International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

How Effective is Spotted Hyena Optimizer for Training Multilayer Perceptrons

Nibedan Panda, Santosh Kumar Majhi

Abstract*: This paper focuses on training multilayer perceptron (MLP) using a recently proposed meta-heuristic algorithm termed as Spotted Hyena Optimizer (SHO). To test the efficacy of the said algorithm fifteen standard datasets are used. At the same time the result of the proposed method is examined by some popular heuristic training algorithms such as Differential Evolution (DE), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Salp Swarm Algorithm (SSA) and Grey Wolf Optimization algorithm (GWO). Final result shows that SHO successfully avoids the local minima trap problem, simultaneously showing higher accuracy in classification as compared to other meta-heuristic methods. The statistical significance of the proposed SHO-MLP has been verified by deploying the Friedman & Holm's test. It has been observed that the SHO-MLP is giving promisingly better result than other compared method for training MLP.

***Index Terms*: Classification, MLP, SHO, DE, GA, PSO, GWO, SSA, ANN**

I. INTRODUCTION

With the rapid growth of technology, the invented elements count also increased massively. Moreover, the behaviour and patterns of the elements are so complex that it makes difficult to examine and remember the patterns of the elements. The simplification of the process need to classify the elements into different groups based on their similarities. Classification is a technique in machine learning which aims to accurately predict the class level. It follows a two-step approach: first phase is learning step or training and the second phase is classifying or testing phase. The process of classification belongs to supervised learning means total number of set of possible class labels will be known previously. The objective is to investigate the given data and to develop a model to classify unknown objects for each identified class by using the patterns found in the data. Broadly classification algorithm is categorized into two types: binary classification and multiclass classification. The algorithm which is responsible for implementing classification is known as a classifier. Generally speaking the effectiveness of the classifier fully depends on the characteristics of the data to be categorized.

Revised Manuscript Received on 30 July 2019.
* Correspondence Author
Nibedan Panda*, Computer Science & Engineering, VSSUT, Burla, Odisha, India. E-mail: nibedan.panda@gmail.com
Dr. Santosh Kumar Majhi, Computer Science & Engineering, VSSUT, Burla, Odisha, India. E-mail: smajhi_cse@vssut.ac.in

Application of classification varies across different dimensions in the arena of engineering and science such as: text categorization [1], biological classification [2], natural language processing [3],
document classification [4], internet search engine [5], pattern recognition [6], medical imaging [7], handwritten character recognition [8], micro-array classification [9], voice classification [10], gene expression classification [11]. Major classifiers available in the literature are Bayesian classifier [12], Support vector machine [13], K-nearest neighbor [14], Decision tree [15] and Artificial Neural Network [16]. Current researcher's gives emphasis on neural network because of its network architecture and algorithms used to solve complex real life problems using evolutionary approaches. The nature inspired evolutionary algorithms are deployed along with neural networks to solve many classification and regression problems. In addition, new meta-heuristic algorithms are emerged day by day due to the increase in complexity in day to day life problems. The newly developed heuristic based algorithms also shows optimum result to the real world problems with minimal complexity used as the training algorithms for artificial neural networks (ANNs). ANNs is just like nervous system of a human being which receives input, processes it and transforms it to output. The important component of an ANN is the artificial neurons which receives the input and represent them as output. Now a days ANNs plays a vital role because its ability to learn and model complex and non-linear relationship. ANN is also treated as universal function approximator which means it can compute and train any function. ANNs are mostly used for solving classification, forecasting, image processing and character recognition problem. In the year 1943 McCulloch and Pitts gave the first idea of neural network as a computational model [17]. At present various type of ANNs are available in the literature such as: Feed forward network [18], Kohonen self-organizing network [19], Radial basis function network [20], Recurrent neural network [21] and spiking neural network [22]. Though there are lot of differences among neural networks, there learning pattern is same. Learning refers to the ability of a neural network that how it behaves from its own experience. This learning is either supervised learning or unsupervised learning. Both supervised and unsupervised learning are the machine learning task. The former one according to the name indicates the learning process with the presence of a supervisor i.e. outcome is known and data require to train the algorithm is previously labelled. Whereas, the later there is no possibility of labelled data or no direct supervisor is required.

The main aim of unsupervised learning is how to find similarities among data without any prior knowledge. The supervised learning process learns by two methods: deterministic learning and heuristic approach for learning. In deterministic learning approach no randomness is involved rather uses some statistical techniques such as linear regression. The set of initial conditions and parameters fully determines the future solution. It assumes that for given fixed input the output is certain. In literature, widely used deterministic methods available are back-propagation method and gradient-based methods. The advantages of deterministic methods are its simplicity, versatility and accuracy. The disadvantage of the deterministic method is that it will suffer from local optimal problem. Local optima means from a set of sub-optimal solutions it will choose one local optima assuming it as global optimal solution. Also gradient-based method is very slow. Comparing to the deterministic method, the heuristic or stochastic method depends on randomness. The initial parameters require for training assigned randomly and evolve multiple times to get improved result. Advantage of heuristic method over deterministic method is the ability to avoid local minima trap for which it is gaining high attention [41]. However, the deterministic one is slower in nature.

Meta-heuristic is the strategy which guides the search process and non-deterministic in nature. The aim is to effectively explore the search space to find the optimal solution. Due to the property that it is not problem dependent so gaining lot of attention in current scenario [40]. Broadly it is divided into three categories. The first one is evolution based, second one is physics based and the third one is swarm based. Evolution based methods follows the law of nature for evaluation. The process initiates by a set of randomly generated parameters called as population and the same is evolved repeatedly for further generations. One of the important evolution based algorithm is Genetic Algorithm (GA) [23], which follows the concept of Darwinian Theory for evaluation. Some other popular methods are Differential Evolution (DE) [24], Population-Based Incremental Learning (PBIL) [25] and Biogeography-Based Optimizer (BBO) [26]. Physics based methods start the search process by applying some physical rules of the universe. In literature popular methods are Simulated Annealing (SA) [27], Black Hole Optimization Algorithm [28] and Galaxy-based Optimization Algorithm [29]. The third type of meta-heuristic algorithm uses swarm or population based methods which is appropriate for global searches. Merits of this type of population based method are that it shows local exploitation and global exploration. Example of this type of algorithm includes Particle Swarm Optimization (PSO) [30], Ant Colony Optimization (ACO) [31] and Artificial Bee Colony (ABC) [32]. The above discussed meta-heuristic algorithms shows high performance in terms of finding global optimal solution which motivates us to test the effectiveness of the recently proposed algorithm termed as Spotted Hyena Optimizer (SHO) [33] in training multilayer perceptron (MLP) to solve classification problem. The flow of the paper is presented as follows: The materials and methods sections presents an overview of Spotted Hyena Optimizer algorithm, the proposed SHO-based MLP and details about datasets and structure of the MLP used. Section 3 represents results and discussions regarding SHO-based MLP trainer along with the statistical performance evaluation. Section 4 presents the conclusion and future scope of the proposed work.

II. MATERIAL AND METHODS

This section describes the background of MLP, SHO and proposed SHO-MLP.

