İSTANBUL UNIVERSITY
CERRAHPAŞA

ORIGINAL ARTICLE
Electrica 2023; 23(3): 597-606

Performance Evaluation of PDO Algorithm through Benchmark Functions and MLP Training

Erdal Eker[1], Murat Kayri[2], Serdar Ekinci[3,5], Mehmet Ali Kaçmaz[4]

[1]Vocational School of Social Sciences, Mus Alparslan University, Mus, Turkey
[2]Department of Computer and Instructional Technology Education, Van Yüzüncü Yıl University, Van, Turkey
[3]Department of Computer Engineering, Batman University, Batman, Turkey
[4]Hacı Rahime Ulusoy Maritime Vocational High School, Turkey
[5]MEU Research Unit, Middle east University, Amman, Jordan

Cite this article as: E. Eker, M. Kayri, S. Ekinci and M. Ali Kaçmaz, "Performance evaluation of PDO algorithm through benchmark functions and MLP training," *Electrica*, 23(3), 597-606, 2023.

ABSTRACT

Metaheuristic algorithms have become very common in the last two decades. The flexibility and ability to overcome obstacles in solving global problems have increased the use of metaheuristic algorithms. In the training of multilayer perceptron (MLP), metaheuristic algorithms have been preferred for many years due to their good classification capabilities and low error values. Therefore, this study evaluates the performance of the Prairie dog optimization (PDO) algorithm for MLP training. In this context, there are two main focuses in this study. The first one is to test the performance of the PDO algorithm through test functions and to compare it with different metaheuristic algorithms for demonstration of its superiority, and the second is to train MLP using the IRIS dataset with the PDO algorithm. As the PDO is one of the most recent metaheuristic algorithms, the lack of any study on this subject is the motivation for the article. PDO algorithm can be used in real-world problems as a powerful optimizer, as it reaches the minimum point in functions, and can also be used as a classification algorithm because it has successfully performed in MLP training.

Index Terms—Classification, metaheuristics, multilayer perceptron, prairie dog optimization

I. INTRODUCTION

Optimization can be regarded as the effort to obtain the solution for a problem with the minimum cost. Therefore, optimization algorithms have become quite popular and are being used in almost all real-world problems [1]. The simplicity of the solution and the effort to reach the optimum values in optimization algorithms have led to the development of new algorithms [2]. In terms of artificial neural network (ANN) training [3], the optimization algorithms can be classified under two main groups, namely gradient [4] and metaheuristics [5]. Although the gradient method is widely used, it has significant disadvantages such as early convergence, high dependence on initial parameters, and stagnating in local optima. These disadvantages can prevent the gradient methods from reaching the global optimum [6]. On the other hand, metaheuristic algorithms perform the optimization process with a randomly generated initial set, which not only facilitates the solution but also provides solution diversity. The exploration and exploitation stages facilitate reaching the optimum values [7]. As stated by "no free lunch theorem," different metaheuristic algorithms are developed to solve different difficult problems more effectively [8]. Metaheuristic algorithms can avoid local optima regardless of the structure of the search space, get rid of excessive dependency at any point by using an initial set, and prevent premature convergence by combining local and global search features [9]. Considering the capability of the metaheuristic algorithms, this paper adopts the prairie dog optimization (PDO) algorithm as a new algorithm for MLP training. Firstly, the problem-solving ability of the PDO algorithm is demonstrated with experimental findings, and then it is used as a supervise in the classifier ANN training [10].

ANNs are one of the popular subjects that solve problems by simulating the neural system in the human brain [11]. Multilayer perceptron (MLP), among forward feedback architecture (FNN), is the most important ANN type due to its robust structure in the classification of nonlinear,

Corresponding author:
Erdal Eker

E-mail:
e.eker@alparslan.edu.tr

Received: October 17, 2022

Accepted: May 8, 2023

Publication Date: June 14, 2023

DOI: 10.5152/electr.2023.22179

complex, and noisy data [12–14]. MLP usually consists of at least three layers, and a hidden layer is placed between the input and output layers [15]. The MLP training process is to find the correct values of the weights that produce the desired output to classify the features correctly. The mean squared error (MSE) value is obtained because of calculating the difference between the output obtained because of training the MLP architecture and the desired output of the dataset [16]

$$MSE = \sum_{m=1}^{n} \frac{\sum_{i=1}^{k} \left(O_i(m) - D_i(m)\right)^2}{n} \quad (1)$$

In the equation, k is the number of outputs, m is the number of features of the trained data set, $D_i(m)$ is the desired output for each feature is the output $O_i(m)$ obtained. Because of its success, MLP has been utilized in many real-world problems [17]. Metaheuristic algorithms have gotten popularity in MLP training in recent years [18]. No previous attempts have been reported for PDO [19] algorithm-based MLP training which makes this work unique in this regard. Therefore, in this paper, the advantages of the PDO algorithm in MLP training is revealed.

The paper structure is arranged as follows. In the second section, the properties of the PDO algorithm are given. In the third section, the MLP training process is explained. In the fourth section, the CEC 2019 test suite, which is relatively more difficult, complex, and up-to-date, compared to CEC 2017 [20] test suite, was used to compare the PDO algorithm with other algorithms. The related test suite consists of ten functions [21], and details of all functions are given in Table I. Testing benchmark functions and updating the winning algorithms each year play an important role in making comparisons. Thus, the winners [22–24] of the CEC2017 unconstrained optimization competition, winners of CEC 2018 [25], Winners of CEC 2020 [26], and recent algorithm [27, 28] may be regardable.

The fifth section concludes the paper.

