Science, University of Hradec Králové, 50003, Hradec Králové, Czech Republic

\*Corresponding Author: K. Venkatachalam. Email: venkatachalam.k@ieee.org

Received: 07 November 2021; Accepted: 10 December 2021

**Abstract:** There is a growing interest in the study development of artificial intelligence and machine learning, especially regarding the support vector machine pattern classification method. This study proposes an enhanced implementation of the well-known whale optimisation algorithm, which combines chaotic and opposition-based learning strategies, which is adopted for hyper-parameter optimisation and feature selection machine learning challenges. The whale optimisation algorithm is a relatively recent addition to the group of swarm intelligence algorithms commonly used for optimisation. The Proposed improved whale optimisation algorithm was first tested for standard unconstrained CEC2017 benchmark suite and it was later adapted for simultaneous feature selection and support vector machine hyper-parameter tuning and validated for medical diagnostics by using breast cancer, diabetes, and erythemato-squamous dataset. The performance of the proposed model is compared with multiple competitive support vector machine models boosted with other metaheuristics, including another improved whale optimisation approach, particle swarm optimisation algorithm, bacterial foraging optimisation algorithms, and genetic algorithms. Results of the simulation show that the proposed model outperforms other competitors concerning the performance of classification and the selected subset feature size.

**Keywords:** Whale optimisation algorithm; chaotic initialisation; opposition-based learning; optimisation; diagnostics

## 1 Introduction

Constructing algorithms to solve non-deterministic polynomial time hard problems (NP-hard) is not typically hard to do. However, executing such an algorithm can, in extreme cases, take thousands of years on contemporary hardware. Such problems are regarded as nearly impossible to solve with traditional approaches for finding a deterministic algorithm. Metaheuristics, as stochastic optimisation approaches, are useful for solving such problems. These approaches do not guarantee an optimum solution. Instead, they provide tolerable sub-optimal outcomes in a reasonable amount of



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

time [1]. The optimisation process aims to discover a solution that is either optimum or near-to-optimal regarding the stated goals and given constraints. Since the turn of the century, many population-based stochastic metaheuristics have been developed to address challenging optimisation issues in both combinatorial and global optimisation [2].

Stochastic population-based metaheuristics begin the search process using a set of random points, but the process is led and directed by some mechanism from iteration to iteration. Natural systems and processes can be used to inspire the concepts that underpin the search mechanism. Some algorithms, such as genetic algorithms (GA) frequently become trapped in local optima. Hybridisation is one method for overcoming the aforementioned difficulties with metaheuristics.

The hybridisation should not be based on a random mix of hybridising algorithms, but rather on a specific analysis of the benefits and drawbacks of each method. Under some situations, the drawback of one algorithm (slow convergence, tendency to quickly become trapped in the local optimum, etc.) is overcome by combining with the structural portion of other algorithms. This can help fix common issues of the original optimal solution search method.

In contrast to deterministic optimisation methods that ensure the optimum answer, heuristic algorithms aim to discover the best possible solution, without ensuring that the optimal solution is obtained [3,4] Many practical and benchmarking problems cannot be solved using deterministic approaches as their solving is dawdling. Since heuristic algorithms are robust, academics worldwide have been interested in their improvement. This has led to the development of metaheuristics for a variety of problems. Metaheuristics is used to create heuristic techniques for solving different kinds of optimisation problems. Two main groups of metaheuristics are those which are inspired and which are not nature-inspired. Metaheuristics inspired by nature can be additionally categorised as evolutionary algorithms (EA) and swarm intelligence. Swarm intelligence has recently become a major research field and they are proven as methods that are able to achieve outstanding results in different fields of optimisation [5–8].

Besides metaheuristics, machine learning is another important domain of artificial intelligence. Notwithstanding good performance of machine learning models for various classification and regression tasks, there are still some challenges from this domain that should be addressed. Among two most important challenges are feature selection and hyper-parameters' optimisation. Performance of machine learning models to the large extent depend on the un-trainable parameters, which are known as the hyper-parameters and finding optimal or suboptimal values of such parameters for concrete problem is NP-hard task itself. Moreover, machine learning models utilize large datasets with many features and typically only subset of such features has significant impact on the target variable. Therefore, the goal of feature selection is to choose a subset of features and to eliminate redundant ones.

Study shown in this article proposes a new approach based on the well-known whale optimisation algorithm (WOA), which belongs to the group of swarm intelligence metaheuristics. Proposed improved WOA was first tested for standard unconstrained CEC2017 benchmark suite and it was later adapted for simultaneous feature selection and support vector machine (SVM) hyper-parameter tuning and validated for medical diagnostics by using breast cancer, diabetes, and erythematous squamous datasets. The contributions of this research are twofold. First, an improved version of the WOA, that addresses known deficiencies of the basic version by implementing the concepts of chaotic initialisation and opposition-based learning (OBL), and by incorporating the firefly search (FS), has been proposed, Second, proposed method was adopted for simultaneously optimisation

of SVM hyper-parameters and feature selection and achieved better performance metrics than other contemporary classifiers for three medical benchmark datasets.

The rest of the manuscript is structured as follows. Section 2 presents an overview of the recent swarm intelligence applications. Section 3 introduces the WOA metaheuristics, their improvements, and proposes the hybrid SVM optimisation approach. Section 4 exhibits the experimental findings and the discussion. The last section brings the final remarks, proposes a future work, and concludes the article. The authors hope to provide an accurate feature detection method used for their diagnostics.

## 2 Background and Related Work

Swarm intelligence is based on natural biological systems, and function with a population of self-organising agents who interact with one another locally and globally with their environment. Although there is no central component that governs and directs individual behaviour, local interactions between agents result in the formation of globally coordinated behaviour. The swarm intelligence algorithm can produce quick, inexpensive and high-quality solutions for challenging optimisation problems [9].

In swarm intelligence algorithms, a swarm is a population consisting of simple homogeneous agents. The swarm might have direct or indirect interactions. Direct interactions refer to audio or visual situations, touch, etc. In indirect interactions, agents are affected by the environment and changes made to it, which can be a result of other agents' actions, or from external influences.

The operators of evolutionary algorithms, notably selection, crossover and mutation, also employ swarm metaheuristics. The crossover operator is employed to look for quicker convergence in a single sub-domain. Mutations help avoid local optima and are used for random searching. The selection operator enables the system to move to the desired conditions.

Popular implementations of swarm intelligence algorithms are the particle swarm algorithm (PSO) [10], the ant colony optimisation (ACO) algorithm [11], the wasp swarm (WS) algorithm [12], the stochastic diffusion search (SDS) algorithm [13], the artificial bee colony (ABC) algorithm [14], the firefly search (FS) algorithm [15], whale optimisation algorithm (WOA) [16], the multi-swarm optimisation (MSO) algorithm [17], the krill herd (KH) algorithm [18], the dolphin echolocation (DE) algorithm [19], etc. Each of these algorithms, as well as many others which fall into this category, have strengths and weaknesses for different optimisation problems. One algorithm may outperform the other when applied to one problem and significantly under perform on another.

As already noted in Section 1, swarm intelligence has seen successful application to different areas of optimisation. Research proposed introduces a variety of the swarm intelligence algorithm, tested on Himmelblau's nonlinear optimisation and speed reducer design problems, and used to attempt to solve problems such as the travelling salesman problem, the robot path planning problem etc. The study presented in [20] reviews earlier works and concludes that there is strong applicability of swarm intelligence algorithms for solving complex practical problems. Authors give overviews of the application of the swarm intelligence algorithm for swarm robotics. A chapter in [21] presents interesting insights and predictions about the use of the swarm intelligence algorithm for swarm robotics in the future. Study presented in [22] provides detailed reviews of the swarm intelligence algorithm and its practical implementation, from early conceptions in the late 80s onward. Aside from application in robotics, swarm intelligence algorithms have found application in computer networks [23,24], power management [25], engineering [26,27], Internet of things [28], social network analysis [29], medical diagnostics [30,31], data mining [32], etc.

Algorithms such as PSO and GA are known to get trapped in sub-optimal search space regions. Algorithms based on maritime animal behaviour are a relatively recent addition to the field of swarm intelligence algorithm research. Aside from the WOA, there are several notable algorithms reported in the literature, inspired by animal behaviour. Review [33] gives a review of other evolutionary algorithms inspired by animals for a specific problem optimisation. However, their review covers a range of maritime animal-inspired algorithms. Study [34] gives an overview of other animal behaviour inspired algorithms, not limited to maritime animals. Research proposed in [35] modifies the salp swarm algorithm optimisation [36,37], based on the salps sailing and foraging behaviour. Work presented introduces a dolphin inspired swarm optimisation algorithm with mimics the dolphins hunting technique, using their available senses. Research presented in demonstrates an optimisation algorithm modification on five real-life engineering problems, etc.

In recent years, SI was frequently applied in different scientific fields with promising results, usually targeting NP-hard problems from the fields of computer science and information technology. Applications include global numerical optimisation problem wireless sensor network efficiency [38], localisation and prolonging the overall lifetime [39,40] and cloud task scheduling [41,42].

Finally, swarm intelligence has also been successfully combined with machine learning models and in this way hybrid methods were devised. Some example of such approaches include: training and feature selection of artificial neural network (ANN) [43–49], assisting prediction of COVID-19 cases [50,51], hyper-parameters tuning [52–54] etc.

### 3 Proposed Methodology

The methodology of this research can be expressed in seven steps, of which the first step was performed manually when all input datasets were acquired from different sources, and the second-to-last step was also automated for the ease of multiple re-runs when needed, like others those before it, but its execution is done manually upon successful completion of the following steps:

- Download and normalise the datasets;
- Randomly split the datasets into 10 sets of training and testing data;
- Execute the optimisation algorithm for the given number of iterations;
- Randomly split the originally downloaded datasets into 10 new validation sets;
- Perform predictions with the trained model using the validation sets;
- Evaluate the prediction results;
- Compare results of this study to results from previous research.

