

## Enhancing extreme learning machines classification with moth-flame optimization technique

Oyekale Abel Alade<sup>1</sup>, Roselina Sallehuddin<sup>2</sup>, Nor Haizan Mohamed Radzi<sup>2</sup>

<sup>1</sup>Department of Computer Science, School of ICT, The Federal Polytechnic, Bida, Nigeria

<sup>2</sup>Department of Computer Science, School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Johor Bahru, Malaysia

### Article Info

#### Article history:

Received Apr 15, 2021

Revised Feb 26, 2022

Accepted Mar 10, 2022

#### Keywords:

Classification

Extreme learning machines

Moth-flame optimization

Performance improvement ratio

### ABSTRACT

Extreme learning machine (ELM) algorithm assigns the input weights and biases in a “one-time stamp” fashion, this method makes the algorithm to be ill-conditioned and reduces its classification accuracy. The contribution of this work is the enhancement of the performance of ELM with the moth-flame optimization (MFO) algorithm to improve classification accuracy. A hybrid of the Moth-flame optimization and extreme learning machine (MFO-ELM) algorithm is implemented in MATLAB. MFO ensures a concurrent simulation of exploration and exploitation of the search space to select an optimum candidate solution. The candidate solution is reshaped into input weights and biases for ELM classification. The hybrid algorithm is validated on five life-selected datasets. The performance improvement of MFO-ELM is compared with ELM-optimized particle swarm optimization (PSO-ELM) and competitive swarm optimization (CSO-ELM) algorithms. The improvement rates are qualitatively and quantitatively evaluated to show the improvement of MFO-ELM on ELM and the other meta-heuristic algorithms. MFO-ELM improved the accuracies of the basic ELM in all 100% of the simulations and performed better than the other meta-heuristic algorithms in 80% of the simulations. The performance of MFO-ELM is more competitive, and it is recommended for solving classification problems.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



### Corresponding Author:

Oyekale Abel Alade

Department of Computer Science, School of ICT, The Federal Polytechnic

Bida, Nigeria

Email: kaleabel@gmail.com

## 1. INTRODUCTION

Machine learning (ML) algorithms have parameters that govern their operational efficiencies [1]. The parameters are initiated before training commences. For the past three decades, most applications of machine learning use a single layer feed forward network (SLFN). The backbone of the SLFN training algorithms is the backpropagation (BP) method. That is, parameters learn at each iteration based on the first-order instantaneous value of the cost function [2]. These parameters have to be tuned to minimize the cost function. The major challenge is that the parameters require iterative tuning, which results in the slowness of the machine learning algorithms [3]. With every training, the old and new datasets retrain in batch learning algorithms [4]. This method consumes much time, therefore a quest among researchers for a fast and scalable machine learning algorithm to alleviate the problem of long training time.

Huang proposed extreme learning machines (ELM) [5]. ELM learning principle is essentially a linear model. ELM randomly assigns the input weights and biases to the hidden neurons, then computes the output of the hidden neurons and uses Moore Penrose generalized inverse to determine the output weights analytically [6]. Therefore, the input weights and biases no longer require iterative tuning as it was in the conventional

learning machines [7], [8]. It is faster than the backpropagation (BP) algorithms as it is a one-pass algorithm [9]. Thus, ELM has been exploited by researchers for classification in recent times [10]. However, ELM has the problem that its weights and biases which are randomly assigned make it to be ill-conditioned, this affects its accuracy and generalization performance [11]. This problem must be addressed to exploit its fast training advantage in the classification of datasets.

Deng *et al.* [12] adopted regularization parameters to address the problem of random assignment of input weights and biases of ELM. They based the approach on ridge regression theory and weighted least squares. Martínez *et al.* in [13] improved the work of Deng *et al.* They proposed the use of ridge regression, elastic net, and lasso methods to prune the size of hidden neurons in ELM architecture. They validated their work with some regression benchmark tasks, and it was proved to scale a more compact network with a competitive result when compared with ELM. However, [10] appraised the generalization of their algorithms but shows that the ridge regularized ELM requires large memory space, and since large matrix inversion is involved, the computational cost is high. Therefore, they proposed the generalized regularized ELM (GR-ELM) approach for multiclass classification tasks. The approach combined the Frobenius norm and  $\ell_{2,1}$  norm of output weights as ELM penalty. The R-ELM was maintained for binary classification tasks. They used alternating direction method for multiplier (ADMM) for implementation. They came up with a more compact network structure. However, the approach becomes more complex and the issue of computational cost remains unresolved.

Other than procedural algorithms above, recent work enhance the performance ELM classification with optimization techniques [14], [15]. Optimization techniques select good parameters for the efficient performance of the ELM algorithm [16]. They generate a Pareto front, from which updating searches an optimum candidate solution. Physical activities like hill-climbing, migration, maneuvering emissions; biological behavior of Beetles, Birds, Bees, Ants, Bats, Fish, Cats; and other evolutionary forms had been modeled in optimization to improve ELM [17]-[19].