TABLE I. DETAILS OF THE CEC 2019 BENCHMARK FUNCTIONS

No	Name of Functions	Dimension	Search Range	F_{min}
F1	Storn's Chebyshev polynomial fitting problem	9	[−8192, 8192]	1
F2	Inverse Hilbert matrix problem	16	[−16384, 16384]	1
F3	Lennard–Jones minimum energy cluster	18	[−4, 4]	1
F4	Rastrigin's function	10	[−100, 100]	1
F5	Griewangk's function	10	[−100, 100]	1
F6	Weierstrass function	10	[−100, 100]	1
F7	Modified Schwefel's function	10	[−100, 100]	1
F8	Expanded Schaffer's F6 function	10	[−100, 100]	1
F9	Happy cat function	10	[−100, 100]	1
F10	Ackley function	10	[−100, 100]	1

II. PRAIRIE DOG OPTIMIZATION (PDO) ALGORITHM

The movements of a type of rodent called Prairie dogs (PDs) living in America are the main inspiration of the PDO algorithm [29]. The proposed algorithm mimics four behaviors of the Prairie dog for performing exploration and exploitation stages. The foraging and nest-building activities of the PDs are used to provide exploratory behavior for the PDO. PDs build their nests around an abundant food source. They seek a new food source and build new nests around it when the food source is depleted, which inspires the exploration phase of the PDO. On the other hand, their specific response to two unique communications or beeps is mimicked for the exploitation of the PDO. PD has signals or sounds for different scenarios, from predatory threats to food availability. Their communication skills play an important role in meeting their nutritional needs and their ability to protect against predators. These two specific behaviors lead the PDs to get closer to a particular or promising place. PDO algorithm consists of initialization, fitness function evaluation, exploration, and exploitation phases. The details are provided in the following subsections.

A. Initialization

In a group of PDs, n individuals are found and PD belongs to m groups. The position of the *i*th Prairie dog in a particular group can be determined by a vector. The following matrix refers to the position of all groups in a colony [19]:

$$CT = \begin{bmatrix} CT_{1,1} & CT_{1,2} & \cdots & CT_{1,d} \\ CT_{2,1} & CT_{2,2} & \cdots & CT_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ CT_{m,1} & CT_{m,2} & \cdots & CT_{m,d} \end{bmatrix} \quad (2)$$

where $CT_{i,j}$ ($n \le m$) represents the *j*th size of the *ith* group in a colony. The following definition expresses the position of all PDs in a group where $PD_{i,j}$ ($n \le m$) represents the size of *jth* PD in the *ith* group.

$$PD = \begin{bmatrix} PD_{1,1} & PD_{1,2} & \cdots & PD_{1,d} \\ PD_{2,1} & PD_{2,2} & \cdots & PD_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ PD_{n,1} & PD_{n,2} & \cdots & PD_{n,d} \end{bmatrix} \quad (3)$$

Each group and PDs are represented by a single distribution as shown in (4) and (5).

$$CT_{i,j} = U(0,1) \times (UB_j - LB_j) + LB_j \quad (4)$$

$$PD_{i,j} = U(0,1) \times (ub_j - lb_j) + lb_j \quad (5)$$

While UB_j and LB_j expresses the lower and upper bounds of the *j*-dimensional optimization problem, $ub_j = \frac{UB_j}{m}$, $lb_j = \frac{LB_j}{m}$ and $U(0,1)$ is a random number with a uniform distribution between 0 and 1.

B. Evaluation of the Fitness Function

The value of each PD's fitness function represents the quality of the available food at a particular resource, its potential to build new nests, and the correct response to predator-prevention stimuli. The array storing the fitness function values is sorted, and the minimum fitness obtained value is declared the best solution for the given minimization problem. The next three best values are considered along with the best value for nest building that helps them escape from predators [19].

$$f(PD)=\begin{bmatrix} f_1[PD_{1,1} & \cdots & PD_{1,d}] \\ \vdots & \ddots & \vdots \\ f_n[PD_{n,1} & \cdots & PD_{n,d}] \end{bmatrix} \tag{6}$$

C. Exploration Phase

This phase includes the PD's foraging and nest-building activities. In PDO, the transitions between exploration and exploitation are divided into four phases. These stages are achieved by dividing the maximum iteration by 4. The first two strategies are used in the exploration phase ($iter < \frac{max\_iter}{4}$ and $\frac{max\_iter}{4} \le iter < \frac{max\_iter}{2}$) and the last two strategies are enforced by exploitation ($\frac{max\_iter}{2} \le iter < 3\frac{max\_iter}{4}$ and $3\frac{max\_iter}{4} \le iter \le max\_iter$).

In the exploration phase, the movements of the PDs are captured with the Lévy flight model so that the concentration in only one area is prevented. The found resource is reported to the group with a special voice, and the decision is made according to the quality of the food. The location update is defined in (7)

$$PD_{i+1,j+1} = GBest_{i,j} - eCBest_{i,j} \times \rho - CPD_{i,j} \times Levy(n), \quad \forall\, iter < \frac{max\_iter}{4} \tag{7}$$

$$PD_{i+1,j+1} = GBest_{i,j} \times rPD \times DS \times Levy(n), \quad \forall \frac{max\_iter}{4} \le iter < \frac{max\_iter}{2} \tag{8}$$

where $GBest_{i,j}$ is the achieved best global value, $eCBest_{i,j}$ is the effect of the obtained best available solution, which is expressed in (9), ρ is fixed custom food source alarm (set to 0.1 kHz), rPD is the position of a generated random solution, rPD refers to the total randomized effect of all PDs in the colony which is expressed by (10)

$$eCBest_{i,j} = GBest_{i,j} \times \Delta + \frac{PD_{i,j} \times mean(PD_{n,m})}{GBest_{i,j} \times (UB_j - LB_j) + \Delta} \tag{9}$$

$$CPD_{i,j} = \frac{GBest_{i,j} - rPD_{i,j}}{GBest_{i,j} + \Delta} \tag{10}$$

The term DS, expressed in (11), refers to the digging power depending on the food source. $Levy(n)$ is a Lévy distribution and contributes to a better and more efficient exploration of the problem space.

$$DS = 1.5 \times r \times \left(1 - \frac{iter}{max\_iter}\right)^{\left(2\frac{iter}{max\_iter}\right)} \tag{11}$$

In the given equations, r takes values between −1 and 1 depending on the existing iterations, providing the probabilistic feature for exploration, Δ represents a small number that explains the small differences that exist in PDs.