#### 3.1 Overview of Basic Whale Optimisation Algorithm

Whales are highly intelligent animals that have twice the number of brain cells similar to human spindle cells, which are responsible for emotions, decision making and social behaviour. Whales are often found in social groups, of families. Most whales prey on fish. Fish are much faster than whales. Because of this, some whales have developed a hunting method that allows them to hunt schools of fish. This method is often called the bubble-net feeding method. Researchers have found two variations of this method, which whales use. A part of this method is an approach where whales encircle their prey in a spiral manoeuvre. They dive down below the school of fish and start circling in a spiral path, slowly working their way to the surface, all the while creating a wall of bubbles above their traversed path. This wall of bubbles directs fish in an ever-smaller area as the circular path becomes smaller and smaller. Finally, the whale can capture large parts of the school as it reaches the surface. Whales do not know where the school of fish is located, ahead of time. This method helps them locate the fish

after the manoeuvre. This feeding method is a direct inspiration of the WOA, which is mathematically modelled and used to perform optimisation [55]. This method can be used to search for the optimal location within the search space, and can be presented by these equations:

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot D \quad (1)$$

$$D = |\vec{B} \otimes \vec{X}_p(t) - \vec{X}(t)| \quad (2)$$

$$\vec{A} = 2a\vec{r} - a \quad (3)$$

$$\vec{B} = 2\vec{r} \quad (4)$$

$$a = 2 - \frac{2t}{t_{max}} \quad (5)$$

In Eqs. (1)–(5),  $\otimes$  is an element-wise multiplication operator,  $t$  is the current iteration,  $\vec{X}_p$  and  $\vec{X}$  are current position vectors of the best-known solution and the position vector,  $\vec{r}$  is a random vector,  $\vec{A}$  and  $\vec{B}$  are coefficient vectors,  $a$  is a control parameter, and  $t_{max}$  is the maximum number of iterations. This model can be applied to an  $n$ -dimensional search space and it has two mechanisms, with equal probability of utilisation. The first is the shrinking encircling manoeuvre and the second is the spiral updating position manoeuvre. The former is achieved by linearly decreasing the control parameter  $a$  from 2 to 0 over the course of iterations, while the latter is achieved by calculating the distance between the whale and its prey, according to the equation for the distance.

$$\vec{X}(t+1) = \vec{D} e^{bl} \cos 2\pi l + \vec{X}^*(t) \quad (6)$$

In Eq. (6),  $\vec{D}$  is the smallest distance between the whale and its prey so far,  $b$  is a constant value determining the shape of the logarithmic spiral, and  $l$  is a random value in  $[-1, 1]$ .  $X^*$  is the best agent in the current iteration  $t$ . As mentioned before, the two mechanisms are used with equal probability based on a random number in  $[0, 1]$ , according to:

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot D, & p < 0.5 \\ \vec{D} e^{bl} \cos 2\pi l + \vec{X}^*(t), & p \geq 0.5 \end{cases} \quad (7)$$

By assigning random values to  $\vec{A}$  in ranges  $[-\infty, -1]$  and  $[1, +\infty]$  an optimal solution search can be performed. In this search behaviour, a random whale is selected, instead of the one with the best search performance. This allows the WOA to search for a global optimum. This can be expressed as:

$$D = |\vec{B} \otimes \vec{X}_r(t) - \vec{X}(t)| \quad (8)$$

$$\vec{X}(t+1) = \vec{X}_r(t) - \vec{A} \cdot D \quad (9)$$

In Eqs. (8) and (9),  $\vec{X}_r$  is a random vector representing a random whale from the current population.

The pseudo-code of the WOA can be found in a paper whereas a modified WOA with simultaneous hyper-parameter tuning is implemented.

The model representing this experimental setup is further explained in the model section. The proposed model balances the searching and utilising features of the algorithm. These results can be further improved by using multi-swarm mechanisms, as reported in the original paper. In addition to this modification, the paper suggests modifying the population generation process. Instead of using a random feature selection for each agent in a population, it suggests generating the population-based

on chaos theory. In this approach, a chaos progression, sensitive to initial conditions, is used to modify the best agents from the previous population when generating the next population.

This can be achieved by introducing a self-adapting chaotic disturbance mechanism, expressed as:

$$\vec{X}_g = \vec{X}^* + R \cdot Chaos \quad (10)$$

In Eq. (10), *Chaos* represents the chaos value, and *R* is used to ensure a better balance between the algorithm's searching and utilising features. *R* is decreased from 1 to 0 in each of the iterations. This population generation method was proven to give better performances [56–59].

### 3.2 Drawbacks of the Original WOA and Proposed Improved Method

The basic WOA implementation has impressive performances, and it is considered to be a very powerful optimiser. However, the extensive experimentations with different benchmark functions sets have revealed drawbacks that can hinder the performances of the WOA under some circumstances. Namely, the basic WOA suffers from scarce exploration and inappropriate intensification-diversification trade-off, which can limit the algorithm's power to explore different promising areas of the search domain and lead to getting stuck in the local optima. Consequently, in some runs, the basic WOA will not converge to the optimal areas of the search domain, obtaining poor mean values. To address these deficiencies, the proposed enhanced WOA metaheuristics incorporate three mechanisms: chaotic population initialisation, OBL, and FS [60].

The first proposed enhancement is the chaotic population initialisation, which was first proposed in [61]. The majority of the metaheuristics depend on random generators, but recent publications suggest that the search can be more intense if chaotic sequences are utilised. Numerous chaotic maps exist today, including circle, Chebushev, iterative, sine, tent, logistic, singer and many more. After thorough experimentations with all mentioned maps, the most promising results were obtained with a logistic map, which has been consequently chosen for implementation in WOA.

The second mechanism that was incorporated in the proposed modified WOA is OBL, first introduced in [62]. The OBL procedure gained popularity fast as it can significantly improve both intensification and diversification processes. The OBL procedure is applied in two stages, first when the initial population is spawned and when a new population is spawned after completing the evaluation and finding the best agent  $X^*$  of the finished iteration. Since the population  $P$  is a set containing a number of agents  $X$ , it can be said that  $P = (X_1, X_2, \dots, X_n)$ , where  $n$  is the number of all agents in a single population. As each agent represents an attempted solution  $(s_1, s_2, \dots, s_N)$ , of  $N$  solutions, for a set of parameters  $(p_1, p_2, \dots, p_M)$ , optimised for  $M$  dimensions, at every given iteration  $t$ , a population contains agents with features for optimisation of a combination of parameters for a given solution. Creating agents can be expressed as:

$$X_{s,p} = \phi (u_p - l_p) + l_p \quad (11)$$

In Eq. (11),  $l_p$  and  $u_p$  are lower and upper bounds of the parameter  $p$  dimensions, while  $\phi$  represents a pseudo-random value. To ensure chaotic initialisation, using the OBL procedure, for every generated agent, another one is generated for the opposite  $p$  dimension of the original  $X_s$ . This guarantees a chaotic component and this process can be described with the following expression:

$$X_{s,p}^o = l_p + u_p - X_{s,p} \quad (12)$$

In Eq. (12),  $X_{s,p}^o$  is an agent for the opposite parameter dimension  $p$ , created for the original agent  $X_{s,p}$ . As a result of creating a complementary set of opposing agents, an oppositional population  $P^o$  is created, which is comprised of oppositional agents, and is hence defined as  $P^o = (X_1^o, X_2^o, \dots, X_n^o)$ , where each  $X_i^o$  is a single agent created in opposition to the original  $X_i$ . Both populations are merged into a single one:

$$P \leftarrow P \cup P^o \quad (13)$$

In Eq. (13), the new population becomes the original population  $P$ , afterwards joined with the opposing population  $P^o$ . This union represents the new population of agents for the next iteration.

To incorporating the FS mechanism in the proposed modified WOA,  $\vec{X}(t+1)$  is updated for a given solution  $s$ , denoted as  $\vec{X}_s(t+1)$ , according to the following expression:

$$\vec{X}_s(t+1) = \vec{X}_s(t) + \frac{\beta_0}{1 + \gamma r_{s,p}^2} (\vec{X}_p - \vec{X}_s) + \alpha(\kappa - 0.5) \quad (14)$$

In Eq. (14),  $\kappa$  is a generalised Gaussian distribution,  $\alpha$  is a control parameter of the FS mechanism, whose dynamic change guarantees improved performance. It is gradually decreased from the initial value, down to its minimum, over the course of the iterations. Parameters  $\beta_0$  and  $\gamma$  are control parameters of randomisation.  $\vec{X}_s$  is the current candidate solution agent,  $\vec{X}_p$  is another candidate with a modification of one or more parameters,  $r_{s,p}$  is the distance between those two candidate agents, which is calculated as:

$$r_{s,p} = \|\vec{X}_s - \vec{X}_p\| = \sqrt{\sum_{i=1}^{|P|} (X_{s,i} - X_{p,i})^2} \quad (15)$$

In Eq. (15), the distance is calculated for each dimension of the given solution  $s$  and parameter  $p$  on all agents of the population, where  $|P|$  is the number of agents in the population.

Inspired by introduced enhancements, proposed method is named chaotic oppositional WOA with firefly search (COWOAFS-SVM).

The proposed modified WOA can be expressed with the pseudo-code depicted in Algorithm 1.

---

**Algorithm 1:** The COWOAFS-SVM pseudo-code

---

Initialize required control parameters  $\alpha$ ,  $\beta_0$  and  $\gamma$  for the FA.

$P \leftarrow$  Initialisation of the starting population of agents.

$P^o \leftarrow$  Chaotic initialisation of the opposing population, based on the original population  $P$ .

$P \leftarrow P \cup P^o$ , join both populations into a single and evaluate each agent.

$X^* \leftarrow$  Obtain the best agent using the fitness function.

**while**  $j < t_{max}$  **do**

$\alpha \leftarrow$  According to Eq. (5).

**for each agent do**

$X(t+1) \leftarrow$  According to Eqs. (7) or (9), dependent on  $A$  (per Eq. (3)).

**end for**

Evaluate  $X(t+1)$  and update  $X^*$ , using the fitness function.

$P \leftarrow$  Spawn and mutate population, based on  $X^*$ , according to Eqs. (10)–(13).

**end while**

---

Experimental result cross-validation analysis is performed. These results are averaged to attain the result for each of the datasets and aggregated by each of the features of these datasets. Results are presented in the experimental results section and further explained in the discussion section of the paper.

#### 4 Experimental Setup, Comparative Analysis and Discussion

For this research, the original algorithm and the algorithm created using the meta-heuristic approach with feature selection and simultaneous hyper-parameter tuning driven by the modified swarm intelligence algorithm are evaluated. This approach was proven to give favourable results in earlier research. Finally, both these algorithms were tested using 30 standard unconstrained benchmarking functions. To test the modified algorithm, first a set of standard benchmark functions retrieved from the well-known Congress on Evolutionary Computation (CEC) benchmark function suite, and afterwards, the proposed method was validated for SVM hyper-parameter tuning and feature selection. In both experiments, proposed improved WOA was tested with the parameters as suggested in the original paper.

##### 4.1 Standard CEC Function Simulations

The proposed method is tested on a set of standard bound-constrained benchmarks from the CEC benchmark suite [63]. Statistical results of these benchmarks are presented in the results section. Functions with a single global optimum ( $F_1$ – $F_7$ ) are used to evaluate the capability of exploitation. Functions with many local optimums ( $F_8$ – $F_{23}$ ) are used to evaluate the capability of exploration. Functions ( $F_{24}$ – $F_{29}$ ) are used to evaluate the algorithms ability to escape local minima, as they are a set of very challenging functions. The proposed algorithm was run 30 times for each benchmark function, every time with a different population. Algorithms included in comparative analysis were implemented for the purpose of this study and tested with default parameters. Opponent algorithms include: Harris hawks' optimisation (HHO) [64], improved HHO (IHHO) [65], differential evolution (DE) [66], grasshopper optimisation algorithm (GOA) [67], grey wolf optimizer (GWO) [68], moth-flame optimisation (MFO) [69], multi-verse optimizer (MVO) [70], particle swarm optimisation (PSO), whale optimisation algorithm (WOA), sine cosine algorithm (SCA) [71] and firefly algorithm (FA).