Eshtay *et al.* [6] employed competitive swarm optimization based neural network (CSONN) to control the complex nature of the ELM network. The algorithm optimized the weights and biases and determined the size of the hidden nodes dynamically. They used 23 benchmark functions for the simulations and compared the results with the static rule based ELM and some other meta-heuristic based ELM. Their results improve the generalization of ELM and perform better than static rule-based ELM. However, the algorithm was more computational complex than the traditional ELM and the static rule-based ELM. Yang and Duan [20] proposed a hybrid model of artificial bee colony (ABC) and differential evolution (DE) optimization techniques to improve the parameter selection of ELM. The model improved the generalization performance with less processing time offered by ELM. The deficiency of initial random assignment of input weights and biases was also improved, and the results of the classification were also improved. However, the exploitation of ABC is poor [21] and the DE is computationally intensive [22]. Xie *et al.* [23] proposed collaborative ELM to prevent repeated computations that emanate from data redundancy. They employed the use of confidence interval to enhance the traditional ELM algorithm. With the approach, they were able to eliminate redundant computations of the neural network nodes. The approach improves the efficiency of ELM classification. However, the approach did not consider the selection optima input weights and biases for ELM, therefore, ELM is still subject to being stuck in a local minimum. There are many other state-of-the-arts optimization techniques used recently to improve the performance of ELM. These include grey wolf optimizer (GWO), bat algorithm (BA), bacterial foraging optimization (BFO), [20], [24]-[26], and many more. These hybrid techniques improved the performance of ELM in some ways; however, they are challenged with being stock in local minimal, which reduced its classification accuracy. MFO has a better trade-off between the exploration and exploitation in the search space than any of the above schemes, with less computational cost. Therefore, we propose an enhanced MFO-ELM. The proposed algorithm is set to achieve the following: i) To improve ELM generalization performance. ii) Introducing a meta-heuristic algorithm to select optima input weights and biases for ELM. iii) Implementation of the proposed MFO-ELM on five machine learning datasets and comparatively evaluating the results. The remaining sections are organized as follows: section 2 is the proposed method, section 3 is the research method, section 4 is the results and discussion, and section 5 is the conclusion.

## 2 THE PROPOSED METHOD

### 2.1. Extreme learning machine

ELM is a single layer feedforward neural (SLFN) network. Assuming an arbitrary  $N$  cases, each instance has  $d$ -dimensional feature and belongs to 1 of  $m$  classes in the set. The dataset can be represented as  $(x_i, y_i)$ ,  $i = 1, 2, \dots, N$ , where  $x_i$  is the input vector  $x_i \in R^n$ , and  $y_i$  is the expected result  $y_i \in R^m$ . For an SLFN network with  $L$  hidden neurons, and  $g(x)$  activation function, the network is to be trained with vectors

of  $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,n}]^T$ ,  $i = 1, 2, \dots, N$ , and the target vector  $Y_i = [y_{i,1}, y_{i,2}, \dots, y_{i,m}]^T$ . The output of the neural network is mathematically modeled in (1).

$$y_i = \sum_{j=1}^L \beta_j g(w_j \cdot x_i + b_j), \quad i \in [1, N] \tag{1}$$

Where  $w_j = [w_{j,1}, w_{j,2}, \dots, w_{j,n}]^T$  is a vector of weights connecting the  $i^{th}$  input node to  $j^{th}$  hidden neuron, and  $b_j$  is the bias of  $j^{th}$  hidden neuron.  $\beta_j = [\beta_{j,1}, \beta_{j,2}, \dots, \beta_{j,m}]^T$  is a vector of output weights between the hidden neurons and the output nodes,  $w_j \cdot x_i$  is the inner product of  $w_j$  and  $x_i$ . In (1) can be compactly presented as in (2) for  $N$  system of equations.

$$Y = H\beta \tag{2}$$

where  $H$ , and  $\beta$  are the hidden neurons output and weight respectively, and  $Y$  is the ELM output. The operating principle of extreme learning machines is based on empirical risk management. ELM is ill-conditioned because the input weights and biases are randomly assigned, therefore it tends to over-fit.

**2.2. Moth-flame optimization technique**

MFO is a population-based bionic optimization algorithm. It regulates the exploration and exploitation during the search process. We consider a search space with  $n$  moths and  $d$  positions. The moth positions are initialized randomly using the (3) within an interval [-1 and 1] as in [27]:

$$X_{ij} = r * (ub_j - lb_j) + lb_j \tag{3}$$

where  $x_{ij}$  is moth  $i^{th}$  in feature location  $j^{th}$  of the search space,  $i = 1, 2, \dots, n, j = 1, 2, \dots, d$ ;  $lb_j$  and  $ub_j$  are lower and upper bounds of moth positions in the space. Each moth represents a candidate solution. The moth positions represent the input weights and biases to be optimized. The position vector of each moth in the search space is regulated by a flag operator. The flag ensures optimal fitness values. Each position vector is passed to the fitness function to calculate the fitness value.

