

# IMPROVED SPIDER MONKEY OPTIMIZATION ALGORITHM TO TRAIN MLP FOR DATA CLASSIFICATION

---

**Prabhat Ranjan Singh**

Department of Computer Science, Wuhan University of Technology (China)

E-mail: [prabhatranjansingh68@gmail.com](mailto:prabhatranjansingh68@gmail.com)

**Diallo Moussa**

Department of Computer Science, Wuhan University of Technology (China)

E-mail: [moussdiall@yahoo.com](mailto:moussdiall@yahoo.com)

**Xiong Shengwu**

Department of Computer Science, Wuhan University of Technology (China)

E-mail: [xiongsww@whut.edu.cn](mailto:xiongsww@whut.edu.cn)

**Bikram Prasad Singh**

Department of Computer Science, Wuhan University of Technology (China)

E-mail: [erbikramprasadsingh@gmail.com](mailto:erbikramprasadsingh@gmail.com)

**Recepción:** 05/03/2019 **Aceptación:** 01/04/2019 **Publicación:** 17/05/2019

## **Citación sugerida:**

Singh, P., R., Moussa, D., Shengwu, X. y Singh, B. (2019). Improved Spider Monkey Optimization Algorithm to train MLP for data classification. *3C Tecnología. Glosas de innovación aplicadas a la pyme. Edición Especial, Mayo 2019*, pp. 142–165. doi: <http://dx.doi.org/10.17993/3ctecno.2019.specialissue2.142-165>

## **Suggested citation:**

Singh, P., R., Moussa, D., Shengwu, X. & Singh, B. (2019). Improved Spider Monkey Optimization Algorithm to train MLP for data classification. *3C Tecnología. Glosas de innovación aplicadas a la pyme. Special Issue, May 2019*, pp. 142–165. doi: <http://dx.doi.org/10.17993/3ctecno.2019.specialissue2.142-165>

## ABSTRACT

In this paper, the modified Spider Monkey Optimization (SMO) with Multi-Layer Perceptron (MLP) is utilized to solve the classification problem on five different datasets. The MLP is a widely used Neural Network (NN) variant which requires training on specific application to tackle the slow convergence speed and local minima avoidance. The original SMO with MLP experiences the problem of finding the optimal classification result; due to that, the SMO is enhanced by other meta-heuristics algorithm to train the MLP. Based on the concept of no free lunch theorem, there is always a possibility to improve the algorithm. With the same expectation, the performance of the SMO algorithm is improved by using Differential Evolution (DE) and Grey Wolf Optimizer (GWO) algorithm to train the MLP. Likewise, the SMO-DE and SMO-GWO are two different concepts employed to improve efficiency. The results of proposed algorithms are compared with other well-known algorithms such as BBO, PSO, ES, SVM, KNN, and Logistic Regression. The results show that the proposed algorithm performs better than others or they are more competitive.

## KEYWORDS

Classification, Neural Network, FNN, Swarm Intelligence, Evolutionary algorithm.

## 1. INTRODUCTION

The Swarm-based intelligence is inspired by the social conduct of creatures, birds, fishes, and insects and other real-world scenario. The spider monkey optimization (SMO) algorithm is a new emerging swarm intelligence algorithm has been inspired from the swarm of spider monkey due to their social foraging behavior (Bansal, Sharma, Jadon & Clerc, 2014). It belongs to the fission-fusion social structure based strategy which splits the larger group into sub-groups and vice-versa in order to find food resources (Symington, 1990). These animals live in groups of up to 50 members generally and mostly led by a female member known as the global leader and responsible for supervising the group. These animals travel for searching for food in different directions as well as to compute the travel position from the leader and group member experiences. Each individual updates their positions within a certain range of core area during the day and at night every member get fused together to share the experience of foraging food at their habitat. In case there is a deficiency of food resource the global leader divides the parent group into smaller groups to forage food separately. During the process of foraging, if any members of different sub-group get closer, the males in each sub- group shows aggressiveness and territorial behavior such as whooping or chattering. Spider monkeys communicate upon the availability of food within groups and their sub-groups maintaining distinct territory boundaries without involving in any physical activities. After returning back to their habitat, they reflect gestures to show they are actually part of a large group.

SMO follows self-organization and fission of labor properties for obtaining swarm intelligence behaviors of animals (Symington, 1990). This research gives a reasonable investigation to change these practices into an algorithmic model, for example, there is a defined search space known as a core area in which a group of the individual need to seek food in various directions based on leader position directions and pursue a similar procedure till they reach the optimum. The SMO algorithm has four control parameters: Local Leader Limit, Global Leader Limit, maximum group (MG) and perturbation rate (pr). The Local Leader Limit indicates that if there is no update in the local group leader in a specified number

of times then the point needs to be redirected to a different direction for foraging. After reaching global Leader Limit value without any update, the global leader breaks the group into smaller subgroups. The  $pr$  and  $MG$  are used to specify the control amount of perturbation in the current iteration and a maximum number of groups in the population respectively. Spider monkey optimization has seven major steps including initialization. All seven different steps are Initialization, Local Leader Phase, Global Leader Phase, Local Leader Learning Phase, Global Leader Learning Phase, Local Leader Decision Phase, and Global Leader Decision Phase.