D. Exploitation Phase

The PD's response to two separate sounds (communication and warning) constitutes the exploitation phase. Their two different responses to the sounds of feeding and protection from predators result in a promising convergence of the PDO in the exploitation phase. At this stage, strategies Δ and $3\frac{max\_iter}{4} \le iter \le max\_iter$ are followed, respectively. These two strategies are expressed in (12) and (13)

$$PD_{i+1,j+1} = GBest_{i,j} - eCBest_{i,j} \times \varepsilon - CPD_{i,j} \times rand \tag{12}$$

$$PD_{i+1,j+1} = GBest_{i,j} \times PE \times rand \tag{13}$$

where ε refers to the quality of the food source, $PD_{i+1,j+1} = GBest_{i,j} \times PE \times rand$ is the cumulative effect of PDs in the colony, and PE is the effect of predators formulated in (14) [19]

$$PE = 1.5 \times \left(1 - \frac{iter}{max\_iter}\right)^{\left(2\frac{iter}{max\_iter}\right)} \tag{14}$$

III. PDO-BASED MLP TRAINING

The main processing unit of the MLP is neurons, which are similar to neurons in the human brain. Neurons are connected to each other by weights. MLP has three layers, and since the number of neurons in the hidden layer can be determined by the user to achieve the best result, it was determined as $2n+1$ (n, number of neurons in the input layer) for this study. The data presented to the input layer is multiplied by certain weights and transmitted to the hidden layer. The neurons in the hidden layer multiply the data again with certain weights and are collected with the bias value and transmitted to the output layer. The neurons in the output layer also process the data in the same way as the input and hidden layers. Then the activation function is applied to the value. The reason for using the sigmoid function as an activation function is that this function is nonlinear and continuous, and it does not need a derivative. The result obtained from the activation function is the output of the neuron [30, 31]. The IRIS data set was used in this study for training of the ANN [32]. Figure 1 represents the overall process of MLP training via PDO algorithm.

IV. EXPERIMENT AND RESULTS

A. Optimization of Problems with PDO Algorithm

The optimization process of a problem is given in Fig. 2.

The performance of the PDO algorithm was assessed by comparing it with the golden eagle optimization (GEO) [34], reptile search algorithm (RSA) [35], and Archimedes optimization algorithm (AOA) [36]. Experiments were carried out using the MATLAB program. Thirty search agents were used for each algorithm. In addition, 51 independent runs and 500 iterations were used to achieve a fair result. When Table II is examined, statistical measurements of all algorithms in the form of mean, best, worst, and standard deviation (Std) are given. Optimal results are shown in bold. The PDO algorithm achieved the most optimal values in all measurements for the F1, F2, F3, and F6 functions. In F5, it provided the most optimal result in all measurements except the best value. It reached the most optimal result in the best value for the F7 function. Other algorithms achieved relatively better results in F8, F9, and F10 functions.

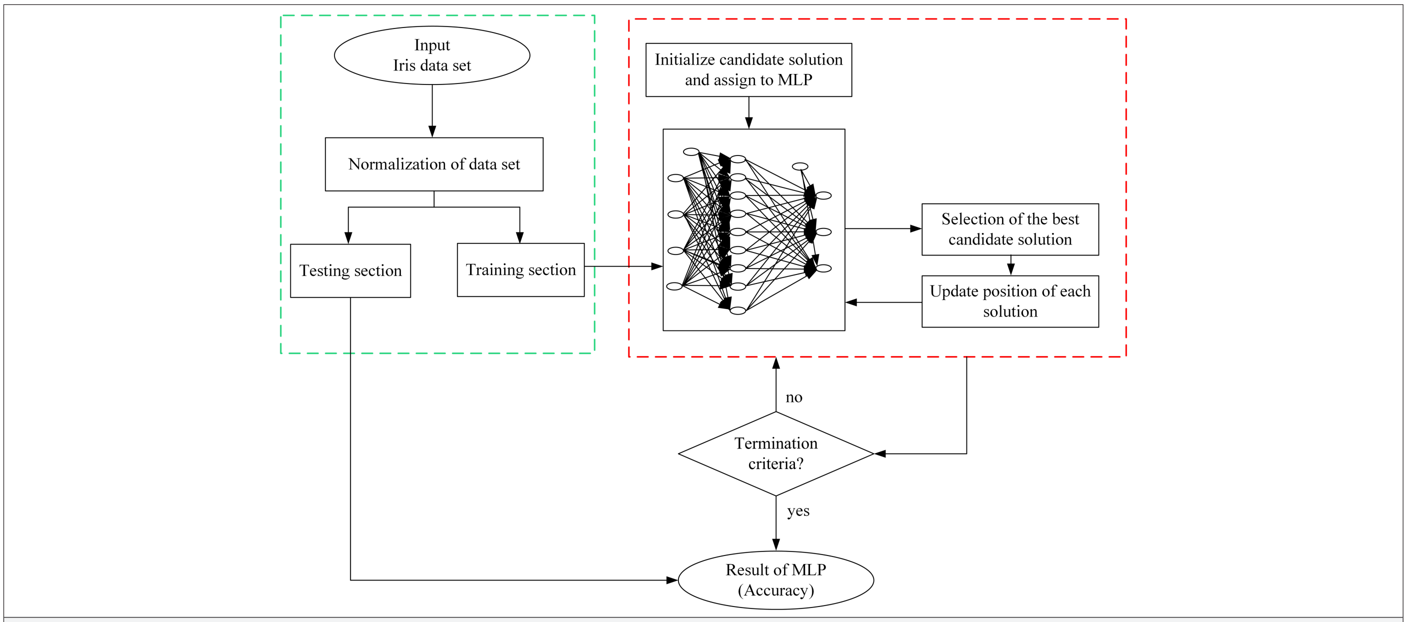

Fig. 1. PDO-MLP flowchart [33]. MLP, multilayer perceptron; PDO, Prairie dog optimization.

Convergence curves are visualized in Fig. 3. When all the results are evaluated, and it is seen that the PDO algorithm behaves in accordance with the results given in Table II. It has been observed that the PDO algorithm has difficulties in solving F8-F10 functions since their optimization is difficult.

The boxplots are provided in Fig. 4. It is seen from this figure that there are output values for F4 that PDO cannot keep within the presented range. In F10, only one result is the outlet value, while for the other eight functions, no outlet value is seen. This leads to the conclusion that PDO keeps the results within a certain range and is a stable algorithm.

