Training feedforward neural networks using hybrid particle swarm optimization, Multi-Verse Optimization

*

1st Rabab Bousmaha
GeCoDe Laboratory
Department of Computer Science
University of Saida
Saida, Algeria
Rabab.bousmaha@gmail.com

2nd Reda Mohamed Hamou
GeCoDe Laboratory
Department of Computer Science
University of Saida
Saida, Algeria
hamoureda@yahoo.fr

3rd Amine Abdelmalek
GeCoDe Laboratory
Department of Computer Science
University of Saida
Saida, Algeria
amineabd1@yahoo.fr

***Abstract*—The learning process of artificial neural networks is an important and complex task in the supervised learning field. The main difficulty of training a neural network is the process of fine-tuning the best set of control parameters in terms of weight and bias. This paper presents a new training method based on hybrid particle swarm optimization with Multi-Verse Optimization (PMVO) to train the feedforward neural networks. The hybrid algorithm is utilized to search better in solution space which proves its efficiency in reducing the problems of trapping in local minima. The performance of the proposed approach was compared with five evolutionary techniques and the standard momentum backpropagation and adaptive learning rate. The comparison was benchmarked and evaluated using six bio-medical datasets. The results of the comparative study show that PMVO outperformed other training methods in most datasets and can be an alternative to other training methods.**

***Index Terms*—Particle swarm optimization, Multi-Verse Optimization, Training feedforward neural networks, Real world Datasets**

I. Introduction

Artificial neural network (ANN) is one of the most important data mining techniques. It has been successfully applied to many fields. The feedforward multilayer perceptron (MLP) is one of the best-known neural networks. The multilayer perceptron (MLP) consists of three layers composed of neurons organized into input, output and hidden layers. The success of an MLP generally depends on the training process that is determined by training algorithms. The objective of the training algorithms is to find the best connection between weights and biases that minimize the classification error. Training algorithms can be classified into two classes: gradient-based and stochastic search methods. Backpropagation (BP) and its variants are gradient-based methods and considered as one of the most popular techniques used to train the MLP neural network. Gradient-based methods have many drawbacks, such as the slow convergence, the high dependency on the initial value of weights and biases and the tendency to be trapped in local minima [1].To address these problems, stochastic search methods, such as metaheuristics have been proposed as alternative methods for training feedforward neural network. Metaheuristics have many advantages: they apply to any type of ANN with any activation function [2], are particularly useful for dealing with large complex problems that generate many local optima [3] [4]. Genetic algorithm (GA) and Particle Swarm Optimization (PSO) considered as the most well-known nature inspired MLP trainers. Montana and Davis proposed one of the earliest works on training the feedforward neural network (FFNN) with GA [22]. They showed that GA outperform BP when solving real problems.Slowik and Bialko [23] employed Differential Evolution (DE) for training MLP and showed that it has promising performance compared to BP and Levenberg-Marquardt methods.

Others metaheuristics algorithms have been applied in training feedforward MLP, such as the modified BAT [5], Multi-Verse Optimization MVO [6], Whale Optimization Algorithm (WOA) [7], Grey Wolf Optimizer (GWO) [8] [9], Biogeography Based on Optimizer (BBO) [10], Moth-Flame Optimization (MFO) [11] and Improved Monarch Butterfly Optimization (IMBO) [12]. Furthermore, several hybrid algorithms have been proposed to train a neural network. Tarkhaneh and Shen [13] suggested a hybrid approach to neural network training by combining PSO, Mantegna Levy flight and neighbor search (LPSONS). The comparison experiments showed that the proposed algorithm can find optimal results. Khan et al [14] introduced a new method based on two algorithms, accelerated particle swarm optimization (APSO) and cuckoo search (CS), named HACPSO. The comparison results demonstrated that the proposed algorithm outperforms other algorithms in term of accuracy, MSE and standard deviation. This paper presents a new training approach based on hybrid particle swarm optimization (PSO) with Multi-Verse Optimization (MVO), called PMVO, to train the feedforward neural network (FFNN). Six datasets were solved by the proposed trainer. Moreover, the application of the trainer was investigated in bio-medical. The performance of PMVO was compared with five well-known trainer metaheuristics algorithms in the literature: PSO [15], MFO [11], MVO [6], WOA [7], HACPSO [14].