There are many improvements in SMO has been done in recent years for many different optimization problems (Gupta & Deep, 2016; Gupta, Deep & Bansal, 2017; Sharma, Sharma, Panigrahi, Kiran & Kumar, 2016; Singh, Elaziz & Xiong, 2018).

In this paper, the original SMO algorithm is enhanced in two ways to train the MLP. The first way is to improve the SMO local leader efficiency with DE mutation, and crossover operators which intensify the exploitation ability of algorithm (Storn & Price, 1997; Zaharie, 2009). The other ways are to apply the GWO strategy to select the three best leaders to update the position after the processing of the local leader phase. The GWO selection strategy strengthens the exploration ability (Mirjalili, Mirjalili, & Lewis, 2014a).

There are many meta-heuristic algorithms used to train MLP. These are PSO (Bell & Oommen, 2017; Shi & Eberhart, 1945), BBO (Mirjalili, 2019), ACO (Gutjahr, 2007), ES (Erfani & Erfani, 2015; Li, *et al.*, 2015) and GA (Deep & Thakur, 2007b, 2007a) algorithms. The MLP is a variant of NN in the field of soft computing. The NN is prominent research because of its fast convergence speed and low computational cost. The primary concept of NN is inspired by biological neurons of the human brain which was first mathematically modeled by (Ferris & Mangasarian, 1995). In details the MLP presented under FNN is one of type of NN. The FNN receive input from one side and generates output to another side (Fine, 1999). In between, it is connected with neurons in different

layers in one direction. The perceptron used in FNN which has two types, first Single-layer perceptron and the second is Multi-layer perceptron. The single layer perceptron is utilized to solve liner problems while the multi-layer perceptron which has more than one perceptron is suitable for solving non-linear problems.

The MLP with one hidden layer can solve continuous and discontinuous functions both stated in (Mirjalili, Mirjalili, & Lewis, 2014b). As the behavior of the human brain, the MLP is progressed from the learning experiences. The observation of learning concept in NN is to find the proper combination between weights and biases to reduce the error for training and test samples. However, the error is still large in some extended period of time and due to that, it can be trapped at local minima rather than global minima. To overcome these problems the meta-heuristic algorithm plays an important role because of its stochastic nature (Hinchey, Sterritt, & Rouff, 2007).

The rest of the paper is structured as follows: Section 2 describes the background of the existing algorithm. The modified version of the SMO algorithm is discussed in Section 3. The datasets and its results are presented in Section 4. The conclusion and future work are mentioned in Section 5.

## 2. BACKGROUND OF EXISTING ALGORITHMS

### 2.1. THE SPIDER MONKEY OPTIMIZATION (SMO)

In the algorithmic methodology, the group can be divided into subgroups until the predefined maximum group limit ( $MG = 4$ ) meet. Finally, the global leader takes the decision to break the groups or simulate into one based on the maximum limit criteria. The algorithm is processed into seven phases such as Initialization phase, Local Leader Phase, Global Leader Phase, Local Leader Learning Phase, Global Leader Learning Phase, Local Leader Decision Phase, and Global Leader Learning Phase. These phases are given as follows:

### 2.1.1. INITIALIZATION

During the initiation process, there is a single group of all population members where members randomly initialize the position in the search space.

$$P_{i,j} = P_{lb,j} + (P_{ub,j} - P_{lb,j})rand(0,1) \quad (1)$$

Where  $P_{i,j}$  ( $i=1,2,\dots, NP$ ) is the  $i^{th}$  population's member in  $j^{th}$  dimension ( $j=1,2,\dots,D$ ). The NP and D are the total number of population and dimensions respectively. Moreover,  $P_{lb,j}$  and  $P_{ub,j}$  are respectively, the lower and upper bound of the search space in  $j^{th}$  dimensions. And the  $rand(0, 1)$  is representing the uniform random distribution number in  $[0,1]$ .

### 2.1.2. LOCAL LEADER PHASE

In this phase, all members of the group update their position from the experience of the local leader and the experiences of local group members as in (2). In this phase, the update process depends on the value of the perturbation rate ( $pr$ ) which is mentioned in (3).

$$P_{i,j} = P_{i,j} + (PLL_{g,j} - P_{i,j})rand(0,1) + (P_{r,j} - P_{i,j})rand(-1,1) \quad (2)$$

Where  $PLL_{g,j}$ , is the local leader position of  $g^{th}$  group in  $j^{th}$  dimension and  $P_{r,j}$  is the randomly selected member from that group. The value between  $[-1, 1]$  is uniformly distributed random number defined as  $rand(-1,1)$  that shows the dispersion or attraction towards  $P_{r,j}$ .

$$pr_{it} = pr_{it} + 0.4 / maxit \quad (3)$$

The  $pr_{it}$  is the perturbation rate value in the current iteration in (3), (where  $pr_1=0.1$ ).