The purpose of Wilcoxon's test is to see whether the PDO algorithm is more advantageous than other algorithms [37]. When Table III is examined, the PDO is seen to be different and more advantageous than the other algorithms in F1, F2, F3, F8, and F10 functions. The PDO algorithm has lost its superiority over the RSA algorithm in the F4 and F9 functions. In F4, F5, and F6 functions, it was equal to the AOA algorithm. However, the PDO algorithm is mostly in a winning position, indicating its superiority and competitiveness.

B. Results of MLP Training

In this study, MATLAB software was used for MLP training. Fair treatment was guaranteed in network training by performing

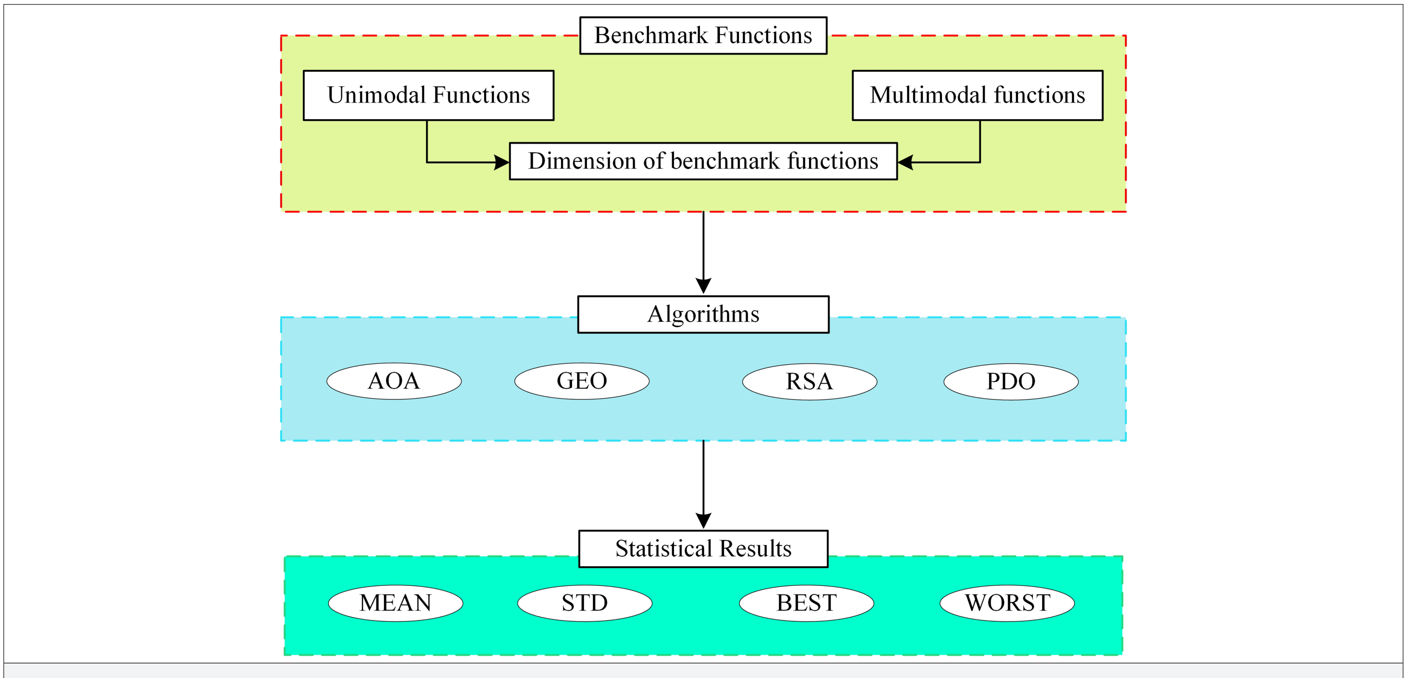

Fig. 2. Optimization process.

TABLE II. STATISTICAL PERFORMANCE COMPARISON OF THE AOA, GEO, RSA, AND PDO ALGORITHMS

No	Metric	AOA	GEO	RSA	PDO
F1	Mean	4.6675E+09	5.6838E+11	6.9282E+04	**5.5828E+04**
	Std	1.6305E+10	4.3674E+11	1.6814E+04	7.4975E+03
	Best	7.5185E+05	7.2459E+10	4.9530E+04	4.5786E+04
	Worst	1.0237E+11	2.4630E+12	1.1975E+05	7.3267E+04
F2	Mean	1.9276E+01	5.3137E+03	1.8283E+01	**1.7566E+01**
	Std	3.8864E−01	1.8415E+03	4.9333E−01	1.2994E−01
	Best	1.8150E+01	1.7322E+03	1.7426E+01	1.7355E+01
	Worst	1.9848E+01	9.4806E+03	1.9464E+01	1.7797E+01
F3	Mean	1.2703E+01	1.2704E+01	1.2703E+01	**1.2702E+01**
	Std	9.5446E−04	7.0815E−04	1.8532E−05	6.8169E−06
	Best	1.2702E+01	1.2703E+01	1.2702E+01	1.2702E+01
	Worst	1.2706E+01	1.2706E+01	1.2703E+01	1.2702E+01
F4	Mean	1.2297E+04	1.3407E+04	**9.9798E+03**	1.0557E+04
	Std	5.5766E+03	4.6490E+03	3.2192E+03	4.8274E+03
	Best	4.5436E+03	5.5480E+03	5.5456E+03	4.2671E+03
	Worst	2.6879E+04	2.3579E+04	1.9860E+04	2.6256E+04
F5	Mean	4.0103E+00	4.4053E+00	4.3393E+00	**3.7113E+00**
	Std	8.4471E−01	8.0436E−01	5.7641E−01	5.0457E−01
	Best	2.5054E+00	2.8771E+00	3.3422E+00	2.6725E+00
	Worst	6.0064E+00	6.3395E+00	5.8566E+00	4.5980E+00
F6	Mean	9.0372E+00	1.3284E+01	1.1144E+01	**8.8861E+00**
	Std	9.2095E−01	8.2895E−01	7.3069E−01	9.2593E−01
	Best	6.8739E+00	1.0330E+01	9.3207E+00	6.6225E+00
	Worst	1.1254E+01	1.4965E+01	1.2360E+01	1.0652E+01
F7	Mean	**2.4736E+02**	1.4578E+03	9.5684E+02	6.6682E+02
	Std	1.1114E+02	2.4024E+02	1.5546E+02	1.8803E+02
	Best	1.0365E+02	9.2333E+02	6.1968E+02	1.0246E+02
	Worst	5.9423E+02	1.8267E+03	1.2545E+03	9.8505E+02
F8	Mean	**5.5575E+00**	7.2670E+00	6.2475E+00	6.0263E+00
	Std	5.3766E−01	3.3913E−01	4.1651E−01	3.4939E−01
	Best	3.7995E+00	6.3814E+00	5.3817E+00	5.0980E+00
	Worst	6.5016E+00	7.7782E+00	6.9490E+00	6.5401E+00
F9	Mean	**9.3826E+02**	2.4455E+03	1.2732E+03	1.3619E+03
	Std	3.9719E+02	8.3920E+02	3.0725E+02	4.0723E+02
	Best	1.9603E+02	4.5674E+02	6.8686E+02	6.6523E+02
	Worst	1.8879E+03	4.6029E+03	2.0662E+03	2.4633E+03
F10	Mean	2.0132E+01	**1.8679E+01**	2.0493E+01	2.0351E+01
	Std	1.2691E−01	4.5487E−01	1.0150E−01	1.4323E−01
	Best	1.9337E+01	1.6980E+01	2.0137E+01	2.0003E+01
	Worst	2.0252E+01	1.9343E+01	2.0646E+01	2.0603E+01