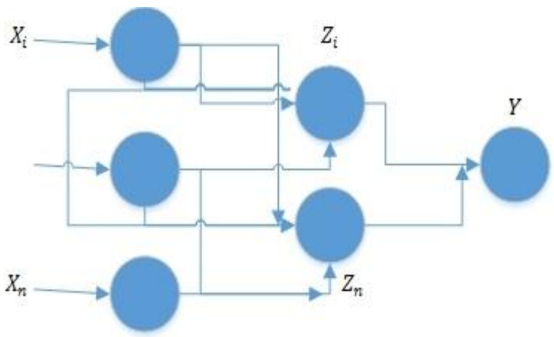
A. Background and Related work

In the literature quite good number of popular multi solution based evolutionary trainers are available: such as GA, PSO, DE, ACO, Grey Wolf Optimization (GWO) and Whale Optimization (WO). The acceptance behind such type of trainers is due to its higher efficacy in terms of attaining global optimum value. In the current year 2019 Bairathi et al. castoff SSA algorithm with MLP network for obtaining optimal set of weights and biases. The efficiency was scrutinized by using standard datasets and by comparing with some recent metaheuristic algorithms [45]. In 2018 Haidari et al. proposed MLP network with Grasshopper Optimization algorithm. The proposed model was applied on five standard datasets to verify the classification accuracy. The outcome was also scrutinized with eight state-of-art algorithms qualitatively as well as quantitatively, for proving the balance between exploration and exploitation along with local optima avoidance [42]. In 2015 Mirjalili castoff GWO as a recent multilayer perceptron training algorithm to test the efficacy of GWO as a suitable trainer. He has used five standard classification datasets from UCI repository, three function approximation datasets and also compared the outcome with other recent MLP based trainers such as PSO, GA, ACO and PBIL to verify the efficacy [34]. In 2016 Ibrahim et al. proposed a new training algorithm based on Whale Optimization method. They have tested the efficacy of the trainer by the help of 20 standard datasets. The outcome was compared by backpropagation algorithm and other six popular evolutionary techniques and the efficiency was confirmed in terms of local optima stagnation and convergence rate [39]. In 2012 Mirjalili et al. proposed feed forward neural network training using hybridization of PSO and Gravitational Search Algorithm (GSA). The resulting accuracy for PSO, GSA, and hybrid PSOGSA was compared in terms of local optima avoidance and convergence rate [43]. In 2007 Socha et al. castoff ACO algorithm for training neural network. They have used the neural network training for pattern classification. Furthermore to validate the efficiency of the ACO based trainer, they have compared with gradient based technique such as backpropagation and evolutionary algorithm as GA [44].

B. Multilayer Perceptron (MLP)

MLP belongs to the class of Feed forward neural network which follows supervised learning technique. It contains at least three layers such as input layer, output layer and hidden layer. The hidden layer is the intermediate layer which is present in between input layer and output layer.

Its job is to process the input which is fed from the previous layer by using activation function. To decide optimum number of neurons and hidden layers completely depends upon the nature of the problem. The presence of activation function converts the input signal of a neuron to output signal, which output also acts as an input to the next layer. The typical architecture of a MLP is given in “Fig. 1”.



“Fig. 1” Basic Structure of MLP

The output can be calculated by using the equation (1).

$$a_j = \sum_{i=1}^{d} W_{ji} * X_i + e_j, \quad j=1, 2, 3\ldots m \qquad (1)$$

Where m signifies the linear combinations of parameters belongs to first layer and d signifies d-number of dimensions as input and

a_j , signifies the outcome from j[th] hidden neuron

W_{ji} , signifies weights

X_i, signifies i_{th} number of inputs

e_j , signifies biases or thresholds.

The individual activations mutate by one activation function, generally a sigmoid and given in equation (2), (3), (4).

$$Sd_j = \frac{1}{1+\exp(-a_j)} \qquad (2)$$

The final output can be calculated as,

$$o_n = \sum_{j=1}^{n} (W_{jn} * a_j) + e_n, \quad n=1, 2, 3\ldots m \qquad (3)$$

Where W_{jn} signifies weights from hidden node to nth output node.

$$O_n = \frac{1}{1+\exp(-o_n)} \qquad (4)$$

C. Overview of Spotted Hyena Optimizer (SHO)

Spotted Hyena Optimizer (SHO) is a recently developed population based meta-heuristic algorithm [33]. Main focus of the proposed algorithm is the social relationship among the spotted hyenas and there hunting behavior. In the proposed algorithm the whole population is divided into two groups, the first one is termed as the current best and the rest. The others except the current best update their positions according to the current best search agent in the search space.

The hunting behavior is represented mathematically in equation (5) and (6).

$$\vec{D_h} = |\vec{B} * \vec{P_p} * (X) - \vec{P}(X)| \qquad (5)$$

$$\vec{P}(X+1) = \vec{P_p}(X) - \vec{E} * \vec{D_h} \qquad (6)$$

Where X signifies the current iteration

$\vec{P_p}$, signifies position vector of prey

$\vec{P}$, signifies the position vector of spotted hyena

Vectors $\vec{B}$ and $\vec{E}$ are denoted as:

$\vec{B} = 2 * \vec{rd_1}$

$\vec{E} = 2 * \vec{h} * \vec{rd_2} - \vec{h}$

$\vec{h} = 5 - (\text{Iteration} * (\frac{5}{maxiteration}))$ and value of $\vec{h}$ is decreased from 5 to 0 linearly

$\vec{rd_1}$ And $\vec{rd_2}$ signifies random vectors in the range 0 to 1.

During hunting the best search agent has the knowledge of the prey. The other search agents forms a cluster according to the best search agent and saved the best solution got to update their own positions. The equations (7), (8) and (9) are used to update the position of hyenas.

$$\vec{D_h} = |\vec{B} \cdot \vec{P_h} - \vec{P_k}| \qquad (7)$$

$$\vec{P_k} = \vec{P_h} - \vec{E} * \vec{D_h} \qquad (8)$$

$$\vec{C_h} = \vec{P_k} + \vec{P_{k+1}} + \ldots + \vec{P_{k+N}} \qquad (9)$$

Where $\vec{P_h}$ signifies the position of best search agent and $\vec{P_k}$ signifies the position of other search agents in the search space. $\vec{C_h}$, signifies the cluster of N number of optimum solutions.

$N = Count_{nos}(\vec{P_h}, \vec{P_{h+1}}, \vec{P_{h+2}} + \ldots + (\vec{P_h} + \vec{M}))$

Where N signifies number of spotted hyenas, M is a random vector, ranges from 0 to 1 and nos makes a count of all the candidate solutions.

Finally the best solution can be calculated by using equation (10).

$$\vec{P}(X+1) = \frac{\vec{C_h}}{N} \qquad (10)$$

Where $\vec{P}(X+1)$ signifies the best solution which is saved in each evolve and other search agents update themselves according to the saved best solution obtained.

D. Proposed SHO-Based MLP trainer

The prime concern with gradient based and other traditional training algorithms are entrapment in local optima and slow convergence rate, which leads to the growth of metaheuristic based optimization algorithms for training neural network. This is the motivation for our work to explore the efficiency of recently proposed SHO algorithm for training MLP. In fact the prime concern to choose the SHO as the trainer is:

- No recent metaheuristic based trainer guarantees for attaining global optimum.
- Higher exploration and exploitation capability of SHO as comparing to other recent trainers.

The current segment explains the proposed SHO-based MLP training. The SHO algorithm is used to train the MLP. During training the MLP using meta-heuristic algorithm, the important aspect is the selection of parameters. According to the nature of heuristic algorithm the required parameters for MLP i.e. weights and biases are initially selected randomly. So the trainer should supply a set of initial values for getting optimum accuracy in terms of classification. In the proposed work, MLP network is designed by using a single hidden layer and applied SHO to train it. SHO uses its parameters in terms of vectors. The parameters are weights and biases represented using the equation (11) and (12).

$$\vec{W} = W_{1,1}, W_{1,1}, W_{1,1}, \ldots, W_{i,j} \qquad (11)$$

$$\vec{B} = b_1, b_2, b_3, \ldots, b_n \quad (12)$$

Where $W_{i,j}$ denotes set of weights and b_n denotes the set of biases or thresholds.

After supplying the initial parameters the effectiveness of the MLP can be calculated by evolving the set of randomly generated parameter lists multiple times. The effectiveness of the MLP can be tested by calculating the root mean square error (RMSE). The RMSE can be calculated by evaluating the difference between the target value and the real value obtained from the MLP at the end of specified number of iterations. The RMSE is represented in equation (13).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i^k - T_i^k)^2}{n}} \quad (13)$$

The other factors such as average root mean square error and standard deviation can be calculated from the RMSE value which is found out by repeated iteration of the parameters through MLP. There is no guarantee to find the best optimum value for any set of random parameters using any meta-heuristic algorithm due to its heuristic nature. Therefore, after successive iterations the RMSE value decreases continuously. However, after a certain number of iterations the value remains constant. Finally the solution we received that must attain a value which is better than the initial random inputs. Flow diagram for SHO-MLP network is represented in “Fig. 2”.