In Algorithm 1, all details of the method of local leader have been mentioned. The  $tgr$  is the total number of the group created up to now. The denoted  $gr[lo]$  and  $gr[hi]$  is the range of population members belong to the current group. As for the value of the function,  $P_{newj}$  gives the update of the new position  $P_{ij}$ . The lowest function value is considered a better in the case of minimization. If the condition is not satisfied then the position of the update will fail.

### 2.1.3. GLOBAL LEADER PHASE

The global leader and randomly selected member positions are important for this phase due that the updating process depends on them. Thus the probability value is utilized to determine the current member stat (is allowed to update or not). Therefore, if the probability of current member ( $Prob(P_j)$ ) is satisfied for updating the  $i^{th}$  member position then it is selecting the dimension randomly as shown in (4).

The probability of a current value is strongly proportional to the fitness value of that one. Whereas the best function value in the population is called *GlobalMin*. The update process will continue until the termination criteria are respected in Algorithm 2.

$$P_{new\ j} = P_{i,j} + rand(0,1)(PGL_j - P_{i,j}) + rand(-1,1)(P_{r,j} - P_{i,j}) \quad (4)$$

where  $PGL_j$  is the global leader position in the  $j^{th}$  dimension.

Algorithm 1: LOCAL LEADER PHASE

```

1  Begin
2  for  $gr = 1$  to  $tgr$  do
3  for  $i = gr[lo]$  to  $gr[hi]$  do
4  randomly select  $P_r$ 
5  for  $j = 1$  to  $D$  do
6  if  $rand(0,1) \geq pr_{it}$  // from (2)
7   $P_{newj} = Apply(2)$ 
8  else
9   $P_{newj} = P_{i,j}$ 
10 end if
11 end for
    
```

Algorithm 1: LOCAL LEADER PHASE

```

12 Calculate  $fun(P_{newj})$ 
13 if  $fun(P_{newj}) \leq fun(P_{i,j})$ 
14    $P_{i,j} = P_{newj}$ 
15 end
16 end for
17 end for
18 end

```

#### 2.1.4. LOCAL LEADER LEARNING PHASE

The best member of the group is selected in this phase as the leader of the group. If the leader position is not updating in current iterations then the value is incremented by 1 in *LocalLimitCount*.

#### 2.1.5. GLOBAL LEADER LEARNING PHASE

Thus, it is during this phase that we select the best position for a global leader by using a greedy selection in the total population. The *GlobalLimitCount* is incrementing by 1 if the position of global leader does not get a change until the *global limit* ( $globallimit=NP/4$ ) is defined.

#### 2.1.6. LOCAL LEADER DECISION PHASE

This phase is based on the result of the value  $pr$  in (3). It checks whether there is stagnation or not in the execution process. According to the *LocalLimitCount* is calculated; if this value has reached *LocalLimit* ( $LocalLimit=NP/6$ ) then all the members of the group will be randomly reset from the experiences of the local and global leader see (5).

$$P_{newj} = P_{i,j} + (PGL_j - P_{i,j})rand(0,1) + (PLL_{g,j} - P_{i,j})rand(0,1) \quad (5)$$



### 2.1.7. GLOBAL LEADER DECISION PHASE

During this phase, we check the *GlobalLimitCount* has reached its maximum limit, also called *global limit*. If it has reached means the global leader stuck on local minima thus the groups are created according to the size of the groups reached the maximum limit (*MG*) or not. In case the group's sizes reached to *MG* then the *GL* tie up all groups into 1 and randomly reset.

Algorithm 2: GLOBAL LEADER PHASE

```
1  Begin
2  for  $gr = 1$  to  $tgr$  do
3     $t = 1, f = 0$ 
4     $i = gr[lo]$ 
5     $GrS = gr^{th}$  group size
6     $prob_i = 0.9 \left( \frac{fun(P_i)}{GlobalMin} \right) + 0.1$ 
7    while  $(t < GrS)$  do
8      if  $rand(0,1) \geq prob_i$  then
9         $f = 0;$ 
10        $t = t + 1$ 
11       randomly select dimesion  $j$ 
12       randomly select  $P_r$ 
13       Apply (4) to find new position
14       Calculate  $fun(P_{new})$ 
15       if  $fun(P_{newj}) \leq fun(P_{i,j})$ 
16          $P_{i,j} = P_{newj}$ 
```