AOA, Archimedes optimization algorithm; GEO, golden eagle optimization; RSA, reptile search algorithm; PDO, Prairie dog optimization.

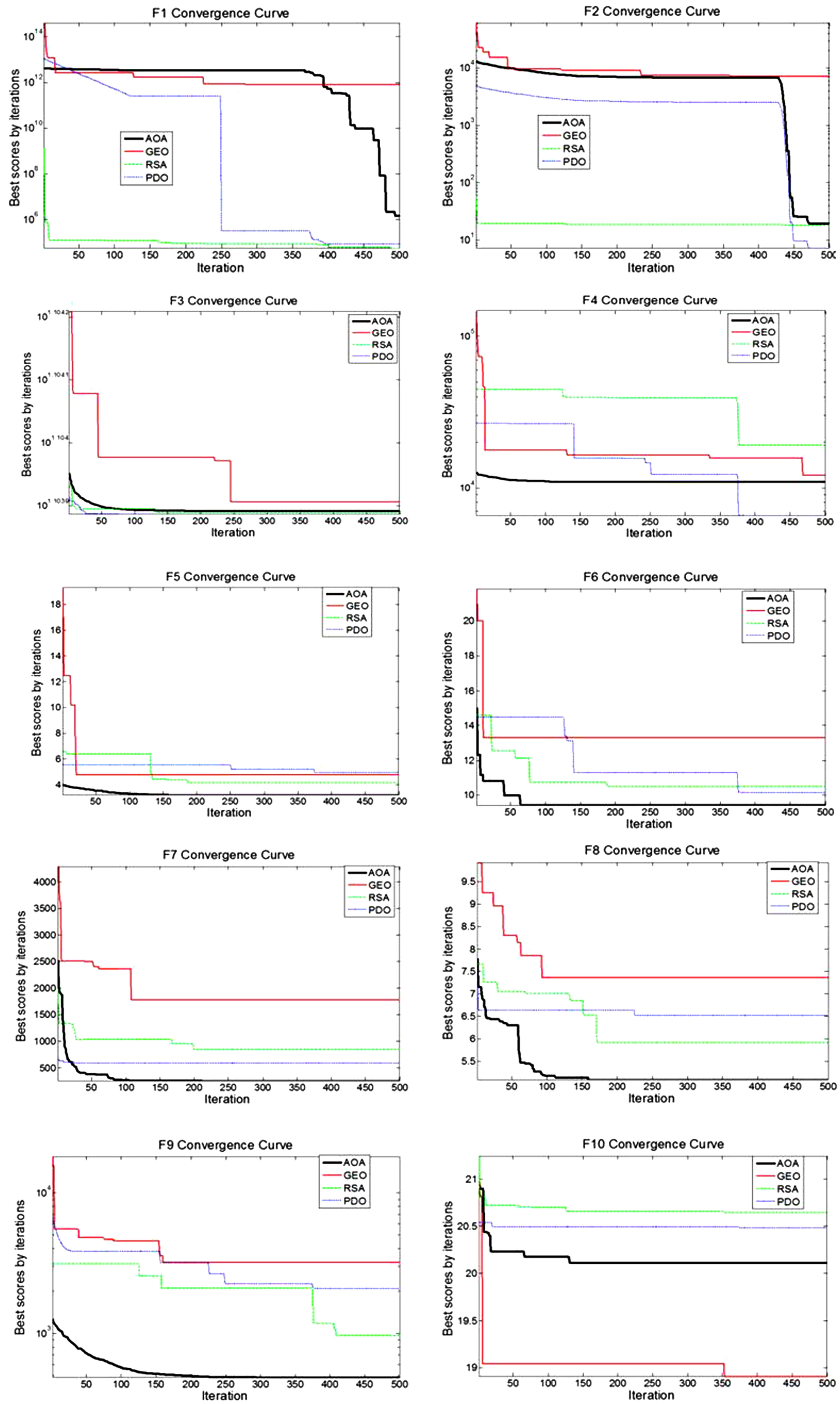

Fig. 3. Convergence curve plots of CEC 2019 functions.