The moths and the flames are solutions in the algorithm search space. The moths represent the particles that move around the flames, however, the flames are a matrix of the best moth positions attained so far. Each flame is assigned to a moth to prevent local optimal stagnation. As the moths search around the flames, they are updated if there is a better solution. The process will continue until it reaches the maximum set iteration for all the moths to attain their best possible solutions. A logarithmic spiral model [22] is used as the main update mechanism of moths. In (4) is the logarithmic spiral for the MFO algorithm.

$$S(M_i, F_j) = D_i e^{at} \text{Cos}(2\pi t) + F_j \tag{4}$$

where  $S$  is a function model that controls the flying of a moth around a flame, which may not necessarily be in the space between them. This is set to regulate the exploration and exploitation of the model.  $M_i$  indicates the  $i^{th}$  moth,  $F_j$  indicates the  $j^{th}$  flame, and  $D_i$  indicates the Euclidian distance of the  $i^{th}$  moth for the  $j^{th}$  flame,  $a$  is a constant for defining the logarithmic spiral's shape,  $t$  parameter-random number in [-1,1] which specifies how close the next moth position should be to the flame.  $2\pi t$  is the distance between successive turns of the spirals.  $a$  is the linear decrease from -1 to -2 throughout the iteration. It determines the convergence of the algorithm. The decrement in the flames attempts to regulate the exploration and exploitation of the search space.

The effectiveness of the algorithm strongly depends on the distance  $D$  between the moths and the flame. The position update of moths relative to  $n$  locations may degrade the exploitation of the candidate solution. Hence, an adaptive model shown in (5) is employed to determine the number of flames.

$$flameNumber = \text{round} \left( N - l \times \frac{N-1}{T} \right) \tag{5}$$

$l$  – the current iteration,  $N$  - the flames count,  $T$  – the maximum iteration. The position update of the moths is carried out with respect to the best flame in the last iteration. After the termination, the best moth is returned as the best optimal approximation.

### 3 METHOD

To improve the classification accuracy of ELM in this study, we use (3) to generate a population of 100 candidate solutions. Then, we employed MFO to optimize the population. The best candidate solution was selected and reshaped into weights and biases. The weights are random numbers in the range  $[-1, 1]$  while the biases are the range of  $[0, 1]$ . These parameters are passed to ELM for classification. We present the conceptual design in Figure 1, the algorithm in Algorithm 1 and the optimization parameters are from [28]-[30].

We run the algorithm on five life datasets. Four (4) of the datasets were drawn from the *UCI* repository: Blood, Breast, Diabetes, and Liver; while the one (1) from the *datahub* repository is the Phoneme dataset. The datasets were normalized to ensure the even distribution of data points and avoid the effect of being skewed towards features with higher values. The datasets were partitioned into training and testing data in ratio 2:1 as shown in the Data pre-processing phase of Figure 1. We constructed ten (10) SLFN at an interval of 5 nodes (range between 5 and 50 nodes). Each simulation reached the best accuracy within the node range. For every SLFN construct, there were thirty (30) trials of simulations, then the average results are computed. The results of particle swarm optimization extreme learning machines (PSO-ELM) and competitive swarm optimization extreme learning machines (CSO-ELM) algorithms are drawn from [31] for performance comparison.

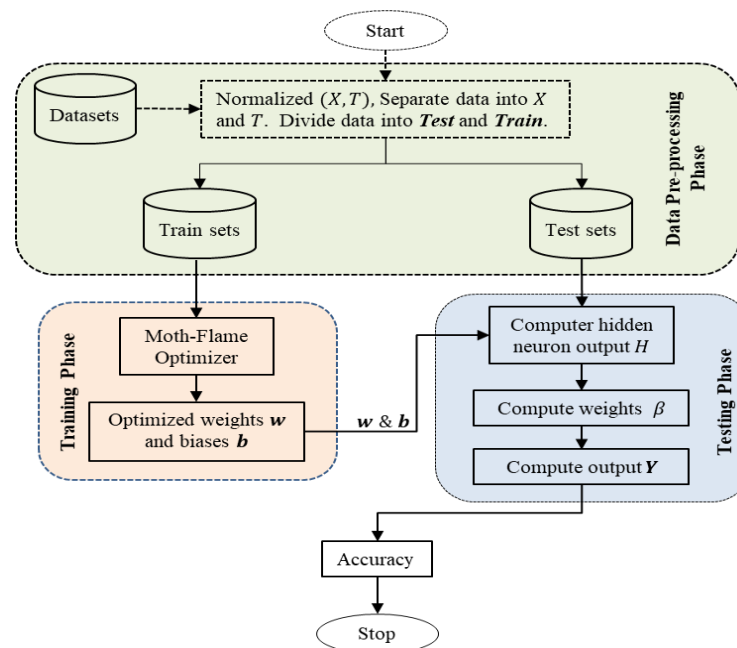


Figure 1. Flowchart of the enhanced MFO-ELM scheme

#### Algorithm 1. Enhanced MFO-ELM algorithm

- 1: **Initialize** MFO parameters
- 2: **Output:** Accuracy
- 3: Generate guess using equation (3) and evaluate the fitness
- 4: **while**  $iter < iterM$  **do**
- 5: Update flame number using equation (5)
- 6: sort moth with fitness values and select the best
- 7: use equation (5) to calculate linear decrement from -1 to -2
- 8:  $D = abs(F_{ij} - X_{ij})$  //  $D$  is the distance between the flame and the corresponding moth
- 9: **for**  $i = 1$  **to**  $N$  **do**
- 10: Update the moth positions with respect to flame
- 11: **end for**
- 12: Select the best flame:  $Iter = iter + 1$
- 13: **end while**
- 14: Reshape the best flame ( $F$ ) into weights and biases ( $w, b$ )
- 15: Calculate hidden neuron output  $H$
- 16: Calculate the output weight  $\beta$
- 17: Calculate ELM output with equation (2)
- 28: Calculate the misclassification and the accuracy

### 4 RESULTS AND DISCUSSION

We compared the improved accuracy of the proposed MFO-ELM with CSO-ELM, PSO-ELM, and ELM in this section for each dataset as shown in Figure 2. The comparative performance for each model in each dataset is summarized in Figure 3. Also, we extend our discussion to include the rate of improvement of the proposed algorithm over these algorithms and analyze the significant improvement test of the proposed algorithm on classical ELM.