## Algorithm 2: GLOBAL LEADER PHASE

```

17  else
18   $f = f + t$ ;
19  end if
20   $i = i + 1$ ;
21  if  $i == gr[hi]$  then
22     $i = gr[lo]$ 
23  end if
24  if  $f \geq t$ 
25     $t = t + 1$ ;
26  end if
27  end if
28  end while
29  end for
30  end

```

## 2.2. GREY WOLF OPTIMIZER

The Grey wolf optimizer (GWO) inspired by Grey wolf which is from Canidae family. These species are predators means considered in the top of the food chain. The common thing is the grey wolf live in a group of 4 to 12. As the behaviors of this species are different and strictly followed. In the algorithmic methodology, the leader is called alpha is either male or female member which planned for a place, sleeping time, time to wake, hunting for food and so on. The decision of alpha is influenced by other wolves belongs to the group. The main identification to recognize alpha by holding his/her tails down. There is one more common from SMO algorithm is that the alpha is a dominant member in the group which is a decision maker.

Similarly as the local leader in SMO, there is a beta member which is either male or female, follow and also instruct to alpha about the group activities. The main work of beta is to deliver the order of alpha to the lower level and regulate the group.

In ranking wise the lowest is called omega wolves which always deliver the information to their dominants. This wolf is only allowed to eat, in some cases, it has been seen that the internal fights occur in the group due to the frustration of omega.

The wolf is not alpha, beta, omega they are called delta. This wolf dominates the omega and delivers the information to alpha and beta.

The main concept of grey wolves is for hunting for target. In that the tracking, chasing, and attacking towards the target behavior is modeled. For that pursuing, harassing and encircling are the special strategies.

For mathematical modeling the social hierarchy, tracking, encircling, and attacking are utilized.

### 2.2.1. SOCIAL HIERARCHY

For modeling the algorithm the best fitted wolf alpha ( $\alpha$ ), second best beta ( $\beta$ ), and the third best is delta ( $\delta$ ) considered. The rest of the wolves are considered as omega ( $\omega$ ).

### 2.2.2. ENCIRCLING TARGET AND ATTACKING

The mathematical equation of encircling prey is mentioned below where  $T$  ( $\vec{T}_1, \vec{T}_2, \vec{T}_3$ ) calculated from the three best wolves' position such as  $\vec{T}_\infty, \vec{T}_\beta, \vec{T}_\gamma$  as follows:

$$\vec{E}_\infty = |\vec{C}_1 \vec{T}_\infty - \vec{T}|, \quad \vec{E}_\beta = |\vec{C}_2 \vec{T}_\beta - \vec{T}|, \quad \vec{E}_\gamma = |\vec{C}_3 \vec{T}_\gamma - \vec{T}| \quad (6)$$

$$\vec{T}_1 = \vec{T}_\infty - \vec{B}_1 \vec{E}_\infty, \quad \vec{T}_2 = \vec{T}_\beta - \vec{B}_2 \vec{E}_\beta, \quad \vec{T}_3 = \vec{T}_\gamma - \vec{B}_3 \vec{E}_\gamma \quad (7)$$

$$\vec{T}(it+1) = \frac{\vec{T}_1 + \vec{T}_2 + \vec{T}_3}{3} \quad (8)$$

Where  $\vec{b}$  is linearly decreasing from 2 to 0, and  $\vec{r}_1, \vec{r}_2$  is random number in [0, 1]. The current iteration is denoted by  $it$ . Moreover  $\vec{E}$  is the distance of current position vector to the best.

### 2.3. DIFFERENTIAL EVOLUTION

The differential evolution is first proposed by Storn and Price (1997). There are three operators used in this algorithm such as Mutation, Crossover and Selection.

#### 2.3.1. MUTATION

A new solution is generated by adding the weight difference between two population members.

Here,  $P_{kj}$  ( $k=r1, r2, r3$ ) is randomly selected  $r^{th}$  member position in  $j^{th}$  dimensions.

$$PU_{new,j} = \begin{cases} P_{r3,j} + F_w (P_{r1,j} - P_{r2,j}); & rand(0,1) < 0.5 \\ P_{r3,j} + 0.5(F_w + 1)(P_{r1,j} + P_{r2,j} - 2P_{r3,j}); & Otherwise \end{cases} \quad (9)$$

#### 2.3.2. CROSSOVER

The crossover solution is generated after the mutation operation. There are many ways of crossover but in this paper it has omitted. Based on the mutant solution the process is further executed

#### 2.3.3. SELECTION

The better solution in the comparison of current solution and crossover solution is selected for the next iterative process.

$$P_{ij} = \begin{cases} PU_{new,j}; & \text{if } fun(PU_{new,j}) < fun(PU_{i,j}) \\ P_{i,j}; & \text{Otherwise} \end{cases} \quad (10)$$