This research utilizes the same approach presented. The research presented in reported the results with  $N = 30$  and  $T = 500$ . Since the proposed COWOAFS-SSVM algorithm uses more FFE in every run, the max FFE was utilised as the finish criteria. All other methods use one FFE per solution during the initialisation and update phase, and to provide fair analysis, max FFE was set to  $15,030 (N + N \cdot T)$ . The values of the control parameters for the other methods can be obtained from.

The reported results from the [Tab. 1](#) show that the proposed COWOAFS-SVM metaheuristics obtained the best results on 30 benchmark functions, and was tied for the first place on seven more occasions. By conducting a comparison of the results of COWOAFS-SVM against the basic WOA and FA, it is clear that hybrid algorithm outperformed basic versions on every single benchmark instance.



**Table 1:** CEC 2017 comparative analysis results

Algorithm	$F_1$		$F_2$		$F_3$		$F_4$		$F_5$	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
IHHO	$1.86 \times 10^{-2}$	26.921	/	/	$3.02 \times 10^{-2}$	52.152	$4.03 \times 10^{-2}$	2.607	$5.05 \times 10^{-2}$	3.251
HHO	$1.75 \times 10^{-6}$	$4.29 \times 10^{+5}$	/	/	$6.71 \times 10^{-2}$	$3.24 \times 10^{+2}$	$4.37 \times 10^{-2}$	53.631	$5.35 \times 10^{-2}$	24.927
DE	$7.54 \times 10^{-7}$	$1.71 \times 10^{+7}$	/	/	$4.59 \times 10^{-3}$	$1.35 \times 10^{+3}$	$4.29 \times 10^{-2}$	8.530	$5.52 \times 10^{-2}$	6.232
GOA	$1.56 \times 10^{-5}$	$5.24 \times 10^{+4}$	/	/	$3.05 \times 10^{-2}$	61.300	$4.15 \times 10^{-2}$	19.48	$5.25 \times 10^{-2}$	16.803
GWO	$1.53 \times 10^{-7}$	$4.85 \times 10^{+6}$	/	/	$3.57 \times 10^{-3}$	$2.77 \times 10^{+3}$	$4.09 \times 10^{-2}$	10.705	$5.19 \times 10^{-2}$	8.543
MFO	$7.17 \times 10^{-6}$	$2.18 \times 10^{+7}$	/	/	$9.04 \times 10^{-3}$	$9.31 \times 10^{+3}$	$4.20 \times 10^{-2}$	27.727	$5.31 \times 10^{-2}$	12.860
MVO	$1.79 \times 10^{+4}$	$7.99 \times 10^{+3}$	/	/	$3.05 \times 10^{-2}$	46.451	$4.06 \times 10^{-2}$	1.392	$5.17 \times 10^{-2}$	9.888
PSO	$9.49 \times 10^{+4}$	$8.42 \times 10^{+2}$	/	/	$3.49 \times 10^{-2}$	65.409	$4.07 \times 10^{-2}$	10.318	$5.26 \times 10^{-2}$	7.305
WOA	$4.27 \times 10^{-7}$	$3.81 \times 10^{+6}$	/	/	$5.16 \times 10^{-3}$	$4.22 \times 10^{+2}$	$4.61 \times 10^{-2}$	69.033	$5.51 \times 10^{-2}$	17.46
SCA	$1.15 \times 10^{+8}$	$5.91 \times 10^{+7}$	/	/	$4.03 \times 10^{-3}$	$8.42 \times 10^{+2}$	$4.85 \times 10^{-2}$	47.271	$5.59 \times 10^{-2}$	9.352
FA	$1.61 \times 10^{-5}$	$3.77 \times 10^{+4}$	/	/	$3.09 \times 10^{-2}$	54.991	$4.17 \times 10^{-2}$	18.858	$5.28 \times 10^{-2}$	19.302
COWOAFS-	$1.42 \times 10^{-2}$	14.285	/	/	$3.02 \times 10^{-2}$	29.421	$4.05 \times 10^{-2}$	3.746	$5.00 \times 10^{-2}$	4.215
SVM										
Algorithm	$F_6$		$F_7$		$F_8$		$F_9$		$F_{10}$	
Algorithm	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
IHHO	$6.00 \times 10^{-2}$	0.082	$7.49 \times 10^{-2}$	10.041	$8.11 \times 10^{-2}$	6.526	$1.13 \times 10^{-3}$	85.42	$1.69 \times 10^{-3}$	$1.31 \times 10^{-2}$
HHO	$6.38 \times 10^{-2}$	12.320	$7.96 \times 10^{-2}$	18.921	$8.29 \times 10^{-2}$	5.700	$1.44 \times 10^{-3}$	$1.24 \times 10^{+2}$	$2.03 \times 10^{-3}$	$3.42 \times 10^{-2}$
DE	$6.28 \times 10^{-2}$	4.744	$8.01 \times 10^{-2}$	10.373	$8.62 \times 10^{-2}$	6.873	$1.76 \times 10^{-3}$	$1.48 \times 10^{+2}$	$2.09 \times 10^{-3}$	$2.01 \times 10^{-2}$
GOA	$6.08 \times 10^{-2}$	10.295	$7.32 \times 10^{-2}$	11.375	$8.31 \times 10^{-2}$	14.512	$9.97 \times 10^{-2}$	93.212	$1.96 \times 10^{-3}$	$3.17 \times 10^{-2}$
GWO	$6.01 \times 10^{-2}$	1.909	$7.35 \times 10^{-2}$	16.343	$8.16 \times 10^{-2}$	5.053	$9.14 \times 10^{-2}$	12.11	$1.76 \times 10^{-3}$	$3.10 \times 10^{-2}$
MFO	$6.02 \times 10^{-2}$	2.411	$7.46 \times 10^{-2}$	22.655	$8.29 \times 10^{-2}$	13.786	$1.23 \times 10^{-3}$	$2.76 \times 10^{+2}$	$2.02 \times 10^{-3}$	$3.27 \times 10^{-2}$
MVO	$6.03 \times 10^{-2}$	4.365	$7.30 \times 10^{-2}$	11.278	$8.25 \times 10^{-2}$	12.216	$9.00 \times 10^{-2}$	0.012	$1.82 \times 10^{-3}$	$3.60 \times 10^{-2}$
PSO	$6.10 \times 10^{-2}$	3.539	$7.26 \times 10^{-2}$	9.008	$8.19 \times 10^{-2}$	5.982	$9.00 \times 10^{-2}$	0.003	$1.50 \times 10^{-3}$	$2.84 \times 10^{-2}$
WOA	$6.36 \times 10^{-2}$	13.695	$7.82 \times 10^{-2}$	23.692	$8.45 \times 10^{-2}$	17.470	$1.54 \times 10^{-3}$	$3.94 \times 10^{+2}$	$2.19 \times 10^{-3}$	$3.16 \times 10^{-2}$
SCA	$6.24 \times 10^{-2}$	4.105	$7.84 \times 10^{-2}$	13.299	$8.47 \times 10^{-2}$	7.577	$1.03 \times 10^{-3}$	85.98	$2.51 \times 10^{-3}$	$2.18 \times 10^{-2}$
FA	$6.71 \times 10^{-2}$	11.393	$7.35 \times 10^{-2}$	11.55	$8.33 \times 10^{-2}$	13.914	$9.97 \times 10^{-2}$	81.44	$1.93 \times 10^{-3}$	$2.96 \times 10^{-2}$
COWOAFS-	$6.00 \times 10^{-2}$	0.082	$7.18 \times 10^{-2}$	11.544	$8.05 \times 10^{-2}$	6.325	$9.28 \times 10^{-2}$	41.57	$1.48 \times 10^{-3}$	$1.26 \times 10^{-2}$
SMV										
Algorithm	$F_{11}$		$F_{12}$		$F_{13}$		$F_{14}$		$F_{15}$	
Algorithm	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
IHHO	$1.12 \times 10^{-3}$	13.523	$4.25 \times 10^{-5}$	$3.05 \times 10^{+5}$	$4.42 \times 10^{-3}$	$2.18 \times 10^{+3}$	$1.42 \times 10^{-3}$	1.651	$2.15 \times 10^{-3}$	$5.65 \times 10^{-2}$
HHO	$1.16 \times 10^{-3}$	45.729	$2.56 \times 10^{-6}$	$1.13 \times 10^{+6}$	$1.92 \times 10^{-4}$	$1.16 \times 10^{+4}$	$1.83 \times 10^{-3}$	$2.41 \times 10^{+2}$	$8.63 \times 10^{-3}$	$5.55 \times 10^{-2}$
DE	$1.14 \times 10^{-3}$	36.317	$9.15 \times 10^{-4}$	$6.58 \times 10^{+4}$	$1.35 \times 10^{-3}$	78.355	$1.46 \times 10^{-3}$	11.826	$1.51 \times 10^{-3}$	18.454
GOA	$1.17 \times 10^{-3}$	58.009	$2.24 \times 10^{-6}$	$1.15 \times 10^{+6}$	$1.65 \times 10^{-4}$	$1.13 \times 10^{+4}$	$2.93 \times 10^{-3}$	$1.15 \times 10^{+3}$	$6.48 \times 10^{-3}$	$4.32 \times 10^{+3}$
GWO	$1.34 \times 10^{-3}$	183.524	$1.31 \times 10^{-6}$	$1.54 \times 10^{+6}$	$1.26 \times 10^{-4}$	$7.82 \times 10^{+3}$	$3.19 \times 10^{-3}$	$1.82 \times 10^{+3}$	$5.63 \times 10^{-3}$	$3.16 \times 10^{+3}$
MFO	$1.23 \times 10^{-3}$	107.133	$2.23 \times 10^{-6}$	$4.81 \times 10^{+6}$	$1.61 \times 10^{-4}$	$1.39 \times 10^{+4}$	$8.42 \times 10^{-3}$	$5.42 \times 10^{+3}$	$1.25 \times 10^{-4}$	$1.02 \times 10^{+4}$
MVO	$1.14 \times 10^{-3}$	27.331	$1.52 \times 10^{-6}$	$1.41 \times 10^{+6}$	$9.89 \times 10^{-3}$	$2.55 \times 10^{+3}$	$2.15 \times 10^{-3}$	$1.03 \times 10^{+3}$	$4.05 \times 10^{-3}$	$2.45 \times 10^{+3}$
PSO	$1.10 \times 10^{-3}$	3.727	$4.35 \times 10^{-4}$	$1.26 \times 10^{+4}$	$1.01 \times 10^{-4}$	$7.23 \times 10^{+3}$	$1.49 \times 10^{-3}$	88.291	$1.81 \times 10^{-3}$	$3.75 \times 10^{-2}$

(Continued)