II. ARTIFICIAL NEURAL NETWORKS (ANNS)

An artificial neural network (ANN) is a computational model based on the structure and functions of the biological brain and nervous system. The feedforward neural network (FFNN) is one of the most popular types of artificial neural network [6].FFNN has three interconnected layers. The first layer consists of input neurons. These neurons send the data to the second layer, called the hidden layer, which sends the output neurons to the third layer. In FFNN, the information travels in one direction, from the input layer to the output layer. The node or the artificial neuron multiplies each of these inputs by weight, as shown in (1):

$$S_j = \sum_{i=1}^{n} w_{i,j} I_i + \beta_j \tag{1}$$

where, n is the total number of neuron inputs, Wij is the connection weight connecting I_j to neuron j and β_j is a bias weight [6]. Then, the node or the artificial neuron adds the multiplications and sends the sum to a transfer function, for example, Sigmoid function presented in (2):

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

The output of the neuron j can be described as follows (3):

$$y_j = f_j(\sum_{i=1}^{n} w_{i,j} I_i + \beta_j) \tag{3}$$

After building the neural network, the set of network weights are adjusted to approximate the desired results. This process is carried out by applying a training algorithm to adapt the weights until error criteria are met [6].

III. PARTICLE SWARM OPTIMIZATION (PSO)

in 1995 Russell Eberhart and James Kennedy have invented the particle swarm optimization which is a population-based stochastic optimization technique inspired by birds flocking around food sources. like each other evolutionary computational algorithms. In PSO, each individual is a bird in the search space. We call it a particle. All of the particles have fitness values which are evaluated by the fitness function to be optimized and flies in the space with a velocity which is dynamically adjusted according to its own flying experience [16].

$$V_{i,j}^{t+1} = V_{i,j}^{t} W + C_1 R_1 (Pbest^t - X^t) + C_2 R_2 (Gbest^t - X^t) \tag{4}$$

$$X^{t+1} = X^t + V^{t+1} - i = 1, 2...NP) And (j = 1, 2...NG) \tag{5}$$

where $Pbest^t$ and $Gbest^t$ denote the best particle position and best group position and w is inertia weight $w = w^{max} - (\frac{(w^{max} - w^{min}) * iteration}{maxiteration})$,$C_1, C_2$ two positive constants , R_1,R_2 are random numbers in the interval of [0, 1], $V_{i,j}^{t+1}$ is the velocity of jth member of ith particle at iteration number (t) and (t+1). The new position values X^{t+1} are obtained by adding the velocity updates determined by the formula given in Equation (5).

IV. MULTI-VERSE OPTIMIZATION(MVO)

Multiverse optimization proposed by Syed Ali mirjalili in 2015 [17] As Inspired by the concepts of white holes, black holes, and wormholes in the multi-verse theory and big bang theory. In this algorithm, the models of these three concepts are developed to perform exploration and exploitation and local search. The fitness function for each search agent is indicated by the inflation rate, and each object and each universe in the search agent represent a candidate solution and a variable in the candidate solution.

In this algorithm, the larger universes tend to send objectives to smaller universes. A large universe is defined based on inflation rate in the multi-verse theory. The following rules are applied to the universes of the MVO:

- If the inflation rate rate is higher, the probability of having a white hole is higher.
- If the inflation rate rate is higher, the probability of having black holes is lower.
- Universes having higher inflation rate rate send the objects through white holes.
- Universes having lower inflation rate rate tend to receive more objects through black holes.
- The objects of all universes may be replaced by the objects of the universe with the greater inflation rate.

The mathematical model of this algorithm is as follows:

$$\mathrm{x}_i^j = \begin{cases} \begin{cases} x_j + TDR + ((ub_j - lb_j) * r_4 + lb_j), \text{if } \mathrm{r}_3 < 0.5 \\ x_j - TDR + ((ub_j - lb_j) * r_4 + lb_j), \text{if } \mathrm{r}_3 \geq 0.5 \end{cases} & \text{,if } \mathrm{r}_2 < WEP \\ x_i^j, \text{ if } \mathrm{r}_2 \geq WEP \end{cases} \tag{6}$$

Where x_j indicates the jth variable in the bests universe, lb_i indicates the lower bound in jth variable, ub_i shows the upper bound in jth variable, r_2 ,r_3 , r_4 are random numbers in the interval of [0, 1], TDP/WEP are coefficients, and x_i^j indicates the jth parameter in ith universe.