E. Benchmark Datasets

The efficacy of the SHO-MLP will be evaluated by fifteen standard datasets in which fourteen are chosen from UCI Machine Learning Repository [35] such as: Balloon, Cancer, Diabetes, Ecoli, Glass, Heart, Ionosphere, Iris, Liver, Seed, Soybean, Vehicle, Wine and Yeast. Including this fourteen another dataset i.e. XOR dataset also used in evaluation process. Among these fifteen datasets Balloon, Cancer, Diabetes, Heart, Liver, Ionosphere and XOR dataset belongs to 2-class problem. Other eight datasets such as: Glass, Iris, Seed, Soybean, Vehicle, Wine and Yeast comes under 3-class or multiclass problem. Balloon has 4-attributes, 20 training as well as 20 testing samples. This is one of the simplest dataset. All the meta-heuristic algorithm we have tested on this dataset shows optimum performance due to its simplicity. Cancer dataset has 9-attributes, 683 training samples and 120 testing samples. Diabetes dataset having 8 attributes, 768 training samples and 140 testing samples. Heart contains 13 attributes, 270 training samples and 60 testing samples. Iris is the mostly used dataset in the literature. It contains 4 attributes and 150 training as well as testing samples. Liver comprises of 6-attributes, 345 training and 70 testing samples. The XOR dataset has 3-attributes, 8-training as well as testing samples and one target. The Ionosphere contains 10-attributes with 351 training instances and 70 testing samples. Seed dataset contains 7-attributes having 210 training as well as testing instances. Glass dataset comprises of 9-attributes, 214 training instances and 42 testing samples. The dataset Ecoli having 7-attributes, 336 number of training instances and 60 testing samples. Soybean dataset has 35-attributes, 47 number of training instances and 12 as testing samples. Vehicle dataset contains 18-attributes, 846 number of training instances and 160 testing samples. Wine dataset having 13-attributes, 178 training instances and 51 testing samples. The Yeast dataset comprises of 8-attributes, 1484 number of training instances and 185 number of testing samples. Brief idea about all the datasets discussed above is represented in Table 2.

Table 1. Structure of MLP for different datasets

Classification of datasets	Number of Attributes	MLP structure
Balloon	4	4-9-1
Cancer	9	9-9-1
Diabetes	8	8-9-1
Ecoli	7	7-9-1
Glass	9	9-9-1
Heart	13	13-9-1
Ionosphere	10	10-9-1
Iris	4	4-9-1
Liver	6	6-9-1
Seed	7	7-9-1
Soybean	35	35-9-1
Vehicle	18	18-9-1
Wine	13	13-9-1
XOR	3	3-9-1
Yeast	8	8-9-1

Table 2. Description of used standard datasets

Classification datasets	Number of attributes	Number of training samples	Number of test samples	Number of class
Balloon	4	20	16	2
Cancer	9	683	120	2
Diabetes	8	768	150	2
Ecoli	7	336	60	8
Glass	9	214	42	6
Heart	13	270	60	2
Ionosphere	10	351	70	2
Iris	4	150	150	3
Liver	6	345	70	2
Seed	7	210	210	3
Soybean	35	47	12	4
Vehicle	18	846	160	4
Wine	13	178	51	3
XOR	3	8	8	2
Yeast	8	1484	185	10

III. RESULT AND DISCUSSIONS

The proposed SHO-based MLP is evaluated by utilizing fifteen standard datasets, which are chosen from UC IRVINE Machine Learning Repository [35]. From the statistical result, the proposed SHO-MLP based trainer shows improved RMSE value as well as standard deviation and higher classification accuracy. It pretends strong evidence for successfully avoiding premature convergence of local minima trap and shows best optimum values for the initial parameters such as weights and biases of the MLP.

The said algorithm also compared with some popular meta-heuristic based trainer to test the effectiveness such as: DE-MLP, GA-MLP, PSO-MLP and GWO-MLP [34]. The statistical results are presented in terms of RMSE, average, standard deviation, sensitivity, specificity and prevalence. The high classification accuracy of the SHO-MLP indicates the smooth balancing of process of exploration and exploitation. All the meta-heuristic algorithms shows highest efficacy in terms of classification accuracy i.e. 100 percent for Balloon dataset due to its simplicity. In case of Soybean and Ecoli dataset, the SHO-MLP as well as GWO-MLP gives same accuracy i.e. 100 percent for Soybean and 93.3 percent for Ecoli as well. SHO-MLP gives very challenging result in case of Cancer and Diabetes dataset as comparing to other MLP trainers. For Diabetes dataset SHO-MLP shows lowest accuracy i.e. 72.2 percent and DE-MLP shows 74.3 percent as highest accuracy and for Cancer SHO-MLP gives 96 percent accuracy whereas all others show 98.4 percent accuracy. For the dataset Iris, Glass, Vehicle, Seed, Heart and Ionosphere SHO-MLP shows superior result in terms of accuracy as compare to others and GWO-MLP shows challenging result to the proposed work. For Liver dataset SHO-MLP gives promisingly better result than others. For the multi class problem i.e. Yeast dataset where the number of class level is 10, SHO-MLP outperforms over other algorithms and exhibits optimum result as 99.5 percent accuracy. In case of Yeast and Wine dataset the proposed work gives better result but GA-MLP and PSO-MLP gives tight comparison. Finally for XOR dataset all the meta-heuristic based trainers shows same accuracy i.e. 75 percent. In 3-class problem i.e. by using Iris dataset the SHO-MLP gives superior result as compare to all other trainers. In Liver and Heart dataset SHO-MLP also gives highest classification accuracy. The performance measure of the proposed SHO-MLP with respect to RMSE, Specificity, Sensitivity and prevalence is presented in Table 3. It has been observed from Table 3 that in most data sets the SHO-MLP obtained minimum RMSE with low standard deviation. In addition, the specificity value of the SHO-MLP is higher than other methods. A tabular representation of accuracy shows by different trainers is presented in Table 4. From Table 4, it is clear that the proposed SHO-MLP gives highest accuracy value in all data sets except Cancer and Diabetes. "Fig. 4" represents the accuracy obtained by different MLP-trainers such as SHO-MLP, GWO-MLP, SSA-MLP, GA-MLP, PSO-MLP and DE-MLP based on fifteen standard datasets. "Fig. 3" represents the convergence curve for fifteen standard datasets for five algorithms such as SHO, GWO, GA, PSO and DE. The convergence curve is based upon the RMSE values. It is obtained by the individual algorithm for ten different runs. The curve shows that SHO gives faster convergence in case of Balloon, Ecoli, Ionosphere, Liver, Seed, Wine, XOR and Yeast datasets and for Cancer, Diabetes, Glass, Heart, Iris, Soybean and Vehicle datasets it gives a very challenging convergence. These results show that the SHO trainer performs better by preventing premature convergence and finds better optimal solution. Backpropagation algorithm has proven acceptance in training MLP for last three decades due to its various advantages (i) comparatively simple for implementation (ii) standard technique and mathematical principle used in the basic algorithm may be applied to all type of networks (iii) no exceptional function features to be learnt. Regardless of the above merits still backpropagation suffers from lot of difficulties, such as: (i) solution stagnates in local optima acquiring near sub-optimal solution (ii) outcome is not fixed, may alter over time with similar constraints also (iii) outcome may fuzzy in nature or nonnumeric (iv) difficult to relate input/output data to the result (v) may exhibit overwhelming computation cost, slow and inefficient to resolve real life complex problems. But MLP based trainer is well accepted when it outperforms to solve day to day life complex problems by generating approximate solutions. This is the main reason for lagging behind of back propagation based trainer as it will faster converge to a local optima without attaining the global optima, which leads to questioning on its capability to its real world application area. As a result metaheuristic algorithms evolved as one best choice, which attains the global optima in most of the cases. In the current work we have compared the proposed SHO-MLP with two recent continuous optimizer algorithm such as: Grew Wolf Optimizer (GWO) and Salp Swarm Optimizer (SSA) [38]. The efficacy of the said work has compared in terms of performance metrics such as min RMSE, standard deviation, average, accuracy, sensitivity, specificity and prevalence as well. In majority cases the proposed SHO-MLP trainer outperforms over all other metaheuristic based training algorithm confirming its acceptance and proving its perfect balance in between exploration and exploitation factor as well as controlling parameters. For a fair comparison among all metaheuristic algorithms, we have considered the search agents as 20 and maximum iteration as 100. To obtain the average and standard deviation error values, 10 number of unique RMSE values are considered. The lower RMSE, standard deviation and best accuracy obtained by SHO endorses its solid indication for better exploration capability of the proposed method, along with capable to attain global optimum with avoiding premature convergence and generates best optimal value for weights and biases.