#### 4.1. Evaluation of accuracy

We evaluated the classification performance Figure 2 of MFO-ELM with CSO-ELM, PSO-ELM, and ELM in Figures 2(a)-(e) for each dataset. We presented the relative performance of the algorithms in Figure 3. The relative performance is the average classification accuracy of each algorithm in the simulations. This is presented in a single chart for easy comparison. The results show that the proposed MFO-ELM performed better in Blood, Breast, Diabetes, and Liver datasets which represents 80% of the simulations. It is only in the simulation on Phoneme datasets that CSO-ELM and PSO-ELM performed better than MFO-ELM.

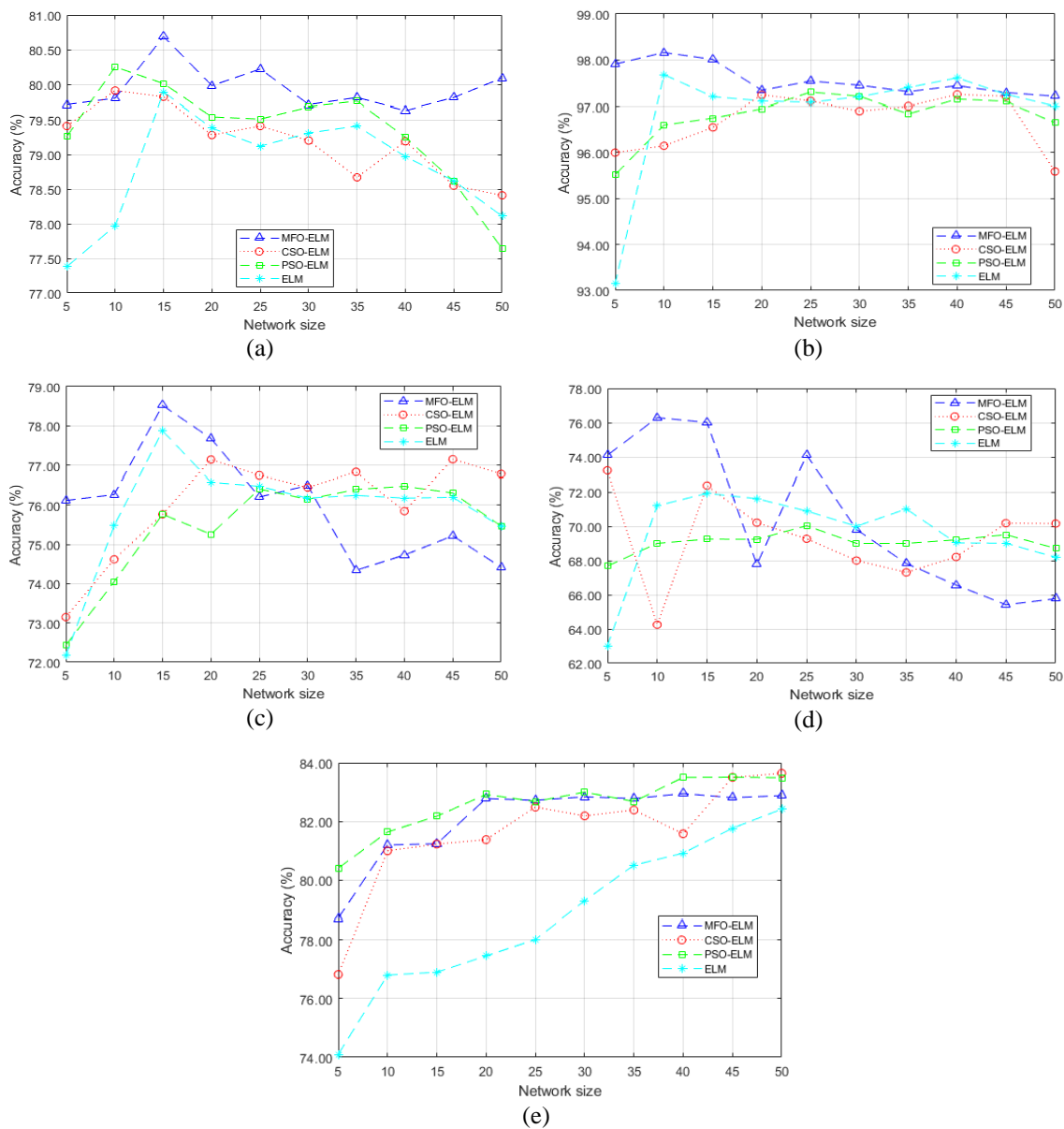


Figure 2. Compared the improved accuracy of the MFO-ELM: (a) performance comparison of classification on blood dataset, (b) performance comparison of classification on breast dataset, (c) performance comparison of classification on diabetes dataset, (d) performance comparison of classification on liver dataset, and (e) performance comparison of classification on phoneme dataset

The overall evaluation of the comparison shows that the enhancement of ELM with MFO improves its performance. The mean accuracies of ELM are improved on all the simulations on the five datasets. That is the accuracies of MFO-ELM is better than ELM in all. Therefore, optimizing the input weights and biases with MFO improved the accuracy of the ELM algorithm.