V. HYBRID PSO-MVO:

Hybrid PSO-MVO is sequential combination of PSO and MVO. The algorithm merges the best strength of both PSO in exploitation and MVO in exploration towards the optimum solution when the universe value of MVO replace the Pbest value of PSO [20] [21]. In this paper we propose a novel training algorithm based on this algorithm for the first time in the following section.The equation can be written as follows:

$$V_{i,j}^{t+1} = V_{i,j}^{t} W + C_1 R_1 (Universes^t - X^t) + C_2 R_2 (Gbest^t - X^t) \tag{7}$$

Step 1: Initialize the PSO values
Step 2: Evaluate the fitness function of each particle
Step 3: Determine Gbest from the Pbest value
Step 4: updated velocity and position values of each particule
Step 5: verify the solution whether it is feasible or not
Step 6: steps 2 to 5 were repeated until the maximum number of iterations was reached.
Step 9: Use the optimal solutions of PSO as boundary to MVO algorithm
Step 10: Initialize the MVO values
Step 11: Evaluate the inflation rate of the universe (fitness function)
Step 12: Update the position of the universes
Step 13: if the convergence criterion is reached; get the results
Step 14: if the convergence criterion is not reached; continue the process from step 11-14

VI. PMVO FOR TRAINING MLP

This section presents the proposed approach based on the PMVO to train the MLP network named PMVO.Two important points are taken into consideration: the fitness function and the representation of the PMVO solutions. In this work, the PMVO algorithm was applied for training MLP network with a single hidden layer and each PMVO solution (weights and biases) was formed by three parts: the connection weights between the input layer and the hidden layer, the weights between the hidden layer and the output layer, and the bias weights.The length of each solution vector is given by equation (8), where n is the number of input features and m is the number of neurons in the hidden layer [6].

$$IndividualLength = (n \times m) + (2 \times m) + 1 \tag{8}$$

PMVO solutions are implemented as real number vectors when each vector belongs to the interval [-1, 1]. The mean square error (MSE) was used to measure the fitness value of PMVO solutions.MSE was calculated based on the difference between the estimated and actual values of the neural network using the training datasets, as shown in equation (9), where n is the number of samples in the training dataset y and $\hat{y}$ are respectively the actual and predicted values:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^{2} \tag{9}$$

VII. EXPERIMENTS AND RESULTS

This section presents the evaluation of the proposed PMVO for training MLP networks on six well-known datasets, which were selected from University Of California Irvine machine learning (UCI)[1] and Kaggle[2] dataset repositories. Table I shows the classification of these datasets in terms of features number, classes, training and testing samples. The comparison of PMVO was carried out with five approaches used to train feedforward neural network in the literature: PSO [15], MFO [11], MVO [6], WOA [7], HACPSO [14] In addition, the proposed algorithm was compared with standard momentum Back-Propagation and adaptive learning rate and (BP), which are gradient-based algorithms.

VIII. EXPERIMENTAL SETUP

The proposed trainer and other algorithms were implemented with Python language and a personal computer with Intel(R) Core(TM) CPU 1.60 GHz 2.30 GHz, 64 Bits Windows 7 operating system and 4 GB (RAM).

The metaheuristics are sensitive to the value of their parameters, which requires a careful initialization. Therefore, the control parameters recommended in the literature were used [15] [6] [7] and summarized in Table II. All datasets were divided into 66% for training and 34% for testing. Moreover, all features were mapped to the interval [0, 1] to eliminate the effect of features that have different scales. Min-max normalization is applied to perform a linear transformation on the original data, were v' is the normalized value of v in the range $[min_A, max_A]$ as given in (10).

$$v' = \frac{v_i - min_A}{max_A - min_A} \tag{10}$$

In the literature, there is no standard method for selecting the number of hidden neurons. In this work, the method proposed in [18] [19] [6] was used; the number of neurons in the hidden layer equals to $2N + 1$, where N, is a number of features in the dataset.

IX. RESULTS

All algorithms were tested ten times for every dataset and the population size and the maximum number of generations were set to 50 and 200, respectively.