A. Statistical Performance Evaluation

Statistical analysis is carried out to test the performance of classifying algorithms and check whether there exist substantial differences among them or not. Therefore, the Friedman test is conducted. This is a nonparametric test which clarifies the significance of results obtained from SHO-based trainer compared to other methods. This type of nonparametric test is useful when there is a need to compare different results obtained from multiple number of comparisons. The null hypothesis means the hypothesis which is usually proven to be true or false. It shows that there is no variation exists among the observed populations. Symbolically it is represented by: H_0. H_0: There is no substantial difference among five classifying algorithms. $\propto$, is the level of significance of rejecting the hypothesis. Here, the value of $\propto$ is taken as 0.05. Each individual algorithm is assigned a rank based on their accuracy value ranging from 1 to N. So, the ranking among all the algorithms will be carried out by the parameter accuracy. The algorithm which shows highest percentage in terms of accuracy assigned a rank 1 and last rank i.e. 6 if the percentage of accuracy is lowest. When more than one algorithm exhibits same accuracy then the rank is assigned by taking the average of the sequential ranks assigned to them.

In Table 5, each column contains some values within Ranking of all the algorithms briefly represented in Table 5. The average rank A_r of the r_{th} algorithm can be represented by equation (14).

$$A_r = \frac{\text{The total sum of ranks obtained by rth algorithm}}{\text{Total number of datasets}} \quad (14)$$

The Friedman statistics can be calculated by equation (15).

$$F_F = \frac{(N-1)X_F^2}{N(K-1)-(X_F^2)} \quad (15)$$

Where $X_F^2 = \frac{12N}{K(K+1)}\left[\sum_r A_r^2 - \frac{k*k+1^2}{4}\right]$

Here N signifies number of datasets and k signifies total number of algorithms used.

The Friedman statistics F_F is distributed according to the F-distribution with (k-1) and (k-1) (N-1) degree of freedom. Here we have taken five number of algorithms and fifteen number of datasets. So the degree of freedom is between 5 and 70. For F (5 70) while $\propto$=0.05 the critical value is 4.3985 [36]. The null hypothesis which is assumed is accepted if the F_F value we got is less than the critical value otherwise the hypothesis is rejected. In our work for five algorithms and fifteen number of datasets the X_F^2 value we got is 22.81 and the F_F value is 6.11. From the above values, it is clear that the critical value is smaller than the F_F value. Therefore, the null hypothesis is rejected and it contains that there exists some parenthesis which represents rank. difference among the considered algorithms. Multiplicity is a problem occurs when several types of hypothesis comes into consideration. So, Holm's procedure plays a vital role to test whether the control algorithm performs better than other algorithms or not. The considered null hypothesis H_0 tells the said two algorithms compared performing equally. To carry out the test, we have to calculate Z-value which can be computed by equation (16).

$$z = \frac{R_i - R_j}{S_E} \quad (16)$$

Where SE=$\sqrt{\frac{k(K+1)}{6*N}}$

According to the computed Z-value, probability P is found out by using Normal Distribution Table [37]. Then the P-value is compared with $\frac{\propto}{(k-i)}$. Here we have consider SHO-MLP is the control algorithm. So from table 6 we conclude that the hypothesis is rejected for all four cases because the P-value is smaller than $\frac{\propto}{(k-i)}$ value. Finally we conclude that SHO-MLP shows significantly better performance than other trainers i.e. GWO-MLP, GA-MLP, PSO-MLP and DE-MLP.

Table 3. Performance comparison of SHO w.r.t DE, GA, PSO, GWO and SSA

Dataset\Algorithm		DE	GA	GWO	SSA	PSO	SHO
XOR	MIN RMSE	0.4961	0.4855	0.4620	0.4612	0.4633	0.4556
	AVG	0.4988	0.4983	0.4771	0.4871	0.4658	0.4625
	STD	0.0019	0.0064	0.0065	0.0056	0.0016	0.0052
	SPECIFICITY	0.7500	0.7500	0.7500	0.7500	0.7500	0.7500
	SENSITIVITY	0.7500	0.7500	0.7500	0.7500	0.7500	0.7500
	PREVALENCE	37.50	37.50	37.50	37.50	37.50	37.50
Balloon	MIN RMSE	0.4233	0.4929	0.3829	0.4555	0.4264	0.3698
	AVG	0.2144	0.4960	0.4272	0.4985	0.4574	0.3712
	STD	0.0158	0.0021	0.0345	0.0151	0.0148	0.0016
	SPECIFICITY	1	1	1	1	1	1
	SENSITIVITY	1	1	1	1	1	1
	PREVALENCE	50	50	50	50	50	50
Cancer	MIN RMSE	0.2551	0.2846	0.3384	0.3733	0.3419	0.3146
	AVG	0.2852	0.2886	0.4087	0.4030	0.3944	0.3355
	STD	0.0141	0.0034	0.0512	0.0160	0.0250	0.0087
	SPECIFICITY	0.9662	0.9662	0.9662	0.9662	0.9662	0.9333
	SENSITIVITY	1	1	1	1	1	0.9833
	PREVALENCE	50	50	50	50	50	49.16
Diabetes	MIN RMSE	0.4418	0.4528	0.4601	0.4632	0.4564	0.4601
	AVG	0.4512	0.4581	0.4729	0.4678	0.4700	0.4640
	STD	0.0026	0.0043	0.0077	0.0030	0.0063	0.0036
	SPECIFICITY	0.6712	0.6283	0.6422	0.6856	0.5713	0.6428
	SENSITIVITY	0.8141	0.8422	0.8283	0.8286	0.8714	0.8000
	PREVALENCE	40.71	42.14	41.42	41.42	43.57	40
Heart	MIN RMSE	0.4735	0.4556	0.3658	0.4674	0.4890	0.4191
	AVG	0.4889	0.4629	0.3812	0.4799	0.4956	0.4242
	STD	0.0072	0.0038	0.0099	0.0085	0.0028	0.0041
	SPECIFICITY	0.7400	0.7777	0.7400	0.8888	0.6666	0.8148
	SENSITIVITY	0.8141	0.8141	0.8511	0.7777	0.8888	0.8148