Table 1: Continued

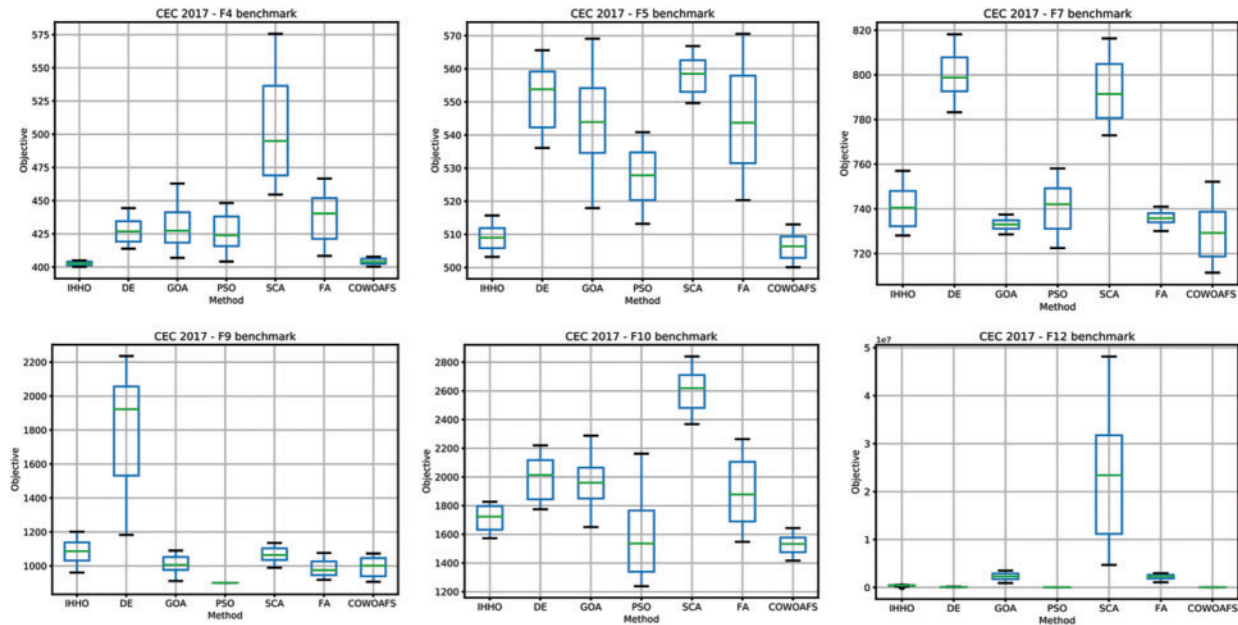
Algorithm	$F_1$		$F_2$		$F_3$		$F_4$		$F_5$	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
WOA	$1.22 \times 10^{-3}$	82.415	$4.85 \times 10^6$	$5.12 \times 10^6$	$1.57 \times 10^4$	$1.38 \times 10^4$	$3.42 \times 10^3$	$9.82 \times 10^2$	$1.42 \times 10^4$	$9.88 \times 10^3$
SCA	$1.24 \times 10^{-3}$	96.535	$2.41 \times 10^7$	$2.05 \times 10^7$	$6.43 \times 10^4$	$4.69 \times 10^4$	$1.99 \times 10^3$	$4.31 \times 10^2$	$3.21 \times 10^3$	$1.41 \times 10^3$
FA	$1.16 \times 10^{-3}$	39.05	$2.32 \times 10^6$	$1.21 \times 10^6$	$1.21 \times 10^4$	$1.05 \times 10^4$	$1.88 \times 10^3$	$3.21 \times 10^2$	$3.67 \times 10^3$	$2.13 \times 10^3$
COWOAFS-SMV	$1.10 \times 10^{-3}$	1.724	$3.24 \times 10^4$	$2.15 \times 10^4$	$1.34 \times 10^3$	22.725	$1.44 \times 10^3$	21.158	$1.47 \times 10^3$	10.152
<b>Algorithm</b>	$F_{16}$		$F_{17}$		$F_{18}$		$F_{19}$		$F_{20}$	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
IHHO	$1.73 \times 10^{-3}$	59.44	$1.73 \times 10^3$	7.519	$4.79 \times 10^3$	$1.68 \times 10^3$	$1.90 \times 10^3$	6.993	$2.02 \times 10^3$	19.561
HHO	$1.89 \times 10^{-3}$	$1.47 \times 10^{+2}$	$1.79 \times 10^3$	65.751	$2.02 \times 10^4$	$1.41 \times 10^4$	$1.71 \times 10^4$	$1.21 \times 10^4$	$2.23 \times 10^3$	86.017
DE	$1.69 \times 10^{-3}$	41.15	$1.77 \times 10^3$	19.514	$1.84 \times 10^3$	23.298	$2.75 \times 10^3$	$8.35 \times 10^2$	$2.05 \times 10^3$	23.711
GOA	$1.78 \times 10^{-3}$	$1.76 \times 10^{+2}$	$1.83 \times 10^3$	$1.21 \times 10^2$	$1.63 \times 10^4$	$1.31 \times 10^4$	$3.25 \times 10^3$	$1.95 \times 10^3$	$2.15 \times 10^3$	74.824
GWO	$1.79 \times 10^{-3}$	$1.11 \times 10^{+2}$	$1.77 \times 10^3$	38.759	$2.55 \times 10^4$	$1.84 \times 10^4$	$2.75 \times 10^4$	$2.38 \times 10^4$	$2.09 \times 10^3$	73.994
MFO	$1.85 \times 10^{-3}$	$15.23 \times 10^{+2}$	$1.78 \times 10^3$	65.311	$2.21 \times 10^4$	$1.39 \times 10^4$	$7.81 \times 10^3$	$6.15 \times 10^3$	$2.13 \times 10^3$	72.321
MVO	$1.80 \times 10^{-3}$	$1.44 \times 10^{+2}$	$1.80 \times 10^3$	46.126	$2.03 \times 10^4$	$1.25 \times 10^4$	$4.63 \times 10^3$	$2.62 \times 10^3$	$2.12 \times 10^3$	86.303
PSO	$1.65 \times 10^{-3}$	65.364	$1.72 \times 10^3$	16.123	$7.63 \times 10^3$	$4.46 \times 10^3$	$3.13 \times 10^3$	$2.05 \times 10^3$	$2.06 \times 10^3$	35.410
WOA	$1.96 \times 10^{-3}$	$14.92 \times 10^{+2}$	$1.82 \times 10^3$	73.459	$2.13 \times 10^4$	$1.95 \times 10^2$	$2.07 \times 10^5$	$1.16 \times 10^5$	$2.19 \times 10^3$	$1.11 \times 10^{+2}$
SCA	$1.73 \times 10^{-3}$	95.425	$1.80 \times 10^3$	$25.30 \times 10^{+3}$	$8.77 \times 10^4$	$9.23 \times 10^2$	$1.15 \times 10^4$	$1.44 \times 10^3$	$2.14 \times 10^3$	46.855
FA	$1.79 \times 10^{-3}$	$1.73 \times 10^{+2}$	$1.82 \times 10^3$	$1.15 \times 10^2$	$1.67 \times 10^4$	$1.45 \times 10^4$	$3.18 \times 10^3$	$1.59 \times 10^3$	$2.12 \times 10^3$	71.303
COWOAFS-SMV	$1.67 \times 10^{-3}$	79.319	$1.70 \times 10^3$	7.512	$1.79 \times 10^3$	21.316	$1.90 \times 10^3$	8.547	$2.00 \times 10^3$	9.583
<b>Algorithm</b>	$F_{21}$		$F_{22}$		$F_{23}$		$F_{24}$		$F_{25}$	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
IHHO	$2.20 \times 10^{-3}$	4.615	$2.28 \times 10^3$	17.820	$2.59 \times 10^3$	14.213	$2.68 \times 10^3$	$1.31 \times 10^2$	$2.87 \times 10^3$	85.338
HHO	$2.35 \times 10^{-3}$	53.711	$2.32 \times 10^3$	25.234	$2.69 \times 10^3$	35.522	$2.82 \times 10^3$	93.623	$2.95 \times 10^3$	49.573
DE	$2.25 \times 10^{-3}$	78.104	$2.29 \times 10^3$	17.513	$2.63 \times 10^3$	15.163	$2.66 \times 10^3$	69.502	$2.91 \times 10^3$	15.543
GOA	$2.30 \times 10^{-3}$	56.877	$2.38 \times 10^3$	$1.08 \times 10^2$	$2.64 \times 10^3$	23.536	$2.73 \times 10^3$	57.833	$2.93 \times 10^3$	32.598
GWO	$2.30 \times 10^{-3}$	32.884	$2.31 \times 10^3$	57.573	$2.62 \times 10^3$	13.862	$2.74 \times 10^3$	25.132	$2.94 \times 10^3$	28.256
MFO	$2.32 \times 10^{-3}$	29.255	$2.35 \times 10^3$	93.557	$2.63 \times 10^3$	11.327	$2.75 \times 10^3$	76.435	$2.96 \times 10^3$	37.776
MVO	$2.32 \times 10^{-3}$	11.839	$2.33 \times 10^3$	$1.11 \times 10^2$	$2.65 \times 10^3$	10.445	$2.74 \times 10^3$	18.246	$2.92 \times 10^3$	84.256
PSO	$2.27 \times 10^{-3}$	49.783	$2.33 \times 10^3$	$1.03 \times 10^2$	$2.60 \times 10^3$	72.300	$2.70 \times 10^3$	76.143	$2.90 \times 10^3$	33.735
WOA	$2.34 \times 10^{-3}$	60.021	$2.48 \times 10^3$	$2.45 \times 10^2$	$2.66 \times 10^3$	29.838	$2.77 \times 10^3$	85.902	$2.98 \times 10^3$	$1.03 \times 10^{+2}$
SCA	$2.29 \times 10^{-3}$	65.229	$2.41 \times 10^3$	66.636	$2.67 \times 10^3$	45.449	$2.78 \times 10^3$	11.548	$2.98 \times 10^3$	37.291
FA	$2.29 \times 10^{-3}$	34.701	$2.36 \times 10^3$	$1.10 \times 10^2$	$2.62 \times 10^3$	17.452	$2.72 \times 10^3$	$1.05 \times 10^2$	$2.93 \times 10^3$	47.019
COWOAFS-SMV	$2.20 \times 10^{-3}$	49.426	$2.25 \times 10^3$	12.871	$2.54 \times 10^3$	21.586	$2.66 \times 10^3$	$1.74 \times 10^2$	$2.79 \times 10^3$	92.573
<b>Algorithm</b>	$F_{26}$		$F_{27}$		$F_{28}$		$F_{29}$		$F_{30}$	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
IHHO	$2.93 \times 10^{-3}$	$1.66 \times 10^{+2}$	$3.19 \times 10^3$	33.657	$3.30 \times 10^3$	48.694	$3.20 \times 10^3$	28.982	$2.30 \times 10^4$	$1.45 \times 10^{+4}$

(Continued)



On some functions, the proposed COWOAFS-SVM drastically improved the results of the basic variants; such is the example of F1, where it obtained 1000 times better results than the basic WOA and FA. Overall results indicate that the proposed COWOAFS-SVM outperformed significantly all approaches covered by the research.