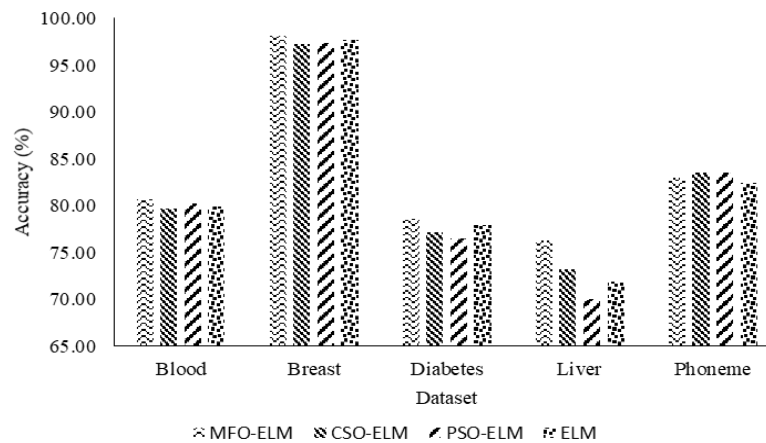


Figure 3. Relative classification accuracy on each dataset

#### 4.2. Percentage performance improvement rate (PIR%)

More so, the percentage performance improvement rate (PIR%) evaluates the efficiency of the MFO-ELM on ELM and compared it with PSO-ELM and CSO-ELM algorithms. The higher the PIR% the better the improvement on the ELM algorithm. PIR% is expressed in (6):

$$PIR\% = \frac{A_s - B_s}{B_s} * 100 \quad (6)$$

where  $A_s$  and  $B_s$  are the scores of two comparative algorithms,  $A_s$  is the proposed algorithm while  $B_s$  is the benchmark algorithm.

Table 1 shows the PIR% of classification accuracy of the MFO-ELM scheme and the other two algorithms (PSO-ELM and CSO-ELM) on ELM for the five selected datasets. On the Blood dataset, the PIR% gained by MFO-ELM over ELM is 1.0014, PSO-ELM is 0.4381, while CSO-ELM scaled negatively with -0.2754. This shows that MFO better enhanced the performance of ELM than the other two meta-heuristic algorithms. For simulation on the Breast cancer dataset, the PIR% of MFO-ELM on ELM is 0.4914, while PSO-ELM and CSO-ELM had negative scores of -0.3890 and 0.4197 respectively. Therefore, MFO-ELM has a superior improvement rate on ELM and the other meta-heuristic algorithms did not.

Table 1. Summary of PIR% of each meta-heuristic algorithm on ELM

	ELM	PSO-ELM		CSO-ELM		MFO-ELM	
	Accuracy	Accuracy	PIR%	Accuracy	PIR%	Accuracy	PIR%
Blood	79.89	80.24	0.4381	76.67	-0.2754	80.69	1.0014
Breast	97.68	97.30	-0.3890	97.27	-0.4197	98.16	0.4914
Diabetes	77.87	76.48	-1.7850	77.15	-0.9246	83.46	0.8647
Liver	71.92	70.03	-2.6279	73.22	1.8076	82.95	6.0762
Phoneme	82.42	83.49	1.2982	83.46	1.2618	82.95	0.6431

Similarly, on the Diabetes dataset, the PIR% of MFO-ELM on ELM is 0.8647, while for PSO-ELM, CSO-ELM the rates are negative with -1.7805 and -0.9246 respectively. MFO-ELM proves its superiority over all algorithms. Also, CSO-ELM showed a better improvement rate than PSO-ELM with respective scores of 1.8076 and -2.6279 respectively. However, MFO-ELM has a substantial higher improvement rate of 6.0762. Contrary to the results of simulations on the previous datasets, the improvement rate of MFO-ELM on the Phoneme dataset is less than PSO-ELM and CSO-ELM. While MFO-ELM has an improvement rate of 0.6431, PSO-ELM and CSO-ELM have improvement rates of 1.2982 and 1.2618 respectively. PSO-ELM shows a higher improvement rate than any other algorithm. This follows the theorem of “No free lunch for all” [32].

The accuracy and PIR% justify that the enhanced MFO-ELM improves the accuracy of ELM classification than the other meta-heuristic optimization algorithms. This is because MFO has a better exploration and exploitation process of the search space than the other two meta-heuristic algorithms, which eventually led to the selection of optimal parameters for ELM.