	PREVALENCE	40.74	40.74	42.59	38.88	44.44	64.70
Iris	MIN RMSE	0.1921	0.2056	0.1764	0.2206	0.2053	0.2035
	AVG	0.2144	0.2120	0.1983	0.2807	0.2251	0.2141
	STD	0.0158	0.0050	0.0116	0.0372	0.0165	0.0058
Liver	MIN RMSE	0.4930	0.4880	0.4918	0.4922	0.4928	0.4657
	AVG	0.2144	0.2120	0.1983	0.2807	0.2251	0.4682
	STD	0.0158	0.0050	0.0116	0.0372	0.0165	0.0023
	SPECIFICITY	0.5000	0.5295	0.4413	0.5882	0.4413	0.6470
	SENSITIVITY	0.6767	0.6175	0.8232	0.6763	0.8232	0.8235
	PREVALENCE	33.82	30.88	41.17	33.82	41.17	41.17
Ionosphere	MIN RMSE	0.4798	0.4852	0.4328	0.4411	0.4797	0.4327
	AVG	0.4813	0.4863	0.4352	0.4488	0.4835	0.4394
	STD	0.0015	0.0014	0.0024	0.0026	0.0030	0.0037
	SPECIFICITY	0.314	0.280	0.800	0.786	0.628	0.771
	SENSITIVITY	0.628	0.657	0.571	0.681	0.485	0.800
	PREVALENCE	31.4	32.8	28.5	27.9	24.285	40
Seed	MIN RMSE	0.4090	0.3839	0.3370	0.3905	0.4082	0.2696
	AVG	0.4149	0.3873	0.3395	0.4043	0.4102	0.2779
	STD	0.0049	0.0024	0.0028	0.0042	0.0021	0.0070
Wine	MIN RMSE	0.3775	0.3872	0.3898	0.3911	0.3867	0.2327
	AVG	0.3825	0.3887	0.4047	0.4088	0.3891	0.2444
	STD	0.0053	0.0011	0.0079	0.0076	0.0020	0.0058
Ecoli	MIN RMSE	0.3556	0.3380	0.2973	0.3002	0.3013	0.2868
	AVG	0.3617	0.3483	0.3034	0.3220	0.3105	0.2929
	STD	0.0055	0.0066	0.0041	0.0089	0.0097	0.0055
Soybean	MIN RMSE	0.3871	0.3858	0.0869	0.2858	0.3875	0.1600
	AVG	0.3887	0.3894	0.1120	0.3015	0.3897	0.1697
	STD	0.0014	0.0035	0.0155	0.1110	0.0024	0.0059
Vehicle	MIN RMSE	0.3618	0.3820	0.3054	0.3014	0.3680	0.3352
	AVG	0.3665	0.3854	0.3158	0.3116	0.3720	0.3423
	STD	0.0026	0.0021	0.0085	0.0079	0.0044	0.0048
Glass	MIN RMSE	0.2721	0.3042	0.1870	0.2115	0.2496	0.2255
	AVG	0.2859	0.3136	0.1960	0.2245	0.2589	0.2296
	STD	0.0089	0.0056	0.0081	0.0145	0.0096	0.0036
Yeast	MIN RMSE	0.1983	0.1935	0.1870	0.2010	0.2038	0.1905
	AVG	0.2042	0.1982	0.1922	0.2098	0.2106	0.1963
	STD	0.0035	0.0022	0.0036	0.0088	0.0053	0.0038

Table 4. Accuracy and Error Rate comparison

Dataset\Algorithm		**DE**	**GA**	**GWO**	**SSA**	**PSO**	**SHO**
XOR	Error Rate (%)	25	25	25	25	25	25
	Accuracy (%)	75	75	75	75	75	75
Balloon	Error Rate (%)	0	0	0	0	0	0
	Accuracy (%)	100	100	100	100	100	100
Cancer	Error Rate (%)	1.6	1.6	1.6	1.6	1.6	4
	Accuracy (%)	98.4	98.4	98.4	98.4	98.4	96
Diabetes	Error Rate (%)	25.7	26.4	26.4	24.2	27.8	27.8
	Accuracy (%)	74.3	73.6	73.6	75.8	72.2	72.2
Heart	Error Rate (%)	22.2	20.3	20.3	16.6	22.2	18.5
	Accuracy (%)	77.8	79.7	79.7	83.4	77.8	81.5
Iris	Error Rate (%)	23.3	13.3	3.3	3.3	10	2.2
	Accuracy (%)	76.7	86.7	96.7	96.7	90	97.8
Liver	Error Rate (%)	41.1	42.6	36.7	36.7	36.7	26.4
	Accuracy (%)	58.9	57.4	63.3	63.3	63.3	73.6
Ionosphere	Error Rate (%)	52.8	52.8	31.4	30.2	44.2	21.4
	Accuracy (%)	47.2	47.2	68.6	69.8	55.8	78.6
Seed	Error Rate (%)	14.7	15.2	12.8	13.8	29.0	10.2

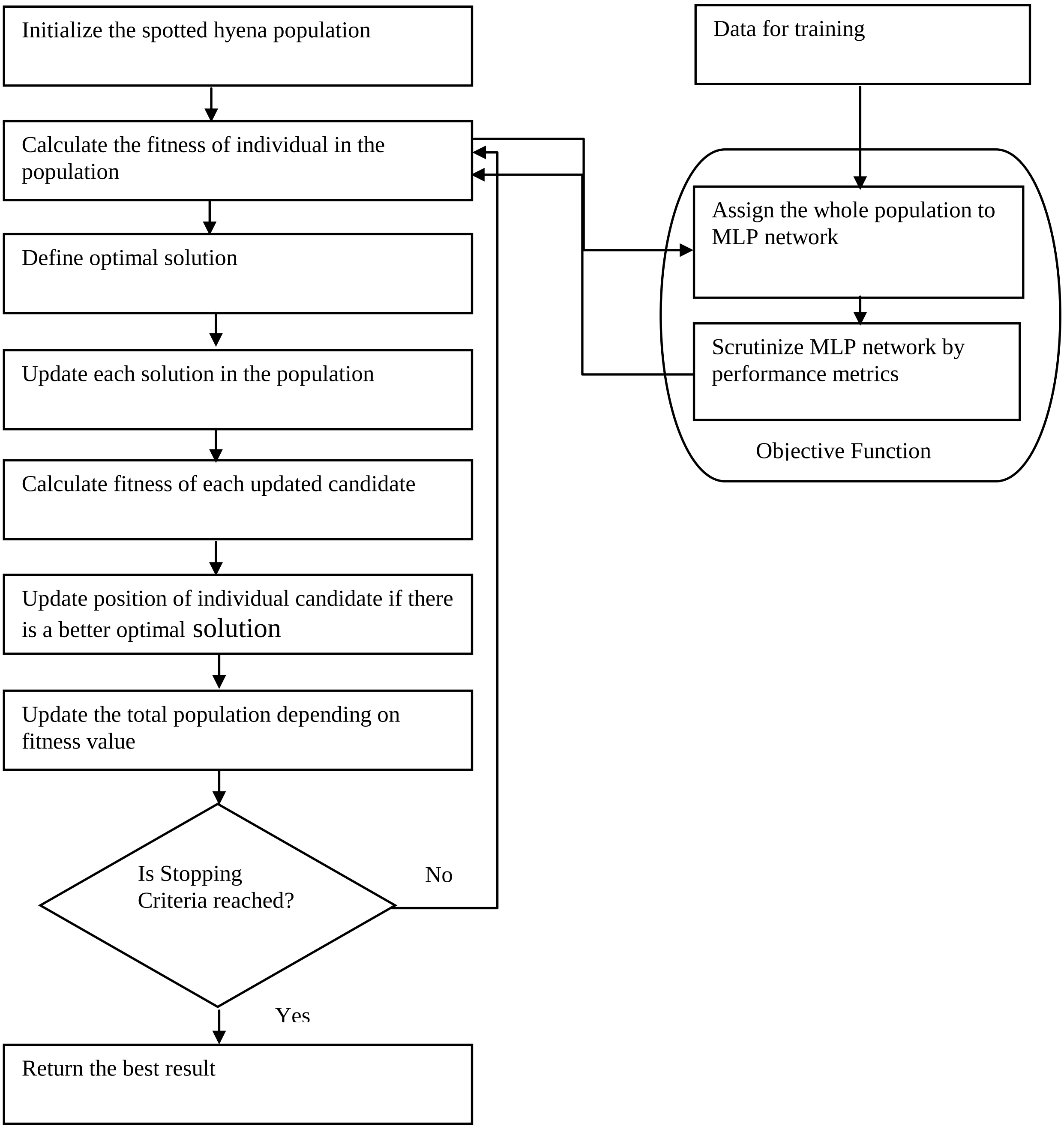
	Accuracy (%)	85.3	84.8	87.2	86.2	71	89.8
Wine	Error Rate (%)	27.4	27.4	23.5	23	15.6	13.7
	Accuracy (%)	72.6	72.6	76.5	77	84.4	86.3
Ecoli	Error Rate (%)	8.4	8.4	6.7	8.47	11.8	6.7
	Accuracy (%)	91.6	91.6	93.3	91.52	88.2	93.3
Soybean	Error Rate (%)	16.6	8.3	0	0	8.3	0
	Accuracy (%)	83.4	91.7	100	100	91.7	100
Vehicle	Error Rate (%)	21.8	20.6	16.2	16.2	19.3	8.1
	Accuracy (%)	78.2	79.4	83.8	83.8	80.7	91.9
Glass	Error Rate (%)	14.2	11.9	9.5	11.9	11.9	2.3
	Accuracy (%)	85.8	88.1	90.5	88.1	88.1	97.6
Yeast	Error Rate (%)	8.6	4.3	4.8	5.94	4.3	0.5
	Accuracy (%)	91.4	95.7	95.2	94.06	95.7	99.5