## 2.4. MULTI-LAYER PERCEPTRON

Multi-layer perceptron (MLP) is a variant of neural network. In details it belongs to the category of feed forward neural network (FFN). The MLP is used to solve nonlinear problems. The main challenge is to find the proper combination of weight and biases to minimize the error. The output of MLP is as follows:

First, the (11) calculates the inputs weighted sums (Csáji, 2001) force is seldom considered in assessment of competency. The objective of this study was to explore the force applied during orotracheal intubation as a method of further discriminating between levels of competence. We sought evidence of construct validity in the form of discriminant, criterion, and concurrent validity. We hypothesized that the force generated during simulated intubation could serve to discriminate skill level among clinicians. Methods A convenience sample of 35 health-care professionals filled a self-reported questionnaire and were then divided into the following three groups: Group 1, experts (n=16).

$$ws_j = \sum_{i=1}^n (W_{ij} I_i) - \beta_j, \quad j = 1, 2, \dots, h \quad WS_j = \text{sigmoid}(ws_j) = \frac{1}{(1 + \exp(-ws_j))} \quad (12)$$

Where  $W_{ij}$  is the connection weight from the  $i^{th}$  node in the input layer to  $j^{th}$  node in the hidden layer, then n is the number of the input node, afterwards  $I_i$  represents the  $i^{th}$  input and  $\beta_j$  indicate the bias of the  $j^{th}$  hidden node.

$$ho_k = \sum_{j=1}^n (W_{jk} WS_j) - \beta'_k, \quad k = 1, 2, \dots, m \quad (13)$$

$$FO_k = \text{sigmoid}(ho_k) = \frac{1}{(1 + \exp(-ho_k))} \quad (14)$$

Where  $W_{jk}$  is the connection weight from the  $j^{th}$  node in the hidden layer to  $k^{th}$  node in the output layer and is represents the bias (threshold) of the  $k^{th}$  output node.

### 3. PROPOSED ALGORITHM

The SMO is modified in two ways in order to find the optimum result as discussed above. The DE or GWO algorithms are employed after local leader phase. The execution of the DE or GWO algorithm is conditional. In detail, the execution model is depicted in Algorithm 3. The random executions of DE and GWO algorithm after local leader phase improve the exploitation and exploration ability respectively. The proposed SMODE and SMOGWO algorithms fill the drawbacks of converging on local optima rather global optima.

Algorithm 3: GLOBAL LEADER PHASE

***Begin***

***Randomly initialize population using Eq. (1)***

Train to MLP and calculate the output

***Initialize Global Leader, local leader***

***Create group***

***while iter ≤ MaxIter***

***for gr = 1 : tgr***

***Local leader phase //See Algorithm 1***

***if rand(0,1) < rand(0,1)***

***Apply mutation from Eq (11) //For SMODE***

***or***

Apply GWO Phase from Eq. (10) //For SMOGWO

***end if***

Update the current position based on the best value calculated from MLP output from Eq. (14)

***end for***

***Global Leader Phase //See Algorithm 2***

***Global Leader Learning Phase***

***Local Leader Learning Phase***

***Local Leader Decision Phase from Eq. (5)***

***Global Leader Decision Phase***

***End while***

Algorithm 3: GLOBAL LEADER PHASE

The classification rate is calculated using MLP in Eq. (14) from the best value of the improved SMO.

**End**

## 4. EXPERIMENT AND RESULTS

There are 5 datasets used for classification in this section. These datasets are XOR, heart, balloon, iris, and breast cancer, obtained from the University of California at Irvine (UCI) Machine Learning Repository. The modified SMO algorithm is compared with BBO, PSO, ES, SVM (Hsu, Chang & Lin, 2008), and KNN on these datasets.

The parameters for the execution of BBO, PSO, ES, and SMO are the range of lower and upper boundary is defined in  $[-10, 10]$ . The population sizes, a total number of iterations are 200, and 300 respectively. The number of independent execution is defined 10 owing to the stochastic nature of algorithm where the SVM and KNN also have executed 10 times independently to record the average which is Average (MEAN), and standard deviation (STD) to determine the dispersion of results.

### 4.1. DATA

In this subsection, we detail the different data to handle in the experimentation part. These datasets were retrieved on UCI Machine Learning Repository, and Table 1 completes the other dataset information.

**Table 1.** Datasets parameters.

Classification datasets	Number of attributes	Number of instances	Number of class
XOR	3	8	2
Balloon	4	16	2
Iris	4	150	3
Breast Cancer	10	599	2
Heart	22	80	2

#### 4.1.1. XOR DATASET CLASSIFICATION PROBLEM

This dataset represents a well-known nonlinear problem, which is the XOR. This XOR has N-bit takes bits and works like this: The XOR result of the input vector must be returned, if the input vector contains an odd number of “1”, the output is “1”, and if the input vector contains an even number of “1’s”, the output is “0”. In this paper, the XOR 3-bits are used.