Table IV shows the statistical results: average, best, worst and standard deviation of classification accuracy. The results of PMVO outperformed other approaches in breast cancer, blood, liver, vertebral with an average accuracy of 0.962, 0.766, 0.752, 0.839. In addition, PMVO was ranked second in diabetes and Parkinson datasets with an average accuracy of 0.783, 0.842 respectively. Moreover, it can also be seen that the PMVO has a smaller Std which indicates that PMVO is stable. Table V shows the average, best and worst MSE with standard deviation, obtained for each algorithm. As a result, it can be noted that PMVO outperforms other techniques in four datasets: breast cancer, blood, liver and vertebral with an average MSE of 0.032, 0.168, 0.176, 0.131, respectively. In addition, it can be also noticed that PMVO has small standard deviation value for all datasets which proves the efficiency and robustness of this algorithm.

Figures 1 2 3 4 5 and 6 show the convergence curves of all metaheuristics training algorithms based on the average values of MSE.The convergence curves show that PMVO has the lowest value of MSE for four datasets: breast cancer, blood, liver and vertebral. Moreover, PMVO has the fastest

[1] http://archive.ics.uci.edu/ml/

[2] https://www.kaggle.com/datasets

convergence speed in liver, vertebral, blood and European datasets. For diabetes dataset, PMVO provides a very close performance compared to MVO algorithm. These results show that PMVO has a faster convergence and a better optimization than other metaheuristic algorithms.

TableIII shows the average ranks obtained by each optimization technique in the Friedman test. The comparative shows that the proposed algorithm outperforms other algorithms.

TABLE I
Summary of the classification datasets

DataSets	Features	Training samples	Testing samples
Blood	4	493	255
Breast cancer	8	461	238
Diabetes	8	506	262
Vertebral	6	204	106
Liver	6	79	41
Parkinson	22	128	67

TABLE II
Initial parameters of the optimization algorithms

Parameter	Definition	Value
	Acceleration constants	[2.1,2.1]
PSO	Intertia weights	[0.9,0.6]
	Number of particles	50
MVO	Minimum wormhole	0.2
	Maximum wormhole	1
	Population size	50
	Numer of generations	200
	Number of search agents	50
MFO	b	1
	t	[-1,1]
	Population size	200
	Numer of generations	200
	r	[0,1]
WOA	Population size	50
HACPSO	B	0.5
	Population size	50
	Numer of generations	200

TABLE III
Average Ranking of the techniques(Friedman)

Algorithms	Ranking
PMVO	1.33
PSO	3.25
MFO	4.08
MVO	3
WOA	4.92
HACPSO	4.91
BP	6.5

X. Conclusion

In this paper, we have proposed a new training approach based on Particle swarm optimization, Multi-Verse Optimization to train the feedforward neural network. The training method took into account the capabilities of the PMVO in terms of high exploration and exploitation to locate the optimal values for weights and biases of FFNN. The approach is