Table 5. Average Rank of Algorithms by Friedman Test

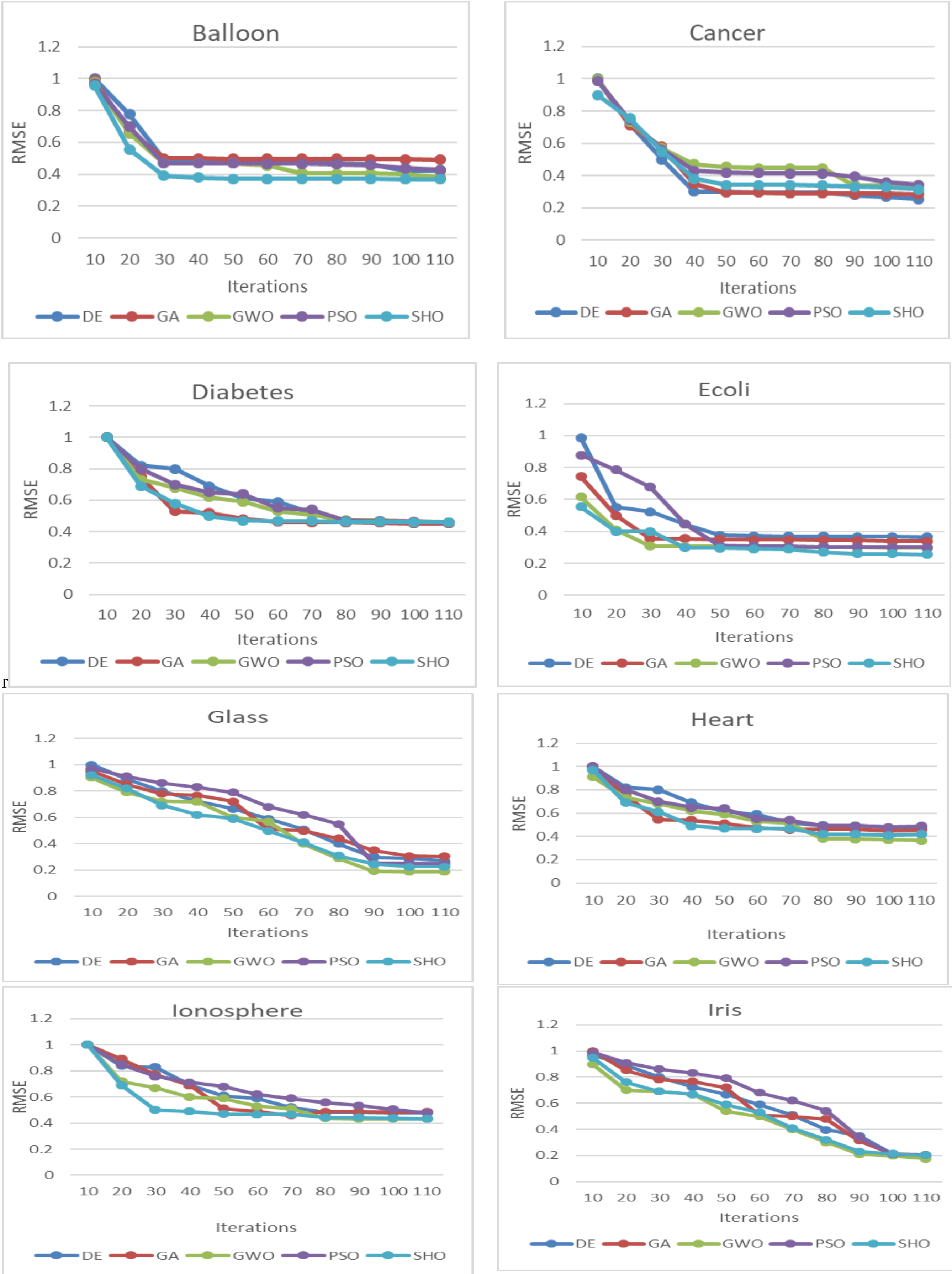
	DE	**GA**	**GWO**	**SSA**	**PSO**	**SHO**
Balloon	100 (3.5)	100 (3.5)	100 (3.5)	100 (3.5)	100 (3.5)	100 (3.5)
Cancer	98.4 (3)	98.4 (3)	98.4 (3)	98.4 (3)	98.4 (3)	96 (6)
Diabetes	74.3 (2)	73.6 (3.5)	73.6 (3.5)	75.8 (1)	72.2 (5.5)	72.2 (5.5)
Ecoli	91.6 (3.5)	91.6 (3.5)	93.3 (1.5)	91.5 (5)	88.2 (6)	93.3 (1.5)
Glass	85.8 (6)	88.1 (4)	90.5 (2)	88.1 (4)	88.1 (4)	97.6 (1)
Heart	77.8 (5.5)	79.7 (3.5)	79.7 (3.5)	83.4 (1)	77.8 (5.5)	81.5 (2)
Ionosphere	47.2 (5.5)	47.2 (5.5)	68.6 (3)	69.8 (2)	55.8 (4)	78.6 (1)
Iris	76.7 (6)	86.7 (5)	96.7 (2.5)	96.7 (2.5)	90 (4)	97.8 (1)
Liver	58.9 (5)	57.4 (6)	63.3 (3)	63.3 (3)	63.3 (3)	73.6 (1)
Seed	85.3 (4)	84.8 (5)	87.2 (2)	86.2 (3)	71 (6)	89.8 (1)
Soybean	83.4 (6)	91.7 (4.5)	100 (2)	100 (2)	91.7 (4.5)	100 (2)
Vehicle	78.2 (6)	79.4 (5)	83.8 (2.5)	83.8 (2.5)	80.7 (4)	91.9 (1)
Wine	72.6 (5.5)	72.6 (5.5)	76.5 (4)	77 (3)	84.4 (2)	86.3 (1)
XOR	75 (3.5)	75 (3.5)	75 (3.5)	75 (3.5)	75 (3.5)	75 (3.5)
Yeast	91.4 (6)	95.7 (2.5)	95.2 (4)	94.06 (5)	95.7 (2.5)	99.5 (1)
Average Rank(A_R)	**4.73**	**4.23**	**2.9**	**2.93**	**4.13**	**2.13**