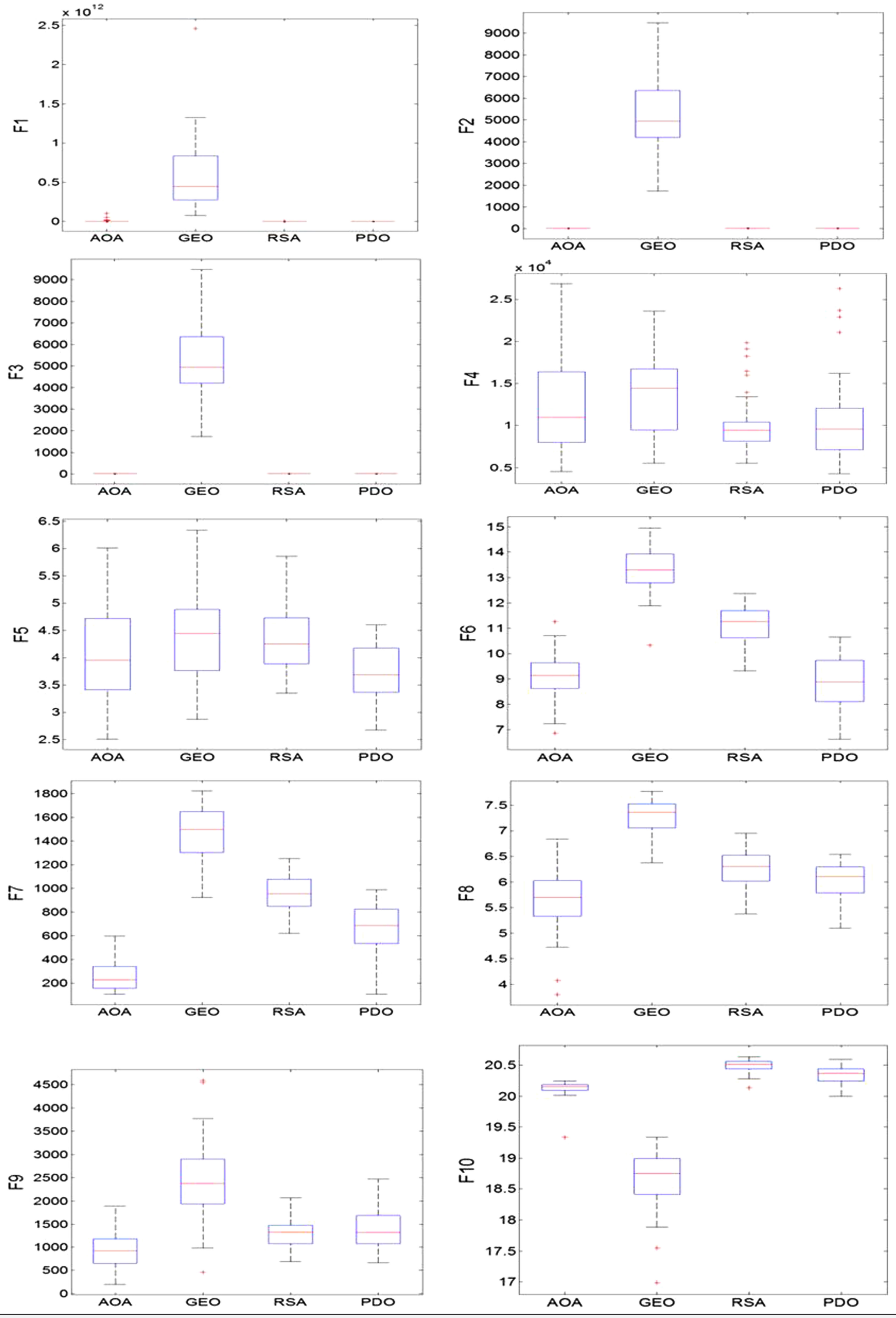

Fig. 4. Comparing statistical results according to boxplot.

TABLE III. RESULTS OF WILCOXON'S TEST

No	Item	AOA Versus PDO	GEO Versus PDO	RSA Versus PDO
F1	*p*-value	5.1453E−10	5.1453E−10	3.1835E−06
	Winner	PDO	PDO	PDO
F2	*p*-value	5.1453E−10	5.1453E−10	9.3064E−10
	Winner	PDO	PDO	PDO
F3	*p*-value	8.6484E−09	5.1453E−10	5.1453E−10
	Winner	PDO	PDO	PDO
F4	*p*-value	1.1532E−01	7.9392E−05	7.0771E−01
	Winner	=	PDO	RSA
F5	*p*-value	9.1560E−02	5.1363E−05	1.7624E−05
	Winner	=	PDO	PDO
F6	*p*-value	4.2017E−01	5.1453E−10	5.4615E−10
	Winner	=	PDO	PDO
F7	*p*-value	8.2719E−10	5.1453E−10	2.9672E−09
	Winner	AOA	PDO	PDO
F8	*p*-value	8.2719E−10	5.1453E−10	2.9672E−09
	Winner	AOA	PDO	PDO
F9	*p*-value	9.2651E−06	2.4440E−08	3.5830E−01
	Winner	AOA	PDO	RSA
F10	*p*-value	1.1402E−08	5.1453E−10	1.9997E−05
	Winner	PDO	GEO	PDO

AOA, Archimedes optimization algorithm; GEO, golden eagle optimization; RSA, reptile search algorithm; PDO, Prairie dog optimization.
= The algorithms performed equally.

30 independent runs. Initially, randomly distributed search agents (*n*) were incremented each time as $N = 50$, 100, 150, and 200, and the results were evaluated according to the statistical measures given in Table IV and the presented results in Figs 5 and 6. Although the classification ability of the PDO algorithm is at an average level, it provides very successful results in MSE values.

TABLE IV. RESULTS OF MSE AND CLASSIFICATION RATE

	Search Agents No (*N*)			
Metric	**50**	**100**	**150**	**200**
Best	0.0319	0.0265	0.0263	0.0246
Std	0.1073	0.1232	0.0913	0.0429
Classification rate	62.0000	62.7500	65.3333	66.6667
Rank	4	3	2	1

MSE, mean squared error.

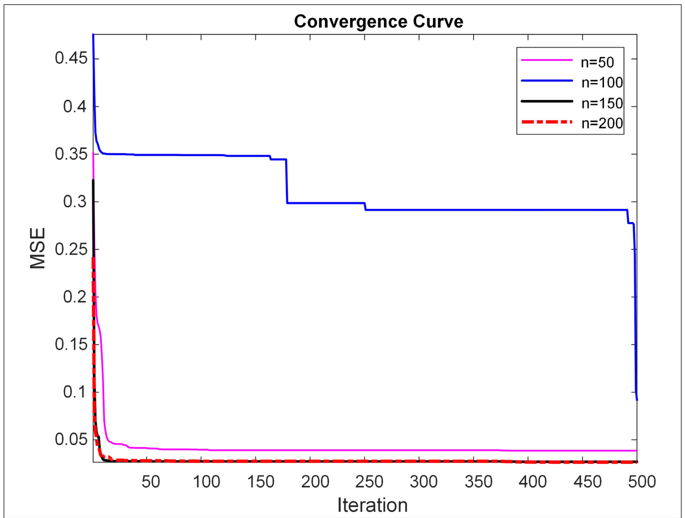

Fig. 5. Convergence curve of MSE values by number of search agents. MSE, mean squared error.