#### 4.1.2. BALLOON DATASET CLASSIFICATION PROBLEM

This dataset is one of the most popular datasets in the machine learning area. This dataset makes it possible to determine the state of swelling of a balloon (yes or no). These data are used in cognitive psychology experiments on 4 attributes that are: color, size, action and age.

#### 4.1.3. IRIS DATASET CLASSIFICATION PROBLEM

Iris is a famous dataset introduced in 1936 by Ronald Aylmer Fisher as an example of discriminant analysis and classification. This dataset contains 150 examples of criteria observed on 3 different species of iris (Setosa, Versicolor, Verginica). Each example is composed of four attributes (length and width of the sepals in cm, length and width of the petals in cm) and a class (the species).

#### 4.1.4. CANCER DATASET CLASSIFICATION PROBLEM

Dr William H. Wolbergby of the University of Wisconsin’s hospitals, Madison, has collected samples on breast cancer reporting his clinical cases to make a dataset. This dataset contains 10 attributes including Sample code number, Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli and Mitoses. And the output (class) is 2 for benign cancers and 4 for malignant cancers.



#### 4.1.5. HEART DATASET CLASSIFICATION PROBLEM

The dataset on cardiac single proton emission tomography (SPECT) images, each patient categorized into two categories: normal and abnormal. This dataset contains 267 sets of images SPECT (patients) has been processed to extract the features that summarize the original SPECT images. As a result, 44 continuous characteristic patterns were created for each patient. The model was then processed to obtain 22 binary characteristic models and it is the latter one that we will use in this paper, with 80 instances in the training set.

#### 4.2. RESULTS AND DISCUSS

The results show that for all datasets evolutionary algorithms are better than machine learning classical algorithms. In this paper, we use the average (MEAN) and standard deviation (STD) matrices to analyze the performance of the algorithm in 10 independent runs.

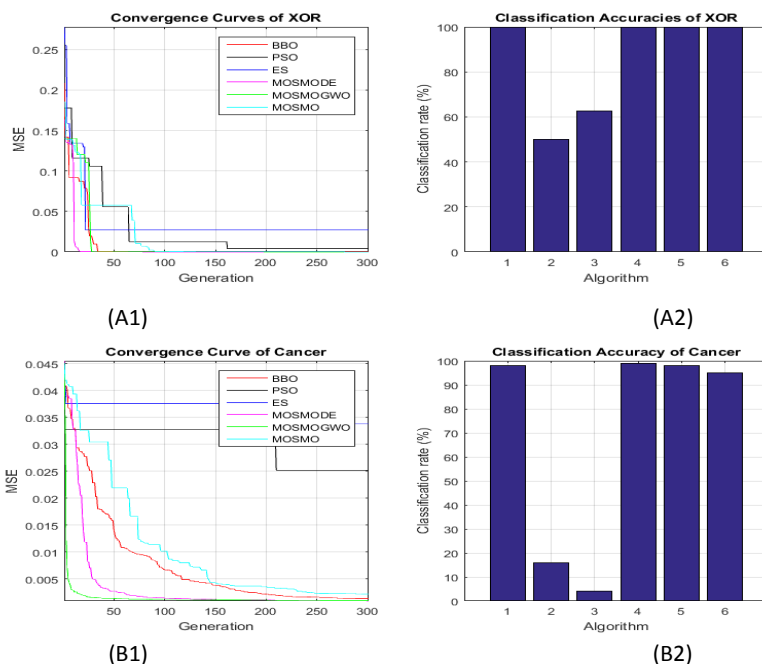
**Table 2.** Results of different algorithms on XOR, Baloon, Iris, Cancer, and Heart datasets.