**4.3. Test of significant improvement of MFO-ELM on ELM classification**

We measured the significant improvement of the MFO-ELM algorithm on the ELM algorithm with the Wilcoxon signed ranked test [31], [33]. Table 2 presents the analysis of the test. Mean is the average classification accuracies of the two comparison algorithms, std dev. is the standard deviation, min and max are the respective minimum and maximum accuracies. The Wilcoxon signed rank test column shows rank (negative (Neg.) or positive (Pos.)), the number of samples N, mean rank M and p-value is the 2-tail significant values. Rank shows the rankings of the algorithms based on the key below the table. The positive rank is the total number of accuracies of MFO-ELM that are better than the ELM’s in the simulations, while the negative rank is that of ELM that is better than MFO-ELM’s accuracies, and Ties is the number of equal accuracies of the algorithms. The better result of the two comparisons for statistical and Wilcoxon rank test measures are bold in the table.

For the blood dataset, the average accuracies of MFO-ELM ranked higher than the ELM algorithm in all the statistical and Wilcoxon rank test measures. We obtained similar results for Breast and Phoneme dataset simulations. Also, the p-values of the simulations of these three datasets are less than 0.05 which shows a significant improvement of MFO-ELM on the standard ELM. The statistical measures for MFO-ELM are higher than the ELM algorithm on the Diabetes dataset’s simulations. However, the Wilcoxon signed rank shows that the improvement measure is not significant. Similar to Diabetes simulations, the results for the Liver dataset for ELM have better standard deviations than MFO-ELM in the statistics columns, The Wilcoxon signed rank test also favours ELM more positively than MFO-ELM, and the p-Value shows no significant improvement.

The overall evaluation of the comparison shows that MFO improves the ELM performance. Statistically, the mean accuracies are improved on all the simulations on the five datasets. That is the accuracies of MFO-ELM is better than ELM in all. Also, the standard deviations of statistics of MFO-ELM are less than ELM in four (4) out of the five (5) datasets, which means the algorithm is more stable in 80% of the whole simulations. More so, with the Wilcoxon Signed rank test, MFO-ELM ranked higher than ELM in three (3), and equal in one, but lost out in only one. This shows a 70% improvement ranking. The p-values also prove a significant improvement in 60% of the simulations. Therefore, optimizing the input weights and biases with MFO significantly improved the stability and accuracy of the ELM algorithm.

Table 2. Wilcoxon signed rank test for MFO-ELM and ELM classification accuracy

Dataset	Algorithm	Statistics				Wilcoxon signed rank test: MFO-ELM/ELM			
		Mean	Std Dev	Min.	Max	Rank	N	M	p-Value
Blood	ELM	78.810	0.007	77.380	79.890	Neg.	0	0.00	<b>0.005</b>
	MFO-ELM	<b>79.940</b>	<b>0.003</b>	<b>79.630</b>	<b>80.690</b>	Pos.	<b>10</b>	<b>5.50</b>	
	Ties	0							
Breast	ELM	96.860	0.013	93.150	97.650	Neg.	2	2.50	<b>0.022</b>
	MFO-ELM	<b>97.560</b>	<b>0.003</b>	<b>97.210</b>	<b>98.160</b>	Pos.	<b>8</b>	<b>6.25</b>	
	Ties	0							
Diabetes	ELM	75.880	0.015	72.170	77.870	Neg.	<u>5</u>	<b>5.80</b>	0.878
	MFO-ELM	<b>76.000</b>	<b>0.014</b>	<b>74.340</b>	<b>78.520</b>	Pos.	<u>5</u>	5.20	
	Ties	0							
Liver	ELM	69.580	<b>0.026</b>	63.010	71.920	Neg.	<b>6</b>	3.83	0.646
	MFO-ELM	<b>70.030</b>	0.043	<b>65.420</b>	<b>76.290</b>	Pos.	4	<b>8.00</b>	
	Ties	0							
Phoneme	ELM	78.810	0.026	74.100	82.420	Neg.	0	0.00	<b>0.005</b>
	MFO-ELM	<b>82.090</b>	<b>0.014</b>	<b>78.710</b>	<b>82.950</b>	Pos.	<b>10</b>	<b>5.50</b>	
	Ties	0							

**Note:** The bold values indicate the best results  
 Neg. Ranks: MFO-ELM < ELM  
 Pos. Ranks: MFO-ELM > ELM  
 Ties: MFO-ELM = ELM

**5. CONCLUSION**

This study proposed an enhanced MFO-ELM algorithm. The proposed algorithm used MFO to set the initial value of input weights and hidden neuron biases for the ELM classifier. It was applied to the classification of some medical datasets. The overall performance shows that the proposed MFO-ELM

algorithm improved the basic ELM algorithm in all five datasets. That is, MFO-ELM has improved the performance of ELM 100% in the simulations. This proves that the initial setting of the input weights and biases of ELM by the MFO optimization scheme enhances the performance of the ELM classifier. More so, when comparing MFO-ELM with other ELM enhanced optimization algorithms, it is only in the Phoneme dataset that PSO-ELM and CSO-ELM performed better than the proposed algorithm. It is superior to the other meta-heuristic algorithms in 80% of the simulations. Further study will be focused on the hybridization of two meta-heuristic algorithms to improve the parameter setting of ELM.

## ACKNOWLEDGEMENT

The authors wish to thank the Ministry of Higher Education, Malaysia, for funding this study under the Fundamental Research Grant Scheme FRGS/1/2019/ICT02/UTM/02/13. The authors would like to thank Research Management Centre (RMC) Universiti Teknologi Malaysia, and ALI@S for their support in research activities.