Dispersion of solutions over 50 runs for some algorithms and chosen functions are shown in Fig. 1.



**Figure 1:** Dispersion of solutions of benchmark functions  $F_4$ ,  $F_5$ ,  $F_7$ ,  $F_9$ ,  $F_{10}$  and  $F_{12}$

To statistically validate the significance of the proposed method, the Friedman test and the two variance ranking analysis were performed. Tabs. 2 and 3 depict the results obtained by twelve observed algorithms on the set of 30 CEC2017 benchmark functions for Friedman test ranks and aligned Friedman ranks, respectively. It is clear from both tables that the proposed COWOAFS-SVM significantly outperformed all other algorithms.

**Table 2:** Friedman test ranks for the compared algorithms over 30 CEC2017 functions

Function	IHHO	HHO	DE	GOA	GWO	MFO	MVO	PSO	WOA	SCA	FA	COWOAFS-SVM
$F_1$	2	7	11	5	9	8	3	4	10	12	6	1
$F_3$	1.5	7	10	3.5	8	12	3.5	6	11	9	5	1.5
$F_4$	1	10	9	6	5	8	3	4	11	12	7	2
$F_5$	2	9	11	5	4	8	3	6	10	12	7	1
$F_6$	1.5	11	9	6	3	4	5	7	10	8	12	1.5
$F_7$	8	11	12	4	5.5	7	3	2	9	10	5.5	1
$F_8$	2	6.5	12	8	3	6.5	5	4	10	11	9	1
$F_9$	8	10	12	5.5	3	9	1.5	1.5	11	7	5.5	4
$F_{10}$	3	9	10	7	4	8	5	2	11	12	6	1
$F_{11}$	3	6.5	4.5	8	12	10	4.5	1.5	9	11	6.5	1.5
$F_{12}$	4	10	3	8	5	7	6	2	11	12	9	1
$F_{13}$	3	11	2	10	7	9	4	5	8	12	6	1

(Continued)

**Table 2:** Continued

Function	IHHO	HHO	DE	GOA	GWO	MFO	MVO	PSO	WOA	SCA	FA	COWOAFS-SVM
F <sub>14</sub>	1	5	3	9	10	12	8	4	11	7	6	2
F <sub>15</sub>	4	10	2	9	8	11	7	3	12	5	6	1
F <sub>16</sub>	5.5	11	4	1	7.5	10	9	2	12	5.5	7.5	3
F <sub>17</sub>	3	7	4.5	12	4.5	6	8.5	2	10.5	8.5	10.5	1
F <sub>18</sub>	3	7	2	5	11	10	8	4	9	12	6	1
F <sub>19</sub>	1.5	10	3	6	11	8	7	4	12	9	5	1.5
F <sub>20</sub>	2	12	3	10	5	8	6.5	4	11	9	6.5	1
F <sub>21</sub>	1.5	12	3	7.5	7.5	9.5	9.5	4	11	5.5	5.5	1.5
F <sub>22</sub>	2	5	3	10	4	8	6.5	6.5	12	11	9	1
F <sub>23</sub>	2	12	6.5	8	4.5	6.5	9	3	10	11	4.5	1
F <sub>24</sub>	3	12	1.5	6	7.5	9	7.5	4	10	11	5	1.5
F <sub>25</sub>	2	9	4	6.5	8	10	5	3	11.5	11.5	6.5	1
F <sub>26</sub>	2	12	3.5	5	10	7	8.5	3.5	11	8.5	6	1
F <sub>27</sub>	12	11	1.5	7	5	3	5	8	10	9	5	1.5
F <sub>28</sub>	4	10	3	5	11	2	8	6.5	12	9	6.5	1
F <sub>29</sub>	2	11	3.5	10	5	8	8	3.5	12	6	8	1
F <sub>30</sub>	2	11	3	6	7	8	4	9	10	12	5	1
Average Ranking	3.116	9.483	5.5	6.862	6.724	8.017	5.914	4.103	10.621	9.603	6.655	1.362
Rank	2	10	4	8	7	9	5	3	12	11	6	1

**Table 3:** Aligned Friedman test ranks for the compared algorithms over 30 CEC2017 functions

Function	IHHO	HHO	DE	GOA	GWO	MFO	MVO	PSO	WOA	SCA	FA	COWOAFS-SVM
F <sub>1</sub>	2	7	347	5	9	8	3	4	346	348	6	1
F <sub>3</sub>	56.5	63	327	58.5	323	334	58.5	61	328	326	60	56.5
F <sub>4</sub>	142	224.5	210	175	163	189	149	156	254	277	182	145
F <sub>5</sub>	138	213	242	191	174	205	168	195	239	245	197	134
F <sub>6</sub>	150.5	234	217	172	157	161	164	178	230	212	271	150.5
F <sub>7</sub>	192	264	267	144	154.5	184	140	136	244	248	154.5	130
F <sub>8</sub>	153	198.5	251	204	167	198.5	190	173	229	231	207	137
F <sub>9</sub>	158	310	318	89.5	80	286	78.5	78.5	313	92	89.5	82
F <sub>10</sub>	81	293	302	266	88	292	97	75	309	317	228	74
F <sub>11</sub>	114	159.5	127.5	185	300	272	127.5	104.5	265	281	159.5	104.5
F <sub>12</sub>	13	19	12	17	14	16	15	11	344	345	18	10
F <sub>13</sub>	43	331	40	324	54	321	45	46	314	337	53	39
F <sub>14</sub>	66	70	68	311	316	333	73	69	319	72	71	67
F <sub>15</sub>	52	329	49	322	303	335	65	51	336	55	62	48
F <sub>16</sub>	290.5	308	278	64	298.5	305	301	235	312	290.5	298.5	258
F <sub>17</sub>	123	224.5	182	270	182	206	240.5	113	262.5	240.5	262.5	103
F <sub>18</sub>	38	76	36	47	332	325	84	41	320	338	50	35
F <sub>19</sub>	28.5	44	30	33	330	37	34	31	339	42	32	28.5
F <sub>20</sub>	99.5	294	112	260.5	152	237.5	222	122	285	249.5	222	95
F <sub>21</sub>	101.5	282	129	226.5	226.5	252.5	252.5	162	273	208.5	208.5	101.5
F <sub>22</sub>	111	143	118	257	131	216	170.5	170.5	297	283	233	98
F <sub>23</sub>	126	280	202.5	222	179.5	202.5	237.5	135	249.5	260.5	179.5	99.5
F <sub>24</sub>	120	288	108	200.5	219	236	219	132.5	259	268.5	176.5	108
F <sub>25</sub>	117	243	169	214.5	232	256	196	141	275.5	275.5	214.5	91
F <sub>26</sub>	85	315	86.5	94	306	108	246.5	86.5	307	246.5	96	77
F <sub>27</sub>	284	279	120	176.5	147	132.5	147	200.5	268.5	219	147	120

(Continued)

**Table 3:** Continued

Function	IHHO	HHO	DE	GOA	GWO	MFO	MVO	PSO	WOA	SCA	FA	COWOAFS-SVM
F <sub>28</sub>	139	287	125	166	289	93	255	193.5	296	274	193.5	83
F <sub>29</sub>	110	295	115.5	211	124	187	187	115.5	304	165	187	106
F <sub>30</sub>	21	342	22	25	26	27	23	340	341	343	24	20
Average Ranking	108.79	221.24	159	170.05	188.86	205.36	144.45	122.66	291.69	244.48	147.69	89.72
Rank	2	10	6	7	8	9	4	3	12	11	5	1

Finally, the Iman and Davenport test has also been conducted, as it can provide better results than  $\chi^2$ . The overview of the Iman and Davenport results is shown in [Tab. 4](#). The obtained result of 38.39 was compared against the F-distribution critical value ( $F(9,9e10) = 1.82$ ) and indicate that the result is significantly higher.

**Table 4:** Result summary of the Friedman and Iman-Davenport statistical test ( $\alpha = 0.05$ )

Friedman value	$\chi^2$ critical value	$p$ -value	Iman–Davenport value	$F$ critical value
$1.844 \times 10^{+2}$	$1.968 \times 10^{+1}$	$1.110 \times 10^{-16}$	$3.839 \times 10^{+1}$	1.820

Additionally, Friedman statistics (184.40) is larger than the  $\chi^2$  critical value with ten degrees of freedom (1.82) at the level of significance 0.05. As the result, null hypothesis can be rejected, indicating that the proposed algorithm performs significantly better than other methods in the comparative analysis.

As the null hypothesis has been rejected by both statistical tests that were executed, the non-parametric Holm step-down procedure was also performed, and results are shown in [Tab. 5](#).

**Table 5:** Holm step-down procedure results

Comparison	$p$ values	Ranking	$\alpha = 0.05$	$\alpha = 0.1$	H1	H2
COWOAFS-SVM vs. HHO	0	0	0.00455	0.00909	True	True
COWOAFS-SVM vs. WOA	0	1	0.00500	0.01000	True	True
COWOAFS-SVM vs. SCA	0	2	0.00556	0.01111	True	True
COWOAFS-SVM vs. MFO	$1.04 \times 10^{-12}$	3	0.00625	0.01250	True	True
COWOAFS-SVM vs. GOA	$3.15 \times 10^{-9}$	4	0.00714	0.01429	True	True
COWOAFS-SVM vs. GWO	$7.44 \times 10^{-9}$	5	0.00833	0.01667	True	True

(Continued)

**Table 5:** Continued

Comparison	$p$ values	Ranking	$\alpha = 0.05$	$\alpha = 0.1$	H1	H2
COWOAFS-SVM vs. FA	$1.13 \times 10^{-8}$	6	0.01000	0.02000	True	True
COWOAFS-SVM vs. MVO	$7.65 \times 10^{-7}$	7	0.01250	0.02500	True	True
COWOAFS-SVM vs. DE	$6.21 \times 10^{-6}$	8	0.01667	0.03333	True	True
COWOAFS-SVM vs. PSO	0.001894	9	0.02500	0.05000	True	True
COWOAFS-SVM vs. IHHO	0.029131	10	0.05000	0.10000	True	True

#### 4.2 Hyper Parameters Tuning and Feature Selection Experiments

This section first describes the SVM parameters that have been subjected to the optimisation process, followed by the short description of the datasets, pre-processing and metrics utilised to measure the performances. Lastly, in this section, results of the conducted experiments are presented, with the comparative analysis against other current methods, tested on the same problem instances.

##### 4.2.1 Experimental Setup

For the purpose of this study, all methods included in comparative analysis are implemented and tested with control parameters, as suggested in the original paper [30]. All methods were tested over the course of 150 iterations and the results are average of 10 independent runs, as per the restraints of the experimental setup, suggested in [30]. The experiment was conducted in following the specifications laid out in the proposed methodology, starting with the acquisition, normalisation and preparation of all input datasets for the experiment, following a series of steps, as explained earlier, securing separate training and testing sets, using the former in the proposed algorithm and the later to evaluate the prediction results.