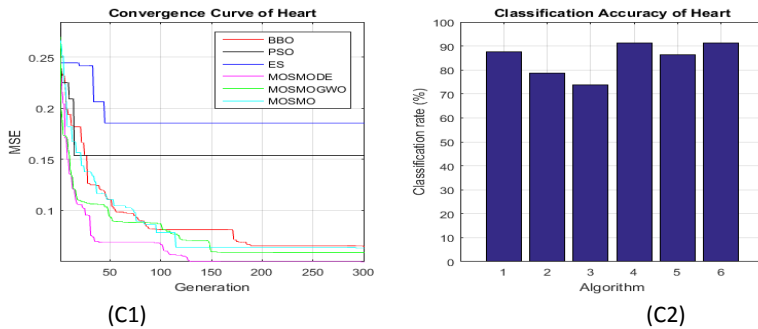
ALGORITHMS	METRICS	DATASETS				
		XOR	BALLOON	IRIS	CANCER	HEART
BBO	MEAN	1.11E-08	2.16E-35	<b>1.78E-02</b>	1.39E-03	6.41E-02
	STD	0.00E+00	2.82E-51	<b>3.47E-18</b>	<b>2.29E-19</b>	<b>1.39E-17</b>
ES	MEAN	6.36E-02	1.58E-03	3.00E-01	3.80E-02	1.83E-01
	STD	3.65E-02	2.94E-03	6.36E-02	1.59E-03	8.41E-03
SMO	MEAN	1.60E-05	2.19E-08	6.68E-01	1.92E-03	6.68E-01
	STD	1.74E-05	6.94E-08	1.48E-03	2.28E-03	1.48E-03
SMO-DE	MEAN	<b>0.00E+00</b>	<b>0.00E+00</b>	6.67E-01	<b>9.53E-04</b>	<b>5.53E-02</b>
	STD	<b>0.00E+00</b>	<b>0.00E+00</b>	1.38E-06	<b>8.75E-05</b>	1.65E-02
SMO-GWO	MEAN	7.96E-06	2.24E-24	6.84E-02	1.13E-03	6.84E-02
	STD	1.29E-05	7.07E-24	8.06E-03	9.02E-05	8.06E-03
PSO	MEAN	2.72E-02	2.57E-05	2.19E-01	2.85E-02	1.86E-01
	STD	1.97E-02	4.97E-05	2.20E-02	3.85E-03	1.03E-02
SVM	MEAN	2.50E-02	1.00E+00	9.65E-01	9.65E-01	7.01E-01
	STD	7.50E-02	0.00E+00	7.21E-03	3.30E-03	1.93E-02
KNN	MEAN	3.75E-01	9.80E-01	9.40E-01	9.67E-01	3.98E-01
	STD	0.00E+00	3.32E-02	8.38E-03	1.43E-03	2.95E-02
LR	MEAN	2.50E-02	1.00E+00	4.98E-03	9.55E-01	5.98E-01
	STD	7.50E-02	0.00E+00	6.21E-03	4.42E-03	3.47E-02

According to those metrics the proposed algorithm gives best results for all datasets. Thus we can see that for XOR and Balloon datasets SMO-DE gives the

best result. Afterwards for IRIS dataset BBO algorithm is the best. In addition, for CANCER and HEART datasets the average (MEAN) over 10 runs the SMO-DE algorithm is better.

In Figure 1, the convergence curve and classification accuracy of XOR, Cancer, and Heart datasets are depicted. The SVM, KNN, and LR are excluded from the graphical representation. In the figure the X-axis of classification accuracy graph has mentioned like 1 for BBO, 2- PSO, 3- ES, 4-MOSMODE, 5-MOSMOGWO, and 6- MOSMO. The convergence curves in (A1) and the classification accuracy in (A2) on XOR dataset except PSO and ES other algorithms are performing well. In that, the convergence speed in (A1) of MOSMODE is faster than others. However, (B1) is showing that the speed of MOSMOGWO is better than others but from (B2), the classification accuracy of MOSMODE is better. Similarly, C1, is showing better convergence speed of MOSMODE while C2 is showing the similar classification accuracy rate as MOSMO algorithm.





**Figure 1.** Convergence curve and Classification rate of different datasets.

## 5. CONCLUSION

From overall results, the performance of the proposed algorithm is better than other competitive algorithms. The results on XOR, and Baloon datasets the performance of SMO-DE is best while on Iris dataset it is very competitive. There are two other datasets Such as Breast cancer, and Heart the average of SMO-DE is better. Also, the proposed algorithm is more competitive than machine learning classical algorithms, for example, KNN, SVM and LR shown in Table 2.

## REFERENCES

- Bansal, J. C., Sharma, H., Jadon, S. S. & Clerc, M.** (2014). Spider Monkey Optimization algorithm for numerical optimization. *Memetic Computing*, 6(1), pp. 31–47. doi: <http://dx.doi.org/10.1007/s12293-013-0128-0>
- Bell, N. & Oommen, B. J.** (2017). A novel abstraction for swarm intelligence: particle field optimization. *Autonomous Agents and Multi-Agent Systems*, 31(2), pp. 362–385. doi: <http://dx.doi.org/10.1007/s10458-016-9350-8>
- Csáji, B.** (2001). Approximation with artificial neural networks. *MSc. Thesis*, 45.
- Deep, K. & Thakur, M.** (2007a). A new crossover operator for real coded genetic algorithms, *188*(1), pp. 895–911. doi: <http://dx.doi.org/10.1016/j.amc.2006.10.047>
- Deep, K. & Thakur, M.** (2007b). A new mutation operator for real coded genetic algorithms, *193*(1), pp. 211–230. doi: <http://dx.doi.org/10.1016/j.amc.2007.03.046>
- Erfani, T. & Erfani, R.** (2015). An evolutionary approach to solve a system of multiple interrelated agent problems. *Applied Soft Computing Journal*, 37, pp. 40–47. doi: <http://dx.doi.org/10.1016/j.asoc.2015.07.049>
- Ferris, M. C. & Mangasarian, O. L.** (1995). Breast Cancer Diagnosis via Linear Programming. In *IEEE Computational Science and Engineering*, 2(3), pp. 70–71. doi: <http://dx.doi.org/10.1109/MCSE.1995.414885>
- Guo, G., Wang, H., Bell, D., Bi, Y. & Greer, K.** (2003). KNN Model-Based Approach in Classification. In Meersman R., Tari Z., Schmidt D.C. (eds) *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. OTM 2003. Lecture Notes in Computer Science*, vol 2888. Springer, Berlin, Heidelberg. doi: [http://dx.doi.org/10.1007/978-3-540-39964-3\\_62](http://dx.doi.org/10.1007/978-3-540-39964-3_62)
- Gupta, K. & Deep, K.** (2016). Proceedings of Fifth International Conference on Soft Computing for Problem Solving, 437. doi: <http://dx.doi.org/10.1007/978-981-10-0451-3>