## REFERENCES




- [1] X. N. Du, H. Xu, and F. Zhu, "Understanding the effect of hyperparameter optimization on machine learning models for structure design problems," vol. 135, 2021, *arXiv: 103013*, doi: 10.1016/j.cad.2021.103013.
- [2] H. Guo, X. Zhuang, and T. Rabczuk, "A deep collocation method for the bending analysis of Kirchhoff plate," *Comput. Mater. Contin.*, vol. 59, no. 2, pp. 433–456, 2021, doi: 10.32604/cmc.2019.06660.
- [3] N. A. Almansour *et al.*, "Neural network and support vector machine for the prediction of chronic kidney disease: A comparative study," *Comput. Biol. Med.*, vol. 109, no. October 2018, pp. 101–111, 2019, doi: 10.1016/j.combiomed.2019.04.017.
- [4] T. Wang, D. Zhou, and Q. Gu, "Provably efficient reinforcement learning with linear function approximation under adaptivity constraints," *Advances in Neural Information Processing Systems*, pp. 1–17, 2021.
- [5] G. Bin Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," *IEEE Int. Conf. Neural Networks - Conf. Proc.*, vol. 2, 2004, pp. 985–990, doi: 10.1109/IJCNN.2004.1380068.
- [6] M. Eshtay, H. Faris, and N. Obeid, "A competitive swarm optimizer with hybrid encoding for simultaneously optimizing the weights and structure of extreme learning machines for classification problems," *International Journal of Machine Learning and Cybernetics.*, vol. 11, no. 8, pp. 1801–1823, 2020, doi: 10.1007/s13042-020-01073-y.
- [7] O. A. Alade, A. Selamat, and R. Sallehuddin, "A review of advances in extreme learning machine techniques and its applications," in *Recent Trends in Information and Communication Technology*, pp. 885–895, 2018.
- [8] Z. Zhou, J. Chen, and Z. Zhu, "Regularization incremental extreme learning machine with random reduced kernel for regression," *Neurocomputing*, vol. 321, pp. 72–81, 2018, doi: 10.1016/j.neucom.2018.08.082.
- [9] C. M. C. Baron and J. Zhang, "Reliable on-line re-optimization control of a fed-batch fermentation process using bootstrap aggregated extreme learning machine," in *Lecture Notes in Electrical Engineering*, vol. 495, pp. 272–294, 2020, doi: 10.1007/978-3-030-11292-9\_14.
- [10] F. K. Inaba, T. E. O. Salles, S. Perron, and G. Caporossi, "DGR-ELM—distributed generalized regularized ELM for classification," *Neurocomputing*, vol. 275, pp. 1522–1530, 2018, doi: 10.1016/j.neucom.2017.09.090.
- [11] H. M. Ni, Y. J. Liu, and Y. C. Fan, "Optimization of injection scheme to maximizing cumulative oil steam ratio based on improved artificial bee colony algorithm," *Journal of Petroleum Science and Engineering*, vol. 173, no. May 2018, pp. 371–380, 2019, doi: 10.1016/j.petrol.2018.10.032.
- [12] W. Deng, Q. Zheng, and L. Chen, "Regularized extreme learning machine," *2009 IEEE Symp. Comput. Intell. Data Mining, CIDM 2009 - Proc.*, no. 60825202, 2009, pp. 389–395, doi: 10.1109/CIDM.2009.4938676.
- [13] J. M. M.-Martínez, P. E.-Montero, E. S.-Olivas, J. D. M. Guerrero, R. M. Benedito, and J. G. Sanchis, "Regularized extreme learning machine for regression problems," *Neurocomputing*, vol. 74, no. 17, pp. 3716–3721, 2011, doi: 10.1016/j.neucom.2011.06.013.
- [14] L. I. U. Jingsen, L. I. U. Li, and L. I. Yu, "A differential evolution flower pollination algorithm with dynamic switch probability \*," *Chinese Journal of Electronics*, vol. 28, no. 4, pp. 737–747, 2019, doi: 10.1049/cje.2019.04.008.
- [15] M. A. basset, R. Mohamed, R. K. Chakraborty, M. J. Ryan, and S. Mirjalili, "Computers & industrial engineering an efficient binary slime mould algorithm integrated with a novel attacking-feeding strategy for feature selection," *Comput. Ind. Eng.*, vol. 153, p. 107078, 2021, doi: 10.1016/j.cie.2020.107078.
- [16] M. Maimaitiyiming, V. Sagan, P. Sidike, and M. T. Kwasniewski, "Dual activation function-based extreme learning machine (ELM) for estimating grapevine berry yield and quality," *Remote Sens.*, vol. 11, no. 7, 2019, doi: 10.3390/rs11070740.
- [17] S. Sengupta, S. Basak, R. Alan, and P. Ii, "Particle swarm optimization : a survey of historical and recent developments with hybridization perspectives," *Machine Learning and Knowledge Extraction*, vol. 1, no. 1, pp. 157–191, 2018, doi: 10.3390/make1010010.
- [18] C. Gambella, B. Ghaddar, and J. N. Sawaya, "Optimization problems for machine learning: A survey," *European Journal of Operational Research*, vol. 290, no. 3, pp. 807–828, 2021, doi: 10.1016/j.ejor.2020.08.045.
- [19] R. A. I. Alhayali, M. Aljanabi, A. H. Ali, M. A. Mohammed, and T. Sutikno, "Optimized machine learning algorithm for intrusion detection," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, no. 1, pp. 590–599, 2021, doi: 10.11591/ijeecs.v24.i1.pp590-599.
- [20] Y. Yang and Z. Duan, "An effective co-evolutionary algorithm based on artificial bee colony and differential evolution for time series predicting optimization," *Complex & Intelligent Systems*, vol. 6, no. 2, pp. 299–308, 2020, doi: 10.1007/s40747-020-00149-0.
- [21] G. Li, P. Liu, C. Le, and B. Zhou, "A novel hybrid meta-heuristic algorithm based on the cross-entropy method and firefly algorithm for global optimization," *Entropy*, vol. 21, no. 5, 2019, doi: 10.3390/e21050494.
- [22] M. Shehab, L. Abualigah, H. Al Hamad, H. Alabool, M. Alshinwan, and A. M. Khasawneh, "Moth–flame optimization algorithm: variants and applications," *Neural Computing and Applications*, vol. 6, 2019, doi: 10.1007/s00521-019-04570-6.
- [23] R. Xie, I. Khalil, S. Badsha, and M. Atiquzzaman, "Collaborative extreme learning machine with a confidence interval for P2P learning in healthcare," *Comput. Networks*, vol. 149, pp. 127–143, 2019, doi: 10.1016/j.comnet.2018.11.002.