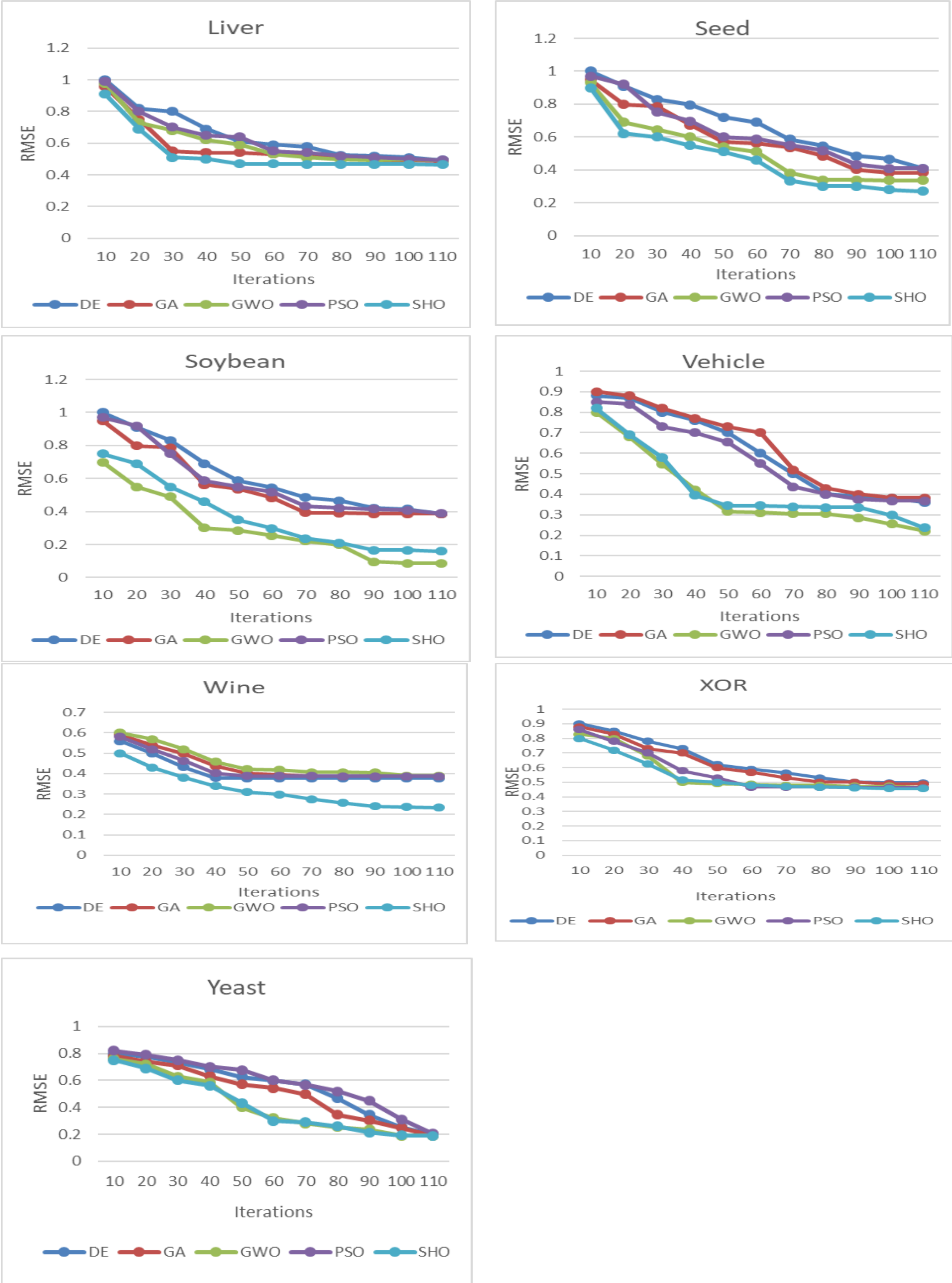
Table 6. Results from Holm's Method

i	**Algorithms**	**Z-value**	**P-value**	$\frac{\alpha}{(k-i)}$	**Hypothesis**
1	DE	-7.35	0.00004	0.01	Rejected
2	GA	-6.35	0.00004	0.012	Rejected
3	PSO	-4.88	0.00004	0.16	Rejected
4	SSA	-3.41	0.00028	0.02	Rejected
5	GWO	-1.94	0.02559	0.05	Rejected

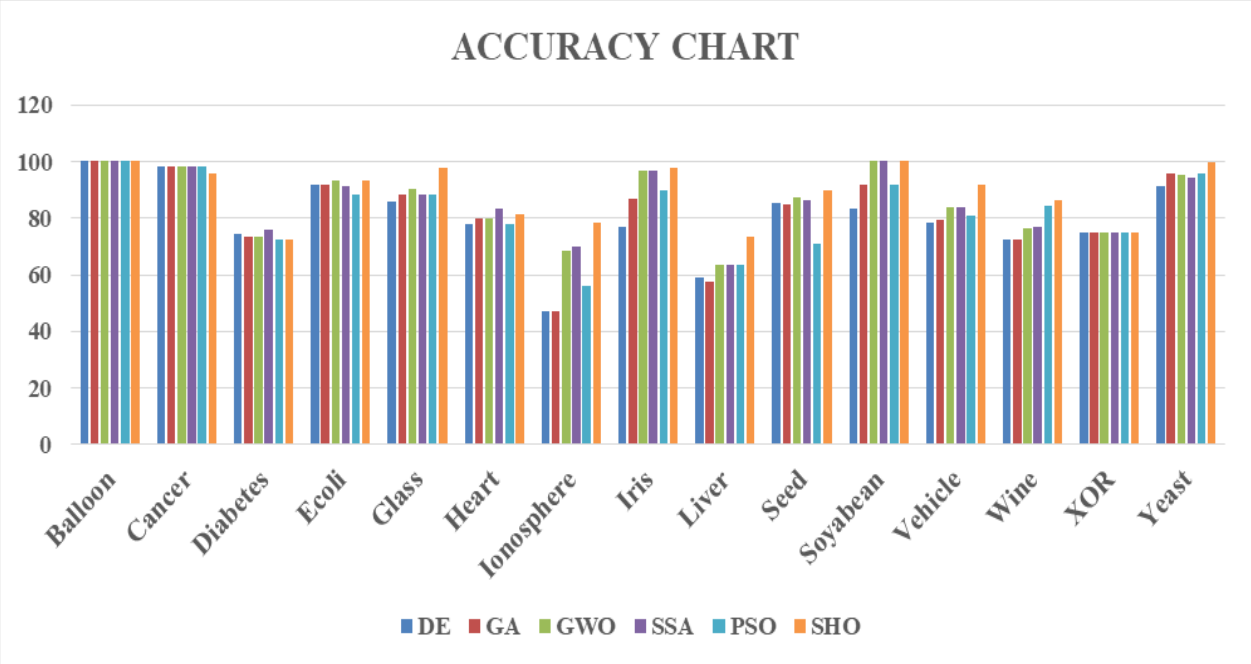


"Fig. 2" Flow diagram of SHO-MLP MODEL


Balloon
RMSE
Iterations
DE GA GWO PSO SHO
Cancer
RMSE
Iterations
DE GA GWO PSO SHO
Diabetes
RMSE
Iterations
DE GA GWO PSO SHO
Ecoli
RMSE
Iterations
DE GA GWO PSO SHO
Glass
RMSE
Iterations
DE GA GWO PSO SHO
Heart
RMSE
Iterations
DE GA GWO PSO SHO
Ionosphere
RMSE
Iterations
DE GA GWO PSO SHO
Iris
RMSE
Iterations
DE GA GWO PSO SHO



"Fig. 3" Convergence curve for fifteen standard datasets for five algorithms such as SHO, GWO, GA, PSO and DE.



"Fig. 4" Performance of the algorithms w.r.t Accuracy

IV. CONCLUSION

In this paper, we proposed the use of recent meta-heuristic algorithm, SHO for training MLPs. The research starts on training MLP to optimize the set of initial parameters i.e. weights and biases and finding minimum value of RMSE. Our main purpose to carry out the work, training MLP using SHO is its high exploration and exploitation. In the proposed work we have tested the performance of six different evolutionary algorithms including SHO in terms of MIN RMSE, Average, Standard Deviation, Accuracy, Sensitivity, Specificity and Prevalence as well. Due to the difference in datasets we have designed different MLP structures in terms of input, output and hidden layers for the training of SHO. For a better comparison of the result obtained we have taken another four conventional as well as recent evolutionary algorithms such as DE, GA, PSO, GWO and SSA. From the results, it is clear that SHO-MLP outperforms over other algorithms on maximum number of datasets. The first most advantage is higher accuracy is due to high exploitation of the algorithm, which also results in getting optimized initial parameters i.e. weights and biases. Second advantage is its capability to avoid local minima problem successfully due to its high exploration. Third advantage is its convergence speed: though it is exhibiting very promising performance in terms accuracy, its convergence rate is also very faster for majority of datasets. The proposed SHO-MLP can be used in classification and regression analysis. In future work, the SHO may be used to train higher order neural networks.