- Gupta, K., Deep, K. & Bansal, J. C.** (2017). Improving the Local Search Ability of Spider Monkey Optimization Algorithm Using Quadratic Approximation for Unconstrained Optimization. *Computational Intelligence*, 33(2), pp. 210–240. doi: <http://dx.doi.org/10.1111/coin.12081>
- Gutjahr, W. J.** (2007). Mathematical runtime analysis of ACO algorithms : survey on an emerging issue. *Swarm Intelligence*, 1(1), pp. 59–79. doi: <http://dx.doi.org/10.1007/s11721-007-0001-1>
- Hinchey, M. G., Sterritt, R. & Rouff, C.** (2007). Swarms and swarm intelligence. *Computer*, 40(4), pp.111-113. doi: <http://dx.doi.org/10.1109/MC.2007.144>
- Hsu, C., Chang, C. & Lin, C.** (2008). A Practical Guide to Support Vector Classification. *BJU International*. doi: <http://dx.doi.org/10.1177/02632760022050997>
- L. Fine, T.** (1999). Feedforward Neural Network Methodology.Pdf.
- Li, K., Deb, K., Zhang, Q. & Kwong, S.** (2015). An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition. *IEEE Transactions on Evolutionary Computation*, 19(5), pp. 694–716. doi: <http://dx.doi.org/10.1109/TEVC.2014.2373386>
- Mirjalili, S.** (2019). Biogeography-based optimisation. In *Studies in Computational Intelligence*, pp.57-72. doi: [http://dx.doi.org/10.1007/978-3-319-93025-1\\_5](http://dx.doi.org/10.1007/978-3-319-93025-1_5)
- Mirjalili, S., Mirjalili, S. M. & Lewis, A.** (2014a). Grey Wolf Optimizer. *Advances in Engineering Software*, 69, pp. 46–61. doi: <http://dx.doi.org/10.1016/j.advengsoft.2013.12.007>
- Mirjalili, S., Mirjalili, S. M. & Lewis, A.** (2014b). Let a biogeography-based optimizer train your Multi-Layer Perceptron. *Information Sciences*, 269, pp. 188–209. doi: <http://dx.doi.org/10.1016/j.ins.2014.01.038>
- Storn, R. & Price, K.** (1997). Differential Evolution - A simple and efficient heuristic for global Optimization over continuous spaces. *Journal of Global Optimization*, 11(4), pp. 341–359. doi: <http://dx.doi.org/10.1023/A:1008202821328>

- Sharma, A., Sharma, A., Panigrahi, B. K., Kiran, D. & Kumar, R.** (2016). Ageist Spider Monkey Optimization algorithm. *Swarm and Evolutionary Computation*, 28, pp. 58–77. doi: <http://dx.doi.org/10.1016/j.swevo.2016.01.002>
- Shi, Y. & Eberhart, R. C.** (1945). Empirical Study of Particle Swarm Optimization, pp. 1945–1950.
- Singh, P. R., Elaziz, M. A. & Xiong, S.** (2018). Modified Spider Monkey Optimization based on Nelder–Mead method for global optimization. *Expert Systems with Applications*, 110, pp. 264–289. doi: <http://dx.doi.org/10.1016/j.eswa.2018.05.040>
- Symington, M. M. F.** (1990). Fission-fusion social organization in Ateles and Pan. *International Journal of Primatology*, 11(1), pp. 47–61. doi: <http://dx.doi.org/10.1007/BF02193695>
- UCI Machine Learning Repository.** Retrieved from <https://archive.ics.uci.edu/ml/index.php>
- Zaharie, D.** (2009). Influence of crossover on the behavior of Differential Evolution Algorithms. *Applied Soft Computing Journal*, 9(3), pp. 1126–1138. doi: <http://dx.doi.org/10.1016/j.asoc.2009.02.012>