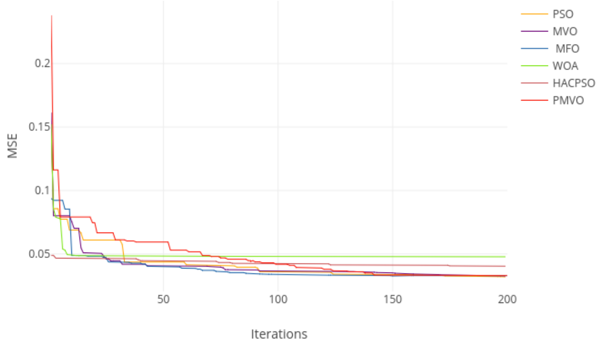

Fig. 1. MSE convergence curve of Breast cancer

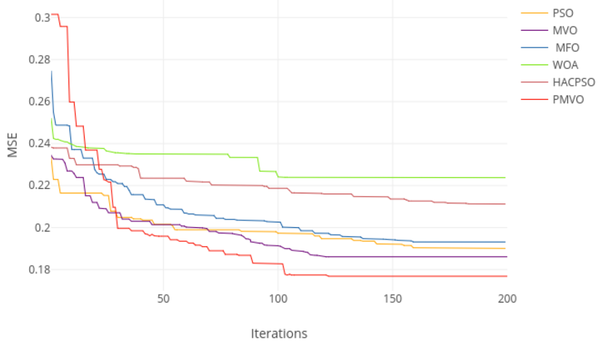

Fig. 2. MSE convergence curve of Blood

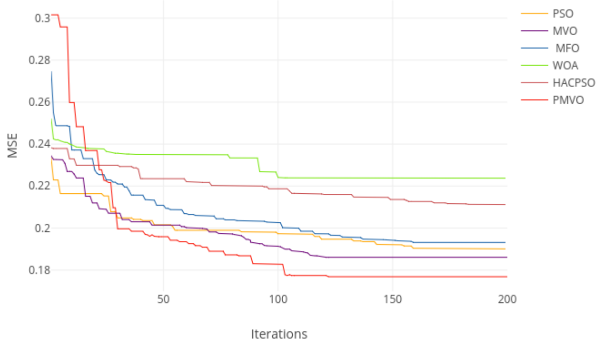

Fig. 3. MSE convergence curve of Diabetes

TABLE IV
Accuracy results

		PMVO	PSO	MFO	MVO	WOA	HACPSO	BP
B.cancer	Avg	0.962	0.959	0.958	0.958	0.956	0.959	0.744
	STd	0.002	0.003	0.002	0.002	0.006	0.004	0.254
	Best	0.964	0.960	0.963	0.960	0.963	0.965	0.945
	Worst	0.957	0.953	0.954	0.954	0.947	0.952	0.680
Blood	Avg	0.766	0.762	0.763	0.764	0.762	0.760	0.744
	STd	0.003	0.002	0.005	0.009	0.004	0.003	0.254
	Best	0.768	0.765	0.765	0.784	0.774	0.760	0.945
	Worst	0.762	0.760	0.760	0.760	0.758	0.758	0.680
Diabetes	Avg	0.783	0.780	0.724	0.792	0.720	0.768	0.619
	STd	0.018	0.011	0.008	0.006	0.028	0.006	0.08
	Best	0.792	0.796	0.735	0.802	0.750	0.774	0.690
	Worst	0.732	0.758	0.709	0.782	0.657	0.764	0.601
Liver	Avg	0.752	0.722	0.722	0.737	0.665	0.679	0.519
	STd	0.003	0.017	0.010	0.006	0.030	0.020	0.055
	Best	0.785	0.748	0.744	0.744	0.700	0.709	0.586
	Worst	0.721	0.700	0.709	0.726	0.603	0.669	0.495
Vertebral	Avg	0.839	0.836	0.836	0.836	0.774	0.717	0.651
	STd	0.006	0.012	0.007	0.005	0.052	0.015	0.170
	Best	0.845	0.848	0.843	0.838	0.862	0.833	0.866
	Worst	0.828	0.808	0.82	0.828	0.676	0.794	0.627
Parkinson	Avg	0.842	0.841	0.802	0.802	0.824	0.875	0.750
	STd	0.029	0.040	0.022	0.046	0.016	0.013	0.199
	Best	0.882	0.898	0.0.828	0.867	0.852	0.905	0.849
	Worst	0.788	0.773	0.773	0.695	0.800	0.863	0.623

TABLE V
MSE Results

		PMVO	PSO	MFO	MVO	WOA	HACPSO	BP
B.cancer	Avg	0.032	0.032	0.033	0.032	0.047	0.040	0.049
	Std	0.001	0.001	0.001	0.002	0.004	0.001	0.015
	Best	0.030	0.030	0.310	0.030	0.043	0.038	0.030
	Worst	0.0336	0.326	0.036	0.032	0.058	0.041	0.050
Blood	Avg	0.168	0.178	0.176	0.170	0.180	0.169	0.174
	Std	0.005	0.003	0.005	0.008	0.004	0.003	0.009
	Best	0.160	0.182	0.174	0.155	0.174	0.162	0.172
	Worst	0.174	0.182	0.177	0.181	0.187	0.175	0.175
Diabetes	Avg	0.151	0.155	0.171	0.147	0.186	0.163	0.179
	Std	0.002	0.004	0.005	0.001	0.013	0.002	0.066
	Best	0.1491	0.151	0.165	0.145	0.168	0.160	0.168
	Worst	0.153	0.157	0.175	0.149	0.213	0.166	0.180
Liver	Avg	0.176	0.191	0.193	0.186	0.220	0.212	0.210
	Std	0.004	0.003	0.002	0.004	0.007	0.002	0.003
	Best	0.170	0.185	0.189	0.179	0.210	0.208	0.190
	Worst	0.180	0.195	0.194	0.194	0.233	0.215	0.220
Vertebral	Avg	0.131	0.134	0.136	0.133	0.163	0.146	0.168
	Std	0.006	0.002	0.002	0.002	0.018	0.002	0.015
	Best	0.120	0.132	0.135	0.131	0.137	0.142	0.160
	Worst	0.138	0.135	0.137	0.134	0.202	0.147	0.172
Parkinson	Avg	0.134	0.141	0.158	0.147	0.165	0.119	0.158
	STd	0.014	0.023	0.019	0.020	0.030	0.005	0.018
	Best	0.119	0.099	0.135	0.125	0.127	0.112	0.137
	Worst	0.157	0.182	0.203	0.197	0.228	0.130	0.203