- [24] D. Gabi, A. S. Ismail, A. Zainal, Z. Zakaria, and A. Abraham, "Orthogonal taguchi-based cat algorithm for solving task scheduling problem in cloud computing," *Neural Computing and Applications*, vol. 30, no. 6, pp. 1845–1863, 2018, doi: 10.1007/s00521-016-2816-4.
- [25] W. Deng, J. Xu, Y. Song, and H. Zhao, "Differential evolution algorithm with wavelet basis function and optimal mutation strategy for complex optimization problem," *Applied Soft Computing*, vol. 100, p. 106724, 2021, doi: 10.1016/j.asoc.2020.106724.
- [26] M. Wang and H. Chen, "Chaotic multi-swarm whale optimizer boosted support vector machine for medical diagnosis," *A Applied Soft Computing*, vol. 88, p. 105946, 2020, doi: 10.1016/j.asoc.2019.105946.
- [27] Y. Xu *et al.*, "An efficient chaotic mutative moth-flame-inspired optimizer for global optimization tasks," *Expert Systems with Applications*, vol. 129, no. 2019, pp. 135–155, 2019, doi: 10.1016/j.eswa.2019.03.043.
- [28] G. I. Sayed, A. Darwish, and A. E. Hassanien, "Binary whale optimization algorithm and binary moth flame optimization with clustering algorithms for clinical breast cancer diagnoses," *Journal Classif.*, 2019, doi: 10.1007/s00357-018-9297-3.
- [29] R. Cheng and Y. Jin, "A competitive swarm optimizer for large scale optimization," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 191–204, 2015, doi: 10.1109/TCYB.2014.2322602.
- [30] H. Lu, B. Du, J. Liu, H. Xia, and W. K. Yeap, "A kernel extreme learning machine algorithm based on improved particle swarm optimization," *Memetic Computer*, vol. 9, no. 2, pp. 121–128, 2017, doi: 10.1007/s12293-016-0182-5.
- [31] M. Eshtay, H. Faris, and N. Obeid, "Improving extreme learning machine by competitive swarm optimization and its application for medical diagnosis problems," *Expert Systems with Applications*, vol. 104, pp. 134–152, 2018, doi: 10.1016/j.eswa.2018.03.024.
- [32] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Computer*, vol. 1, no. 1, pp. 67–82, 1997, doi: 10.1109/4235.585893.
- [33] S. Anuar, A. Selamat, R. Sallehuddin, "A modified scout bee for artificial bee colony algorithm and its performance on optimization problems," *Journal of King Saud University - Computer and Information Sciences*, Vo. 28, no. 4, pp. 395–406, 2016.

## BIOGRAPHIES OF AUTHORS






**Oyekale Abel Alade**    holds a Ph.D. in Computer Science from Universiti Teknologi Malaysia. He has his Higher National Diploma, Post Graduate Diploma, and Masters of Technology in Computer Science. He is a lecturer in the Federal Polytechnic, Bida, Niger State Nigeria. He has been a Production Editor of the Polymath Journal. He is a reviewer of some international journal publications. His main research interests include data science, machine learning, optimization, and digital library. He can be contacted at email: kaleabel@gmail.com.



**Associate Professor Dr. Roselina Sallehuddin**    is a Ph.D. holder in Computer Science from Universiti Teknologi Malaysia (UTM). She has more than 25 years of experience in teaching and research. Her research interests are Artificial Intelligence, Data Mining, Combinatorial Optimization, Time Series Forecasting, and Operation Research. Her H-index is 14 with 473 citations. She can be contacted at email: roselina@utm.my.



**Nor Haizan Mohamed Radzi**    is currently a Senior Lecturer in the School of Computing, Universiti Teknologi Malaysia. She is graduated with Ph.D. from the University of Strathclyde United Kingdom in 2011. Her research interest is in the field of artificial intelligence, machine learning, and optimization. She can be contacted at email: haizan@utm.my.