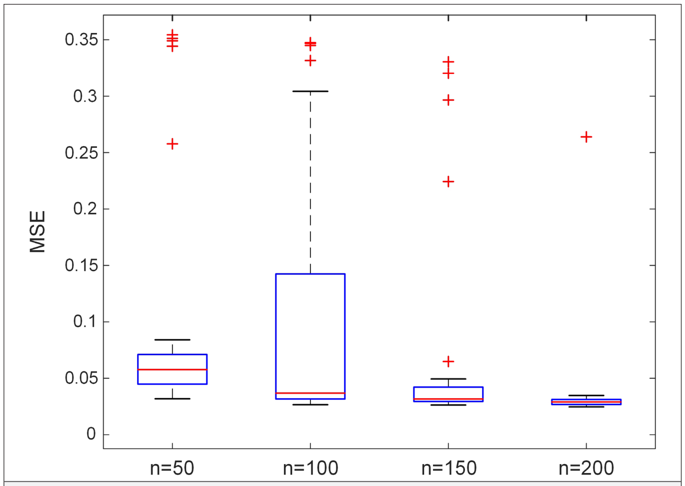

Fig. 6. Boxplot of MSE values by number of search agents. MSE, mean squared error.

When the convergence curve graph is examined, as the number of search agents increases, the MSE value approaches 0, and the error rate decreases. The box plots show that the number of search agents is 200, and the values are close to each other with an error rate closer to 0.

V. CONCLUSION

This paper investigates the performance of the PDO algorithm for MLP training. The competitiveness of the PDO algorithm is tested through the CEC 2019 functions. As seen from the presented results, PDO suffers from providing desirable values for some of the test functions, although it demonstrates a great advantage in the difficult problem set of CEC 2019. Therefore, further enhancements to this algorithm in various ways may be performed in future potential works. Nonetheless, the PDO algorithm's highly competitive structure offers a great deal for tackling the optimization of real-world problems. This study only reveals a specific application of this algorithm, and the results show how capable it is for MLP training, which indicates its good potential for different challenging engineering optimization problems.

Peer-review: Externally peer-reviewed.

Author Contributions: All authors contributed equally.

Declaration of Interests: The authors have no conflict of interest to declare.

Funding: The authors declared that this study has received no financial support.