This section proposes the application of the devised COWOAFS-SVM algorithm to improve the efficacy of the SVM model (with and without the feature selection) on three standard benchmark medical diagnostic datasets, as described in the referred paper [5]. The datasets included in the research deal with common illness, namely breast cancer, diabetes and erythematous-squamous (ES) disease.

Similarly as in [5], only two SVM hyper-parameters were considered: penalty parameter (C) and kernel bandwidth of kernel function ( $\gamma$ ). Each agent is encoded as an array of two elements (C and  $\gamma$ ). The search boundaries for C parameter are in range  $\{2^{-7.5}, 2^{-3.5}, \dots, 2^{7.5}\}$ , and for  $\gamma$  in range  $\{2^{-7.5}, 2^{-3.5}, \dots, 2^{7.5}\}$ .

This article presents the experimental results of the modified WOA on several datasets, inherited from the original study. Datasets used for this experiment cover diseases affecting many people, such as breast cancer, diabetes and erythematous-squamous. Breakthroughs in diagnostic and detection methods for these diseases are unquestionably beneficial. All features commonly extracted from input sources, used in the diagnostic of these diseases are available in digitised forms from different databases:

- The breast cancer dataset was acquired from the Wisconsin breast cancer database.

- The database dataset was acquired from Pima Indian heritage and contains anonymized data about their female patients.
- The dermatology database has clinical and histopathological data, acquired, normalised, and prepared in the same manner as explained in [30].

The original study gives full descriptions and details of all three datasets.

Values from these datasets are normalised to  $[-1, 1]$  to avoid calculation issues and dominance and drag that much lower or much greater values may have on results. This normalisation strategy is performed on the dataset at the very beginning. Each dataset was split into ten segments to ensure more reliable and stable experimental results. Each run was repeated ten times to improve these results.

The final results are averaged. To assure comparability of the results of this study with the results reported in the earlier study, this setup was limited to the same parameters of the population, iteration count, repetitions and segmentation of the datasets. This way, the hardware impact is eliminated and only the final performance of the optimisation of the target function is measured.

The algorithm performance evaluation was inherited from the original research [30], to ensure result comparability. These metrics are:

$$Accuracy = \frac{100\% (TP + TN)}{TP + FP + FN + TN} \quad (16)$$

$$Sensitivity = \frac{100\% TP}{TP + FN} \quad (17)$$

$$Specificity = \frac{100\% TN}{FP + TN} \quad (18)$$

In Eqs. (16)–(18), formulas are given for the classification accuracy, and measurements are the sensitivity and the specificity, all referring to the tested algorithm. Values  $TP$ ,  $TN$  and  $FN$  are numbers of true positives, true negatives and false negatives, respectively. All values are expressed in percentage.

#### 4.2.2 Experimental Results

The results of this experimental research are shown in this section. The performances of the proposed COWOAFS-SVM method were compared to the results published in [30]. To provide valid grounds for the comparisons, this research utilised the same experimental setup and datasets as in [30]. The results for MWOA-SVM, WOA-SVM, BFO-SVM, PSO-SVM, GA-SVM and pure SVM were also taken from [30].

##### *Results Pertaining to Breast Cancer Diagnostics*

The breast cancer dataset is comprised of 699 entries, 458 of which were benign, while 241 were malignant. The performances of the proposed COWOAFS-enhanced SVM method without feature selection are shown in Tab. 6 and with feature selection in Tab. 7. The results show average values obtained through 10 independent runs of ten-fold CV. The results indicate that the SVM structure generated by the COWOAFS-SVM algorithm outperforms all other SVM approaches, in terms of accuracy, area under curve, sensitivity and specificity. COWOAFS-SVM achieved average accuracy of 96.84%, together with the standard deviation that is smaller than the results of the MWOA-SVM and other compared methods, indicating that the COWOAFS-SVM is capable of producing consistent results.



**Table 6:** Breast cancer-results without feature selection

Methods	Metrics				
	(C, $\gamma$ )	Accuracy	Area under ROC	Sensitivity	Specificity
MWOA-SVM	(31, 0.0543)	0.9681 $\pm$ 0.0013	0.9665 $\pm$ 0.0025	0.9707 $\pm$ 0.0013	0.9632 $\pm$ 0.0063
WOA-SVM	(28.79, 0.2166)	0.9665 $\pm$ 0.0018	0.9644 $\pm$ 0.0030	0.9705 $\pm$ 0.0016	0.9573 $\pm$ 0.0064
BFO-SVM	(22.342, 0.5456)	0.9562 $\pm$ 0.0096	0.9528 $\pm$ 0.0069	0.9651 $\pm$ 0.0212	0.9404 $\pm$ 0.0201
PSO-SVM	(11.779, 0.075)	0.9604 $\pm$ 0.0157	0.9586 $\pm$ 0.0110	0.9585 $\pm$ 0.0307	0.9645 $\pm$ 0.0141
GA-SVM	(25.557, 0.452)	0.9545 $\pm$ 0.0099	0.9538 $\pm$ 0.0083	0.9591 $\pm$ 0.0188	0.9484 $\pm$ 0.0211
SVM	(1.000, 0.00000)	0.9319 $\pm$ 0.0176	0.9424 $\pm$ 0.0094	0.9434 $\pm$ 0.0193	0.9354 $\pm$ 0.0287
<b>COWOAFS-SVM</b>	<b>(30.5, 0.0678)</b>	<b>0.9684 <math>\pm</math> 0.0012</b>	<b>0.9671 <math>\pm</math> 0.0027</b>	<b>0.9711 <math>\pm</math> 0.0012</b>	<b>0.9644 <math>\pm</math> 0.0061</b>

**Table 7:** Breast cancer-results with feature selection

Methods	Metrics				
	(C, $\gamma$ )	Accuracy	Area under ROC	Sensitivity	Specificity
MWOA-SVM	(31, 0.0678)	0.9697 $\pm$ 0.0008	0.9702 $\pm$ 0.0025	0.9693 $\pm$ 0.0022	0.9710 $\pm$ 0.0071
WOA-SVM	(24.567, 0.221)	0.9678 $\pm$ 0.0014	0.9657 $\pm$ 0.0025	0.9662 $\pm$ 0.0014	0.9641 $\pm$ 0.0081
BFO-SVM	(16.451, 0.065)	0.9573 $\pm$ 0.0082	0.9548 $\pm$ 0.0052	0.9674 $\pm$ 0.0192	0.9532 $\pm$ 0.0182
PSO-SVM	(18.232, 0.562)	0.9655 $\pm$ 0.0097	0.9643 $\pm$ 0.0104	0.9603 $\pm$ 0.0246	0.9677 $\pm$ 0.0120
GA-SVM	(21.5612, 0.39)	0.9634 $\pm$ 0.0084	0.9567 $\pm$ 0.0074	0.9652 $\pm$ 0.0136	0.9545 $\pm$ 0.0189
<b>COWOAFS-SVM</b>	<b>(30.56, 0.0685)</b>	<b>0.9704 <math>\pm</math> 0.0006</b>	<b>0.9708 <math>\pm</math> 0.0020</b>	<b>0.9698 <math>\pm</math> 0.0014</b>	<b>0.9715 <math>\pm</math> 0.0065</b>

Similarly as in [30], the performances of the proposed COWOAFS-SVM method were further enhanced by performing the optimisation of the SVM structure and feature selection together. The proposed method once again outperformed all competitors, by achieving the accuracy of 97.04%, the best values for area under curve, sensitivity and specificity, while having the smallest standard deviation.

#### *Results Pertaining to Diabetes Diagnostics*

The second dataset related to the diabetes diagnostics consisted of 768 entries, 268 belonging to the diabetic patients, while 500 were normal healthy instances. The results of the proposed approach without feature selection are shown in Tab. 8. It can be clearly seen that the proposed COWOAFS-SVM approach outperformed all other competitors, for which the results were taken from paper [30]. The average metrics of ten runs show the accuracy of 77.59%, which is slightly better than MWOA-SVM, and significantly better than other compared methods. Other metrics are also in favour of the proposed COWOAFS-SVM method, including the smallest standard deviation values.

**Table 8:** Diabetes-results without feature selection

Methods	Metrics				
	(C, $\gamma$ )	Accuracy	Area under ROC	Sensitivity	Specificity
MWOA-SVM	(35, 0.0639)	0.7745 $\pm$ 0.0033	0.7232 $\pm$ 0.0015	0.5731 $\pm$ 0.0092	0.9012 $\pm$ 0.0075
WOA-SVM	(27.688, 0.542)	0.7685 $\pm$ 0.0046	0.7213 $\pm$ 0.0040	0.5431 $\pm$ 0.0102	0.8901 $\pm$ 0.0053
BFO-SVM	(19.621, 0.032)	0.7604 $\pm$ 0.0059	0.7121 $\pm$ 0.0074	0.5382 $\pm$ 0.0131	0.8861 $\pm$ 0.0067
PSO-SVM	(15.321, 0.934)	0.7586 $\pm$ 0.0057	0.7146 $\pm$ 0.0046	0.5418 $\pm$ 0.0135	0.8874 $\pm$ 0.0071
GA-SVM	(25.542, 0.45)	0.7589 $\pm$ 0.0061	0.7114 $\pm$ 0.0062	0.5412 $\pm$ 0.0113	0.8816 $\pm$ 0.0074
<b>COWOAFS-SVM</b>	<b>(34.4, 0.0651)</b>	0.7759 $\pm$ 0.0031	0.7266 $\pm$ 0.0012	0.5793 $\pm$ 0.0098	0.9048 $\pm$ 0.0065

Tab. 9 depicts the results obtained by the proposed COWOAFS-SVM approach with feature selection, from ten runs of ten-fold CV. Once again, the results clearly indicate that the proposed method outperformed other Metaheuristic based approaches in terms of accuracy, area under curve, sensitivity and specificity, while having the smallest values of the standard deviation.

**Table 9:** Diabetes-results with feature selection

Methods	Metrics				
	(C, $\gamma$ )	Accuracy	Area under ROC	Sensitivity	Specificity
MWOA-SVM	(28, 0.0548)	0.7800 $\pm$ 0.0031	0.7312 $\pm$ 0.0019	0.5809 $\pm$ 0.0064	0.9042 $\pm$ 0.0065
WOA-SVM	(24.567, 0.431)	0.7745 $\pm$ 0.0042	0.7248 $\pm$ 0.0034	0.5552 $\pm$ 0.0412	0.8968 $\pm$ 0.0041
BFO-SVM	(18.447, 0.076)	0.7635 $\pm$ 0.0039	0.7211 $\pm$ 0.0061	0.5432 $\pm$ 0.0114	0.8892 $\pm$ 0.0054
PSO-SVM	(16.467, 0.873)	0.7612 $\pm$ 0.0037	0.7135 $\pm$ 0.0038	0.5469 $\pm$ 0.0126	0.8946 $\pm$ 0.0051
GA-SVM	(21.661, 0.28)	0.7609 $\pm$ 0.0041	0.7135 $\pm$ 0.0058	0.5492 $\pm$ 0.0110	0.8957 $\pm$ 0.0031
<b>COWOAFS-SVM</b>	<b>(28, 0.0539)</b>	0.7822 $\pm$ 0.0025	0.7351 $\pm$ 0.0016	0.5826 $\pm$ 0.0055	0.9079 $\pm$ 0.0062

#### *Results Pertaining to ES Disease Diagnostics*

The ES disease dataset consisted of 358 instances with 34 features and six classes of diseases. As in paper [30], the only metric that was utilised is the accuracy, because of the multi-classification nature of the challenge. Tabs. 10 and 11 contain results attained using the proposed COWOAFS-SVM method compared to competitor approaches, averaged on 10 runs of 10 ten-fold CV, without and with feature selection, respectively.