REFERENCES

1. M. Jiang, Y. Liang, X. Feng, X. Fan, Z. Pei, Y. Xue and R. Guan, "Text classification based on deep belief network and softmax regression", Neural Computing and Applications, 29(1), pp.61-70, 2018.
2. C. Affonso, A.L.D. Rossi, F.H.A. Vieira and A.C.P. de Leon Ferreira, "Deep learning for biological image classification", Expert Systems with Applications, 85, pp.114-122, 2017.
3. T. Young, D. Hazarika, S. Poria and E. Cambria, "Recent trends in deep learning based natural language processing", ieee Computational intelligenCe magazine, 13(3), pp.55-75, 2018.
4. B. Parlak and A.K. Uysal, "On Feature Weighting and Selection for Medical Document Classification", In Developments and Advances in Intelligent Systems and Applications (pp. 269-282). Springer, Cham, 2018.
5. D. Achunala, M. Sathiyanarayanan and B. Abubakar, "Traffic classification analysis using omnet++", In Progress in Intelligent Computing Techniques: Theory, Practice, and Applications (pp. 417-422). Springer, Singapore, 2018.
6. K. Nogueira, O.A. Penatti and J.A. dos Santos, "Towards better exploiting convolutional neural networks for remote sensing scene classification", Pattern Recognition, 61, pp.539-556, 2017.
7. A. Kumar, J. Kim, D. Lyndon, M. Fulham and D. Feng, "An ensemble of fine-tuned convolutional neural networks for medical image classification", IEEE journal of biomedical and health informatics, 21(1), pp.31-40, 2017.
8. B.B. Chaudhuri and C. Adak, "An approach for detecting and cleaning of struck-out handwritten text", Pattern Recognition, 61, pp.282-294, 2017.
9. M. Dashtban and M. Balafar, "Gene selection for microarray cancer classification using a new evolutionary method employing artificial intelligence concepts", Genomics, 109(2), pp.91-107, 2017.
10. J. Rusz, M. Novotný, J. Hlavnička, T. Tykalová and E. Růžička, " High-accuracy voice-based classification between patients with Parkinson's disease and other neurological diseases may be an easy task with inappropriate experimental design", IEEE Transactions on Neural Systems and Rehabilitation Engineering, 25(8), pp.1319-1321, 2017.
11. S.A. Ludwig, S. Picek and D. Jakobovic, "Classification of Cancer Data: Analyzing Gene Expression Data Using a Fuzzy Decision Tree Algorithm", In Operations Research Applications in Health Care Management (pp. 327-347). Springer, Cham, 2018.
12. I. Rish, "An empirical study of the naive Bayes classifier", In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol. 3, No. 22, August 2001, pp. 41-46). New York: IBM.
13. E. Romero and D. Toppo, "Comparing support vector machines and feedforward neural networks with similar hidden-layer weights", IEEE Transactions on Neural Networks, 18(3), pp.959-963, 2007.

Retrieval Number: B3736078219/19©BEIESP
DOI: 10.35940/ijrte.B3736.078219
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

14. S. Zhang, X. Li, M. Zong, X. Zhu and R. Wang, "Efficient knn classification with different numbers of nearest neighbors", IEEE transactions on neural networks and learning systems, 29(5), pp.1774-1785, 2018.
15. T. Lajnef, S. Chaibi, P. Ruby, P.E. Aguera, J.B. Eichenlaub, M. Samet, A. Kachouri and K. Jerbi, "Learning machines and sleeping brains: automatic sleep stage classification using decision-tree multi-class support vector machines", Journal of neuroscience methods, 250, pp.94-105, 2015.
16. J.J. Hopfield, "Artificial neural networks", IEEE Circuits and Devices Magazine, 4(5), pp.3-10, 1988.
17. W.S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity", The bulletin of mathematical biophysics, 5(4), pp.115-133, 1943.
18. S. Wang, R.V. Rao, P. Chen, Y. Zhang, A. Liu and L. Wei, "Abnormal breast detection in mammogram images by feed-forward neural network trained by Jaya algorithm", FundamentaInformaticae, 151(1-4), pp.191-211, 2017.
19. Z. Zhang, "Artificial neural network", In Multivariate Time Series Analysis in Climate and Environmental Research (pp. 1-35). Springer, Cham, 2018.
20. A. Singh and K.K. Singh, "Satellite image classification using genetic algorithm trained radial basis function neural network, application to the detection of flooded areas", Journal of Visual Communication and Image Representation, 42, pp.173-182, 2017.
21. D. Yogatama, C. Dyer, W. Ling and P. Blunsom, "Generative and discriminative text classification with recurrent neural networks", arXiv preprint arXiv: 1703.01898, 2017.
22. W. Maass, "Networks of spiking neurons: the third generation of neural network models", Neural networks, 10(9), pp.1659-1671, 1997.
23. D.E. Goldberg and J.H. Holland, 1988, "Genetic algorithms and machine learning", Machine learning, 3(2), pp.95-99, 1988.
24. 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, 13(2), pp.398-417, 2009.
25. S. Baluja, "Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning "(No. CMU-CS-94-163). Carnegie-Mellon Univ Pittsburgh Pa DeptOf Computer Science, 1994.
26. D. Simon, "Biogeography-based optimization", IEEE transactions on evolutionary computation, 12(6), pp.702-713, 2008.
27. P.J. Van Laarhoven and E.H. Aarts, "Simulated annealing", In Simulated annealing: Theory and applications (pp. 7-15). Springer, Dordrecht, 1987.
28. A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering", Information sciences, 222, pp.175-184, 2013.
29. I. FisterJr, X.S. Yang, I. Fister, J. Brest and D. Fister, "A brief review of nature-inspired algorithms for optimization", arXiv preprint arXiv: 1307.4186, 2013.
30. J. Kennedy, "Particle swarm optimization", In Encyclopedia of machine learning (pp. 760-766). Springer, Boston, MA, 2011.
31. M. Mavrovouniotis and S. Yang, "Training neural networks with ant colony optimization algorithms for pattern classification", Soft Computing, 19(6), pp.1511-1522, 2015.
32. Y. Xue, J. Jiang, B. Zhao and T. Ma, "A self-adaptive artificial bee colony algorithm based on global best for global optimization", Soft Computing, pp.1-18, 2018.
33. G. Dhiman and V. Kumar, "Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications", Advances in Engineering Software, 114, pp.48-70, 2017.
34. S. Mirjalili, "How effective is the Grey Wolf optimizer in training multi-layer perceptrons", Applied Intelligence, 43(1), pp.150-161, 2015.
35. K. Bache and M. Lichman, UCI machine learning repository, 2013.
36. F Distribution Table, March 2018, Retrieved from http://www.socr.ucla.edu/applets.dir/f_table.html.
37. Normal Distribution Table. Retrieved from http://math.arizona.edu/~rsims/ma464/standardnormaltable.pdf.
38. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, and S.M. Mirjalili, "Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems", Advances in Engineering Software, 114, pp.163-191, 2017.
39. Aljarah, I., Faris, H. and Mirjalili, S., 2018. Optimizing connection weights in neural networks using the whale optimization algorithm. *Soft Computing*, *22*(1), pp.1-15.
40. Dey, N. ed., 2017. *Advancements in Applied Metaheuristic Computing*. IGI Global.
41. Tang, R., Fong, S. and Dey, N., 2018. Metaheuristics and Chaos Theory. In *Chaos Theory*. IntechOpen.
42. Heidari, A.A., Faris, H., Aljarah, I. and Mirjalili, S., 2018. An efficient hybrid multilayer perceptron neural network with grasshopper optimization. *Soft Computing*, pp.1-18.
43. Mirjalili, S., Hashim, S.Z.M. and Sardroudi, H.M., 2012. Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. *Applied Mathematics and Computation*, *218*(22), pp.11125-11137.
44. Socha, K. and Blum, C., 2007. An ant colony optimization algorithm for continuous optimization: application to feed-forward neural network training. *Neural Computing and Applications*, *16*(3), pp.235-247.
45. Bairathi, D. and Gopalani, D., 2019. Salp Swarm Algorithm (SSA) for Training Feed-Forward Neural Networks. In *Soft Computing for Problem Solving* (pp. 521-534). Springer, Singapore.

AUTHORS PROFILE

Nibedan Panda Ph.D. Scholar, Department of CSE, VSSUT, Burla, Odisha, India**.**

Dr. Santosh Kumar Majhi Asst. Professor, Department of CSE, VSSUT, Burla, Odisha, India.