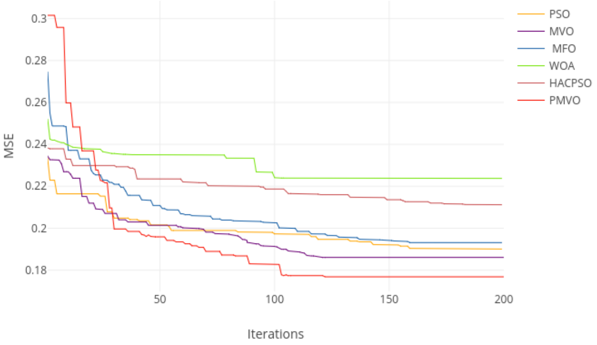

Fig. 4. MSE convergence curves of Liver

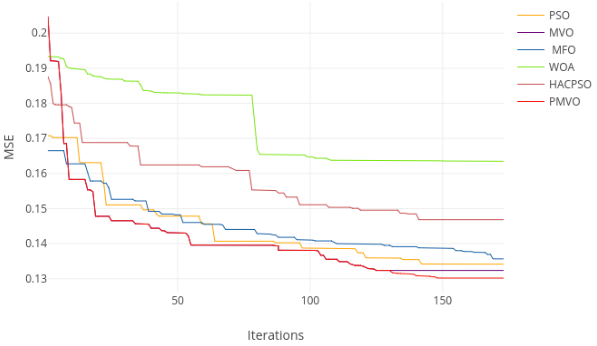

Fig. 5. MSE convergence curve of Vertebral

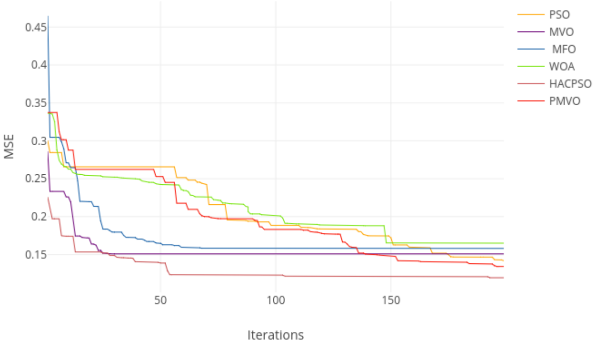

Fig. 6. MSE convergence curve of Parkinson

proposed to minimize the training error and to increase the accuracy. The approach is benchmarked and evaluated using six standard bio-medical datasets.

The comparison between the proposed algorithm and PSO, MFO, MVO, WOA, HACPSO and standard BP with momentum term and adaptive learning rate shows the superiority of the PMVO algorithm with high accuracy and small MSE in most of the datasets compared with other training algorithms. Moreover, the small value of standard deviation shows that our trainer is robust and stable. Finally, from the experiment, we can conclude that PMVO can give good results and can be an alternative to other training methods.

In future works, we focus on how to extend this work to solve a more real world problem and we test the performance of PMVO to train other types of neural networks.

REFERENCES

[1] J.-R. Zhang, J. Zhang, T.-M. Lok, and M. R. Lyu, A hybrid particle swarm optimizationback-propagation algorithm for feedforward neural network training, Applied Mathematics and Computation, vol. 185, no. 2, pp. 10261037, 2007.

[2] S. Kiranyaz, T. Ince, A. Yildirim, and M. Gabbouj, Evolutionary artificial neural networks by multi-dimensional particle swarm optimization, Neural Networks, vol. 22, no. 10, pp. 14481462, 2009.