In both cases, the proposed method outperformed other competitors both in terms of highest accuracy achieved and smallest standard deviation, indicating consistent performances of the approach.

**Table 10:** ES-results without feature selection

Methods	Metrics	
	Accuracy	(C, $\gamma$ )
MWOA-SVM	$0.9865 \pm 0.0026$	(18, 0.031)
WOA-SVM	$0.9804 \pm 0.0031$	(22.37, 0.341)
BFO-SVM	$0.9754 \pm 0.0045$	(19.35, 0.089)
PSO-SVM	$0.9760 \pm 0.0049$	(17.81, 0.954)
GA-SVM	$0.9700 \pm 0.0053$	(15.12, 0.531)
SVM	$0.9643 \pm 0.0104$	(1.00, 0.00)
<b>COWOAFS-SVM</b>	$0.9874 \pm 0.0023$	<b>(17.5, 0.039)</b>

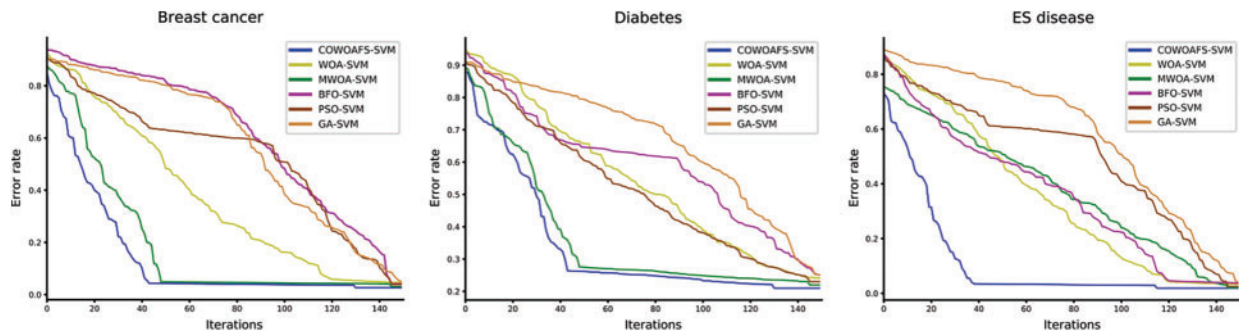
**Table 11:** ES-results with feature selection

Methods	Metrics	
	Accuracy	(C, $\gamma$ )
MWOA-SVM	$0.9887 \pm 0.0026$	(21, 0.225)
WOA-SVM	$0.9862 \pm 0.0027$	(28.47, 0.031)
BFO-SVM	$0.9832 \pm 0.0037$	(17.65, 0.097)
PSO-SVM	$0.9801 \pm 0.0035$	(18.72, 0.831)
GA-SVM	$0.9731 \pm 0.0041$	(19.29, 0.049)
<b>COWOAFS-SVM</b>	$0.9893 \pm 0.0024$	<b>(20.8, 0.232)</b>

### 4.3 Discussion

The presented research proposed a novel WOA method, enhanced with chaotic initialisation, opposition-based learning and firefly search, which was later utilised to improve the performances of the SVM for medical diagnostics. The experimental findings listed in the preceding section plainly show that the proposed COWOAFS-SVM enhanced SVM model obtained superior performances than other modern approaches subjected to the comparative analysis [5]. The proposed COWOAFS-SVM is capable of constructing a superior SVM with higher accuracy and consistency than other competitor methods. It is particularly interesting to compare the results of the proposed COWOAFS-SVM against the CMWOAFS-SVM, which obtained the best results in [5]. It is possible to conclude from the presented results that the proposed COWOAFS-SVM achieved significantly enhanced performances than CMWOAFS-SVM on all datasets used in the experiments. The improvement were observed both for the accuracy and other indicators (area under curve, sensitivity and specificity), as well as for the consistency of the results, that is reflected in the smaller average standard deviation.

To gain better insights into methods' performance, convergence speed graphs for all methods included in comparative analysis with respect to accuracy are shown in Fig. 2.



**Figure 2:** Average convergence speed graphs for classification error rate metrics for 150 iterations

## 5 Conclusion

In this paper, an improved model for simultaneous feature selection and hyper-parameter tuning for SVM is proposed. The algorithm that was utilised for improving SVM performances was based on the WOA SI metaheuristics. The proposed method was then used to improve the SVM for medical diagnostics of breast cancer, diabetes, and erythemato-squamous illnesses. COWOAFS-SVM was evaluated and compared to other state-of-the-art metaheuristics based approaches, and achieved superior performances both in terms of higher accuracy and consistency reflected through smaller values of the standard deviation on all three observed datasets. The achieved promising results indicate the possible direction of the future research, which will focus on validating the proposed model on other datasets from different application domains. The other direction of the possible future work is to apply the proposed COWOAFS-SVM metaheuristics in solving other challenging problems from wide spectrum of application domains, including cloud computing, optimisation of the convolutional neural networks, and wireless sensor networks.

**Funding Statement:** The authors state that they have not received any funding for this research.

**Conflicts of Interest:** The authors state that they have no conflicts of interest regarding this research.

## References

- [1] J. C. Spall, "Stochastic Optimization," in *Handbook of Computational statistics*, Berlin, Heidelberg: Springer, pp. 173–201, 2012.
- [2] Z. Beheshti and S. M. H. Shamsuddin, "A review of population-based meta-heuristic algorithms," *International Journal of Advances in Soft Computing and Its Applications*, vol. 5, pp. 1–35, 2013.
- [3] L. T. Biegler and I. E. Grossmann, "Retrospective on optimization," *Computers & Chemical Engineering*, vol. 28, pp. 1169–1192, 2004.
- [4] S. Desale, A. Rasool, S. Andhale and P. Rane, "Heuristic and meta-heuristic algorithms and their relevance to the real world: A survey," *International Journal of Computer Engineering in Research Trends*, vol. 351, pp. 2349–7084, 2015.
- [5] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: Sparrow search algorithm," *Systems Science & Control Engineering*, vol. 8, pp. 22–34, 2020.
- [6] X. Dai, S. Long, Z. Zhang and D. Gong, "Mobile robot path planning based on ant colony algorithm with A heuristic method," *Frontiers in Neurorobotics*, vol. 13, pp. 15, 2019.
- [7] W. Yong, W. Tao, Z. Cheng-Zhi and H. Hua-Juan, "A new stochastic optimization approach-dolphin swarm optimization algorithm," *International Journal of Computational Intelligence and Applications*, vol. 15, pp. 1650011, 2016.

- [8] G. Dhiman and V. Kumar, "Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications," *Advances in Engineering Software*, vol. 114, pp. 48–70, 2017.
- [9] M. Mavrovouniotis, C. Li and S. Yang, "A survey of swarm intelligence for dynamic optimization: Algorithms and applications," *Swarm and Evolutionary Computation*, vol. 33, pp. 1–17, 2017.
- [10] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. of ICNN'95-Int. Conf. on Neural Networks*, Perth, Australia, 1995.
- [11] M. Dorigo, V. Maniezzo and A. Coloni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996.
- [12] P. Pinto, T. A. Runkler and J. M. Sousa, "Wasp swarm optimization of logistic systems," in *Adaptive and Natural Computing Algorithms*, Vienna: Springer, 2005.
- [13] J. Bishop, "Stochastic searching networks," in *Proc. First IEE Int. Conf. on Artificial Neural Networks*, London, UK, No. 313, pp. 329–331, 1989.
- [14] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," *Computer Science Journal*, vol. 200, pp. 1–10, 2005.
- [15] X. S. YANG, "Nature Inspired Metaheuristic Algorithms," Bristol, UK: Luniver Press, 2008. [Online]. Available: <https://dl.acm.org/doi/book/10.5555/2655295>.
- [16] M. Seyedali and L. Andrew, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [17] T. Hendtlass, "WoSP: A multi-optima particle swarm algorithm," in *Proc. IEEE Congress on Evolutionary Computation*, Edinburgh, UK, Vol. 1, pp. 727–734, 2005.
- [18] A. H. Gandomi and A. H. Alavi, "Krill herd: A new bioinspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4831–4845, 2012.
- [19] A. Kaveh and N. Farhoudi, "A new optimization method: Dolphin echolocation," *Advances in Engineering Software*, vol. 59, pp. 53–70, 2013.
- [20] L. Cao, Y. Cai and Y. Yue, "Swarm intelligence-based performance optimization for mobile wireless sensor networks: Survey, challenges, and future directions," *IEEE Access*, vol. 7, pp. 161524–161553, 2019.
- [21] A. Nayyar and N. G. Nguyen, "Introduction to swarm intelligence," *Advances in Swarm Intelligence for Optimizing Problems in Computer Science*, pp. 53–78, 2018. <https://www.taylorfrancis.com/chapters/edit/10.1201/9780429445927-3/introduction-swarm-intelligence-anand-nayyar-nhu-gia-nguyen>.
- [22] B. Jakimovski, B. Meyer and E. Maehle, "Swarm intelligence for self-reconfiguring walking robot," in *2008 IEEE Swarm Intelligence Symp.*, St. Louis, MO, USA, pp. 1–8, 2008.
- [23] D. Liu and B. Wang, "Biological swarm intelligence based opportunistic resource allocation for wireless ad Hoc networks," *Wireless Personal Communications*, vol. 66, pp. 629–649, 2011.
- [24] Y. Qawqzeh, M. T. Alharbi, A. Jaradat and K. N. Abdul Sattar, "A review of swarm intelligence algorithms deployment for scheduling and optimization in cloud computing environments," *PeerJ Computer Science*, vol. 7, pp. e696, 2021.
- [25] S. Naka, T. Genji, T. Yura and Y. Fukuyama, "A hybrid particle swarm optimization for distribution state estimation," *IEEE Transactions on Power Systems*, vol. 18, pp. 60–68, 2003.
- [26] S. Yuan, S. Wang and N. Tian, "Swarm intelligence optimization and its application in geophysical data inversion," *Applied Geophysics*, vol. 6, pp. 166–174, 2009.
- [27] W. Zhao, Z. Zhang and L. Wang, "Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications," *Engineering Applications of Artificial Intelligence*, vol. 87, pp. 103300, 2020.
- [28] S. Mishra, R. Sagban, A. Yakoob and N. Gandhi, "Swarm intelligence in anomaly detection systems: An overview," *International Journal of Computers and Applications*, vol. 43, pp. 109–118, 2021.
- [29] R. Olivares, F. Muñoz and F. Riquelme, "A Multi-objective linear threshold influence spread model solved by swarm intelligence-based methods," *Knowledge-Based Systems*, vol. 212, pp. 106623, 2021.
- [30] M. Wang and H. Chen, "Chaotic multi-swarm whale optimizer boosted support vector machine for medical diagnosis," *Applied Soft Computing*, vol. 88, pp. 105946, 2020.