REFERENCES

1. E. Eker, M. Kayri, S. Ekinci, and D. Izci, "A new fusion of ASO with SA algorithm and its applications to MLP training and DC motor speed control," *Arab. J. Sci. Eng.*, vol. 46, no. 4, pp. 3889–3911, 2021. [CrossRef]
2. X. S. Yang, and A. H. Gandomi, "Bat algorithm: A novel approach for global engineering optimization," *Eng. Comput.*, vol. 29, no. 5, 464–483, 2012. [CrossRef]
3. S. Mirjalili, "How effective is the Grey Wolf optimizer in training multi-layer perceptrons," *Appl. Intell.*, vol. 43, no. 1, pp. 150–161, 2015. [CrossRef]
4. Y. Nesterov, "Gradient methods for minimizing composite functions," *Math. Program.*, vol. 140, no. 1, pp. 125–161, 2013. [CrossRef]
5. K. Sörensen, and F. Glover, 2013, "Metaheuristics," *Encyclopedia of Operations Research and Management Science, 62*, pp. 960–970.
6. I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN Comput. Sci.*, vol. 2, no. 3, pp. 160, 2021. [CrossRef]
7. S. E. De León-Aldaco, H. Calleja, and J. A. Alquicira, "Metaheuristic optimization methods applied to power converters: A review," *IEEE Trans. Power Electron.*, vol. 30, no. 12, pp. 6791–6803, 2015. [CrossRef]
8. D. H. Wolpert, and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, 1997. [CrossRef]
9. H. Y. Chong, H. J. Yap, S. C. Tan, K. S. Yap, and S. Y. Wong, "Advances of metaheuristic algorithms in training neural networks for industrial applications," *Soft Comput.*, vol. 25, no. 16, pp. 11209–11233, 2021. [CrossRef]
10. A. Kattan, R. Abdullah, and R. A. Salam, "Harmony search based supervised training of artificial neural networks," In 2010 International Conference on Intelligent Systems, Modelling and Simulation. IEEE Publications, 2010, pp. 105–110. [CrossRef]
11. A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," in *Computer*, vol. 29, no. 3, pp. 31–44, 1996. [CrossRef]
12. A. Pinkus, "Approximation theory of the MLP model in neural networks," *Acta Numer.*, vol. 8, pp. 143–195, 1999. [CrossRef]
13. I. Aljarah, H. Faris, and S. Mirjalili, "Optimizing connection weights in neural networks using the whale optimization algorithm," *Soft Comput.*, vol. 22, no. 1, pp. 1–15, 2018. [CrossRef]
14. H. Taud, and J. Mas, "Multilayer Perceptron (MLP)," In *Lect. Notes Geoinf. Cartogr.*, M. Camacho Olmedo, M. Paegelow, J. F. Mas, and F. Escobar, Ed. *Geomatic Approaches for Modeling Land Change Scenarios*. Cham: Springer, 2018. [CrossRef]
15. V. K. Ojha, A. Abraham, and V. Snášel, "Metaheuristic design of feedforward neural networks: A review of two decades of research," *Eng. Appl. Artif. Intell.*, vol. 60, pp. 97–116, 2017. [CrossRef]
16. H. Ramchoun, M. Amine, J. Idrissi, Y. Ghanou, and M. Ettaouil, "Multilayer Perceptron: Architecture optimization and training," *IJIMAI*, vol. 4, no. 1, 2016. [CrossRef]
17. E. Eker, M. Kayri, S. Ekinci, and D. Izci, "Training multi-layer Perceptron using Harris hawks optimization," In 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). IEEE Publications, 2020, pp. 1–5. [CrossRef]
18. N. H. Kadhim, and Q. Mosa, "Review optimized artificial neural network by meta-heuristic algorithm and its applications," *J. Al-Qadisiyah Comput. Sci. Math.*, vol. 13, no. 3, p. 34, 2021.
19. A. E. Ezugwu, J. O. Agushaka, L. Abualigah, S. Mirjalili, and A. H. Gandomi, "Prairie dog optimization algorithm," *Neural Comput. Appl.*, vol. 34, no. 22, pp. 20017–20065, 2022. [CrossRef]
20. A. W. Mohamed, A. A. Hadi, and A. K. Mohamed, "Gaining-sharing knowledge based algorithm for solving optimization problems: A novel nature-inspired algorithm," *Int. J. Mach. Learn. Cybern.*, vol. 11, no. 7, pp. 1501–1529, 2020. [CrossRef]
21. B. Abdollahzadeh, F. Soleimanian Gharehchopogh, and S. Mirjalili, "Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems," *Int. J. Intell. Syst.*, vol. 36, no.10, pp. 5887–5958, 2021. [CrossRef]
22. J. Brest, M. S. Maučec, and B. Bošković, "Single objective real-parameter optimization: Algorithm jSO," 2017 IEEE Congress on Evolutionary Computation (CEC), San Sebastian, 2017, pp. 1311–1318. [CrossRef]
23. A. W. Mohamed, A. A. Hadi, A. M. Fattouh, and K. M. Jambi, "LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems," in Proc. IEEE Congr. Evol. Comput. (CEC), pp. 145–152, 2017. [CrossRef]. (LSHADESPACMA algorithm).
24. A. Kumar, R. K. Misra, and D. Singh, "Improving the local search capability of Effective Butterfly Optimizer using Covariance Matrix Adapted Retreat Phase," 2017 IEEE Congress on Evolutionary Computation (CEC), San Sebastian, 2017, pp. 1835–1842. [CrossRef]
25. A. A. Hadi, A. W. Mohamed, and K. M. Jambi, "Single-objective real-parameter optimization: Enhanced LSHADE-SPACMA algorithm," *Heuristics Optim. Learn.*, vol. 906, pp. 103–121, 2021.
26. A. W. Mohamed, A. A. Hadi, A. K. Mohamed, and N. H. Awad, "Evaluating the performance of adaptive GainingSharing knowledge based algorithm on CEC 2020 benchmark problems," IEEE Congress on Evolutionary Computation (CEC), Vol. 2020, 2020, pp. 1–8. [CrossRef]
27. A. W. Mohamed, A. A. Hadi, and K. M. Jambi, "Novel mutation strategy for enhancing SHADE and LSHADE algorithms for global numerical optimization," *Swarm Evol. Comput.* Amsterdam: Elsevier, vol. 50, p. 100455, 2019. [CrossRef]
28. A. K. Mohamed, and A. W. Mohamed, "Real-parameter unconstrained optimization based on enhanced AGDE algorithm," In *Machine Learning Paradigms: Theory and Application. Studies in Computational Intelligence*, Vol. 801, A. Hassanien, Ed. Cham: Springer, 2019, 431–450. [CrossRef]
29. J. L. Hoogland, *The Black-Tailed Prairie Dog: Social Life of a Burrowing Mammal*. Chicago: University of Chicago Press, 1995.
30. Ş. Gülcü, "Training of the feed forward artificial neural networks using dragonfly algorithm," *Appl. Soft Comput.*, vol. 124, p. 109023, 2022. [CrossRef]
31. S. Haykin, and N. Network, "A comprehensive foundation," *Neural Netw.*, vol. 2, p. 41, 2004.
32. D. Dua, and C. Graff, 2019. *UCI Machine Learning Repository* [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
33. E. Pashaei, and E. Pashaei, "Training feedforward neural network using enhanced Black Hole algorithm: A case study on COVID-19 related ACE2 gene expression classification," *Arab. J. Sci. Eng.*, vol. 46, no. 4, pp. 3807–3828, 2021. [CrossRef]
34. A. Mohammadi-Balani, M. D. Nayeri, A. Azar, and M. Taghizadeh-Yazdi, "Golden eagle optimizer: A nature-inspired metaheuristic algorithm," *Comput. Ind. Eng.*, vol. 152, p. 107050, 2021. [CrossRef]
35. L. Abualigah, M. A. Abd Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, "Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer," *Expert Syst. Appl.*, vol. 191, p. 116158, 2022. [CrossRef]
36. F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, "Archimedes optimization algorithm: A new metaheuristic algorithm for solving optimization problems" *Appl. Intell.*, vol. 51, no. 3, pp. 1531–1551, 2021. [CrossRef]
37. J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, 2011. [CrossRef]

Erdal Eker graduated in Mathematics from Van Yuzuncu Yil University, Turkey. He received his MSc in Applied Mathematics from Ataturk University, Turkey, and his PhD from Yuzuncu Yil University in Statistics. He is currently an instructor at Mus Alparslan University, working on Artificial intelligence, the applications of metaheuristic optimization, and statistic.

Murat Kayri graduated from Gazi University with a degree in Computer Science in 1994. He received his Msc from Yuzuncu Yil University in Electrical and Electronic Engineering and his PhD in Biometry and Genetics. He is currently a Professor at Yuzuncu Yil University, working on Artificial intelligence and control technologies.

Serdar Ekinci was born in Diyarbakir, Turkey 1984. He received a BS degree in Control Engineering and MS and PhD degrees in Electrical Engineering from Istanbul Technical University (ITU), in 2007, 2010, and 2015, respectively. Since 2016, he has been an Assistant Professor at the Computer Engineering Department at Batman University, Batman, Turkey. His areas of interest are electrical power systems, stability, control technology, and the applications of heuristic optimization to power system control.

Mehmet Ali Kaçmaz was born in Diyarbakır, Turkey in 1982. He graduated from Marmara University, Turkey, in Computer and Control Teacher. He received his Msc from Sabahattin Zaim University, Turkey, in Education Management and Supervision. He is currently headmaster at one of Maritime Vocational High School, Turkey.