[3] T. Kenter, A. Borisov, C. V. Gysel, M. Dehghani, M. D. Rijke, and B. Mitra, Neural Networks for Information Retrieval, Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining - WSDM 18, 2018.

[4] L. Wang, Y. Li, J. Huang, and S. Lazebnik, Learning Two-Branch Neural Networks for Image-Text Matching Tasks, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 2, pp. 394407, Jan. 2019.

[5] N. S. Jaddi, S. Abdullah, and A. R. Hamdan, Optimization of neural network model using modified bat-inspired algorithm, Applied Soft Computing, vol. 37, pp. 7186, 2015.

[6] H. Faris, I. Aljarah, and S. Mirjalili, Training feedforward neural networks using multi-verse optimizer for binary classification problems, Applied Intelligence, vol. 45, no. 2, pp. 322332, May 2016.

[7] I. Aljarah, H. Faris, and S. Mirjalili, Optimizing connection weights in neural networks using the whale optimization algorithm, Soft Computing, vol. 22, no. 1, pp. 115, 2016.

[8] M. F. Hassanin, A. M. Shoeb, and A. E. Hassanien, Grey wolf optimizer-based back-propagation neural network algorithm, 2016 12th International Computer Engineering Conference (ICENCO), 2016.

[9] H. Faris, S. Mirjalili, and I. Aljarah, Automatic selection of hidden neurons and weights in neural networks using grey wolf optimizer based on a hybrid encoding scheme, International Journal of Machine Learning and Cybernetics, vol. 10, no. 10, pp. 29012920, Jun. 2019.

[10] I. Aljarah, H. Faris, S. Mirjalili, and N. Al-Madi, Training radial basis function networks using biogeography-based optimizer, Neural Computing and Applications, vol. 29, no. 7, pp. 529553, 2016.

[11] H. Faris, I. Aljarah, and S. Mirjalili, Evolving Radial Basis Function Networks Using MothFlame Optimizer, Handbook of Neural Computation, pp. 537550, 2017.

[12] H. Faris, I. Aljarah, and S. Mirjalili, Improved monarch butterfly optimization for unconstrained global search and neural network training, Applied Intelligence, vol. 48, no. 2, pp. 445464, 2017.

[13] O. Tarkhaneh and H. Shen, Training of feedforward neural networks for data classification using hybrid particle swarm optimization, Mantegna Lvy flight and neighborhood search, Heliyon, vol. 5, no. 4, 2019.

[14] A. Khan, R. Shah, M. Imran, A. Khan, J. I. Bangash, and K. Shah, An alternative approach to neural network training based on hybrid bio meta-heuristic algorithm, Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 10, pp. 38213830, 2019.

[15] R. Mendes, P. Cortez, M. Rocha, and J. Neves, Particle swarms for feedforward neural network training, Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN02 (Cat. No.02CH37290), 2002.

[16] J. Kennedy, ; R.Eberhart, Particle Swarm Optimization, Proceedings of IEEE International Conference on Neural Networks. ICNN.1995.488968 IV. pp. 19421948 , 1995.

[17] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, Multi-Verse Optimizer: a nature-inspired algorithm for global optimization, Neural Computing and Applications, vol. 27, no. 2, pp. 495513, 2015.

[18] S. Mirjalili, S. M. Mirjalili, and A. Lewis, Let a biogeography-based optimizer train your Multi-Layer Perceptron, Information Sciences, vol. 269, pp. 188209, 2014.

[19] Mirjalili, S. (2015). S. Mirjalili, How effective is the Grey Wolf optimizer in training multi-layer perceptrons, Applied Intelligence, vol. 43, no. 1, pp. 150161, 2015.

[20] Sagarika, T.R.Jyothsna, Tunning of PSO algorithm for single machine and multi machine power system using STATCOM controller, international journal of engineering and technology, vol 2, issue 4, pp.175-182, 2015.

[21] K.Karthikeyan, P.K.Dhal, Transient stability enhancement by optimal location and tuning of STATCOM using PSO, Procedia technology, 2015.

[22] DJ.Montana , L.Davis Training feedforward neural networks using genetic algorithms. In: Proceedings of the 11th International Joint Conference on Artificial Intelligence - Volume 1, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, IJCAI89, pp 762767, 1989.

[23] A.Slowik, M.Bialko Training of artificial neural networks using differential evolution algorithm. In: Conference on Human System Interactions,IEEE, pp 6065,2008.