- [31] C. A. de Pinho Pinheiro, N. Nedjah and L. de Macedo Mourelle, "Detection and classification of pulmonary nodules using deep learning and swarm intelligence," *Multimedia Tools and Applications*, vol. 79, pp. 15437–15465, 2019.
- [32] B. H. Nguyen, B. Xue and M. Zhang, "A survey on swarm intelligence approaches to feature selection in data mining," *Swarm and Evolutionary Computation*, vol. 54, pp. 100663, 2020.
- [33] M. Jahandideh-Tehrani, O. Bozorg-Haddad and H. A. Loáiciga, "A review of applications of animal-inspired evolutionary algorithms in reservoir operation modelling," *Water and Environment Journal*, vol. 35, pp. 628–646, 2021.
- [34] L. Torres Treviño, "A 2020 taxonomy of algorithms inspired on living beings behavior," 2021. [Online]. Available: <https://arxiv.org/>.
- [35] Q. Zhang, H. Chen, A. A. Heidari, X. Zhao, Y. Xu *et al.*, "Chaos-induced and mutation-driven schemes boosting salp chains-inspired optimizers," *IEEE Access*, vol. 7, pp. 31243–31261, 2019.
- [36] T. Bezdan, A. Petrovic, M. Zivkovic, I. Strumberger, V. K. Devi *et al.*, "Current best opposition-based learning salp swarm algorithm for global numerical optimization," in *Proc. Zooming Innovation in Consumer Technologies Conf.*, Novi Sad, Serbia, pp. 5–10, 2021.
- [37] N. Bacanin, A. Petrovic, M. Zivkovic, T. Bezdan and A. Chhabra, "Enhanced salp swarm algorithm for feature selection," in *Intelligent and Fuzzy Techniques for Emerging Conditions and Digital Transformation*, Cham: Springer, 2021.
- [38] M. Zivkovic, T. Zivkovic, K. Venkatachalam and N. Bacanin, "Enhanced dragonfly algorithm adapted for wireless sensor network lifetime optimization," in *Proc. Data Intelligence and Cognitive Informatics*, Springer, Singapore, pp. 803–817, 2021.
- [39] M. Zivkovic, N. Bacanin, E. Tuba, I. Strumberger, T. Bezdan *et al.*, "Wireless sensor networks life time optimization based on the improved firefly algorithm," in *Proc Int. Wireless Communications and Mobile Computing*, Harbin, China, pp. 1176–1181, 2020.
- [40] N. Bacanin, E. Tuba, M. Zivkovic, I. Strumberger and M. Tuba, "Whale optimization algorithm with exploratory move for wireless sensor networks localization," in *Proc. Int. Conf. on Hybrid Intelligent Systems*, Springer, Porto, Portugal, pp. 328–338, 2019.
- [41] T. Bezdan, M. Zivkovic, M. Antonijevec, T. Zivkovic and N. Bacanin, "Enhanced flower pollination algorithm for task scheduling in cloud computing environment," in *Proc. Machine Learning for Predictive Analysis*, Springer, Singapore, pp. 163–171, 2020.
- [42] M. Zivkovic, T. Bezdan, I. Strumberger, N. Bacanin and K. Venkatachalam, "Improved harris hawks optimization algorithm for workflow scheduling challenge in cloud-edge environment," in *Computer Networks, Big Data and IoT*, Springer, Singapore, pp. 87–102, 2021.
- [43] N. Bacanin, N. Vukobrat, M. Zivkovic, T. Bezdan and I. Strumberger, "Improved harris hawks optimization adapted for artificial neural network training," in *Pro. Int. Conf. on Intelligent and Fuzzy Systems*, Springer, Cham, pp. 281–289, 2021.
- [44] I. Strumberger, E. Tuba, N. Bacanin, M. Zivkovic, M. Beko *et al.*, "Designing convolutional neural network architecture by the firefly algorithm," in *Proc. 2019 Int. Young Engineers Forum (YEF-ECE)*, Lisboa, Portugal, pp. 59–65, 2019.
- [45] S. Milosevic, T. Bezdan, M. Zivkovic, N. Bacanin, I. Strumberger *et al.*, "Feed-forward neural network training by hybrid bat algorithm," in *Int. Conf. on Modelling and Development of Intelligent Systems*, Springer, Cham, pp. 52–66, 2021.
- [46] T. Bezdan, C. Stoean, A. A. Naamany, N. Bacanin, T. A. Rashid *et al.*, "Hybrid fruit-fly optimization algorithm with K-means for text document clustering," *Mathematics*, vol. 9, pp. 19–29, 2021.
- [47] N. Bacanin, T. Bezdan, M. Zivkovic and A. Chhabra, "Weight optimization in artificial neural network training by improved monarch butterfly algorithm," in *Mobile Computing and Sustainable Informatics*, Springer, Singapore, pp. 397–409, 2022.
- [48] L. Gajic, D. Cvetnic, M. Zivkovic, T. Bezdan, N. Bacanin *et al.*, "Multi-layer perceptron training using hybridized bat algorithm," in *Computational Vision and Bio-Inspired Computing*, Singapore, Springer, pp. 689–705, 2021.

- [49] N. Bacanin, K. Alhazmi, M. Zivkovic, K. Venkatachalam, T. Bezdán *et al.*, “Training multi-layer perceptron with enhanced brain storm optimization metaheuristics,” *Computers Materials & Continua*, vol. 70, no. 2, pp. 4199–4215, 2022.
- [50] M. Zivkovic, N. Bacanin, K. Venkatachalam, A. Nayyar, A. Djordjevic *et al.*, “COVID-19 cases prediction by using hybrid machine learning and beetle antennae search approach,” *Sustainable Cities and Society*, vol. 66, pp. 102669, 2021.
- [51] M. Zivkovic, K. Venkatachalam, N. Bacanin, A. Djordjevic, M. Antonijevic *et al.*, “Hybrid genetic algorithm and machine learning method for COVID-19 cases prediction,” in *Proc. Int. Conf. on Sustainable Expert Systems*, Springer, Singapore, pp. 169–184, 2021.
- [52] T. Bezdán, M. Zivkovic, E. Tuba, I. Strumberger, N. Bacanin *et al.*, “Glioma brain tumor grade classification from MRI using convolutional neural networks designed by modified FA,” in *Proc. Int. Conf. on Intelligent and Fuzzy Systems*, Istanbul, Turkey, pp. 955–963, 2020.
- [53] T. Bezdán, S. Milosevic, K. Venkatachalam, M. Zivkovic, N. Bacanin *et al.*, “Optimizing convolutional neural network by hybridized elephant herding optimization algorithm for magnetic resonance image classification of glioma brain tumor grade,” in *Proc. 2021 Zooming Innovation in Consumer Technologies Conf. (ZINC)*, IEEE, Serbia, pp. 171–176, 2021.
- [54] J. Basha, N. Bacanin, N. Vukobrat, M. Zivkovic, K. Venkatachalam *et al.*, “Chaotic harris hawks optimization with quasi-reflection-based learning: An application to enhance CNN design,” *Sensors*, vol. 21, pp. 6654, 2021.
- [55] N. Rana, M. S. Abd Latiff and H. Chiroma, “Whale optimization algorithm: A systematic review of contemporary applications, modifications and developments,” *Neural Computing and Applications*, vol. 32, pp. 1–33, 2020.
- [56] G. Zhenyu, C. Bo, Y. Min and C. Binggang, “Self-adaptive chaos differential evolution,” *Advances in Natural Computation*, vol. 4221, pp. 972–975, 2006.
- [57] M. A. El-Shorbagy, A. A. Mousa and S. M. Nasr, “A Chaos-based evolutionary algorithm for general nonlinear programming problems,” *Chaos, Solutions & Fractals*, vol. 85, pp. 8–21, 2016.
- [58] I. Fister, M. Perc and S. M. Kamal, “A review of chaos-based firefly algorithms: Perspectives and research challenges,” *Applied Mathematics and Computation*, vol. 252, pp. 155–165, 2015.
- [59] X. Lu, Q. Wu, Y. Zhou, Y. Ma, C. Song *et al.*, “A dynamic swarm firefly algorithm based on chaos theory and Max-min distance algorithm,” *Traitement du Signal*, vol. 36, pp. 227–231, 2019.
- [60] N. Bacanin, R. Stoean, M. Zivkovic, A. Petrovic, T. A. Rashid *et al.*, “Performance of a novel chaotic firefly algorithm with enhanced exploration for tackling global optimization problems: Application for dropout regularization,” *Mathematics*, vol. 9, no. 21, pp. 1–33, 2021.
- [61] R. Caponetto, L. Fortuna, S. Fazzino and M. Xibilia, “Chaotic sequences to improve the performance of evolutionary algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 3, pp. 289–304, 2003.
- [62] H. R. Tizhoosh, “Opposition-based learning: A new scheme for machine intelligence,” in *Int. Conf. on Computational Intelligence for Modelling, Control and Automation and International Conf. on Intelligent Agents, web Technologies and Internet Commerce (CIMCA-IAWTIC’06)*, Vienna, Austria, IEEE, vol. 1, pp. 695–701, 2005.
- [63] N. Awad, M. Ali, J. Liang, B. Qu, P. Suganthan *et al.*, “Evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization,” *Technology Report*, 2016.
- [64] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja *et al.*, “Harris hawks optimization: Algorithm and applications,” *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.
- [65] A. G. Hussien and M. Amin, “A Self-adaptive harris hawks optimization algorithm with opposition-based learning and chaotic local search strategy for global optimization and feature selection,” *International Journal of Machine Learning and Cybernetics*, vol. 12, pp. 1–28, 2021.
- [66] A. K. Qin, V. L. Huang and P. N. Suganthan, “Differential evolution algorithm with strategy adaptation for global numerical optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.

- [67] S. Z. Mirjalili, S. Mirjalili, S. Saremi, H. Faris and I. Aljarah, "Grasshopper optimization algorithm for multi-objective optimization problems," *Applied Intelligence*, vol. 48, no. 4, pp. 805–820, 2018.
- [68] S. Mirjalili, S. M. Mirjalili and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [69] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.
- [70] S. Mirjalili, S. M. Mirjalili and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, pp. 495–513, 2016.
- [71] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120–133, 2016.