symmetry

MDPI

Article

A Novel Hybrid Method Based on the Marine Predators Algorithm and Adaptive Neuro-Fuzzy Inference System for the Identification of Nonlinear Systems

Ceren Baştemur Kaya

Department of Computer Technologies, Nevsehir Vocational School, Nevsehir Haci Bektas Veli University, 50300 Nevşehir, Türkiye; ceren@nevsehir.edu.tr

Abstract: In this study, a hybrid method based on the marine predators algorithm (MPA) and adaptive neuro-fuzzy inference system (ANFIS) is presented to identify nonlinear systems exhibiting symmetrical or asymmetrical behavior. In other words, the antecedent and conclusion parameters of the ANFIS are adjusted by the MPA. The performance of the MPA is evaluated on eight nonlinear systems. The mean squared error is used as the error metric. Successful results were obtained on the eight systems. The best mean training error values belonging to the eight systems are 1.6×10^{-6}, 3.2×10^{-3}, 1.5×10^{-5}, 9.2×10^{-6}, 3.2×10^{-5}, 2.3×10^{-3}, 1.7×10^{-5}, and 8.7×10^{-6}. In the ANFIS training carried out to solve the related problems, the performance of the MPA was compared with the butterfly optimization algorithm, the flower pollination algorithm, moth–flame optimization, the multi-verse optimizer, the crystal structure algorithm, the dandelion optimizer, the RIME algorithm, and the salp swarm algorithm. The results have shown that the performance of the MPA mostly outperforms other algorithms in both training and testing processes.

Keywords: marine predators algorithm; neuro-fuzzy; ANFIS; nonlinear systems; system identification; swarm intelligence; symmetry in nonlinear systems

Citation: Baştemur Kaya, C. A Novel Hybrid Method Based on the Marine Predators Algorithm and Adaptive Neuro-Fuzzy Inference System for the Identification of Nonlinear Systems. *Symmetry* **2023**, *15*, 1765. https://doi.org/10.3390/sym15091765

Academic Editor: Manuel De La Sen

Received: 24 August 2023
Revised: 10 September 2023
Accepted: 12 September 2023
Published: 15 September 2023

1. Introduction

One of the most important artificial intelligence techniques used for prediction, identification, and modeling is the ANFIS. The training process of the ANFIS has an important place in achieving successful results. Recently, meta-heuristic algorithms have been used extensively in the training of the ANFIS [1]. Yaseen et al. [2] proposed a novel hybrid model based on the ANFIS and the FA for river streamflow forecasting. Tien Bui et al. [3] proposed three hybrid methods based on the ANFIS by using the CA, BA, and IWO for flood susceptibility modeling. Chen et al. [4] optimized the parameters of the ANFIS by utilizing the SBO and TLBO for landslide susceptibility mapping. Namely, they presented two new hybrid models based on the ANFIS. AlRassas et al. [5] carried out a time series analysis based on the ANFIS to forecast oil production from two different oil fields in China and Yemen. They proposed a novel method called AO-ANFIS based on the AO and the ANFIS. They compared the performance of the proposed hybrid method with the GWO, SCA, GA, PSO, and SMA-based approaches and the original ANFIS. Karaboga and Kaya [6] improved a variant of the ABC algorithm called aABC to train the ANFIS and proposed a hybrid method based on the aABC algorithm and the ANFIS to identify the nonlinear dynamic systems. They compared the performance of the proposed method with different neuro-fuzzy-based approaches and reported that the proposed method was successful. Pourdaryaei et al. [7] proposed a hybrid approach (ANFIS-BSA) based on the BSA and ANFIS for short-term electricity price forecasting. The BSA was used in the learning process of the ANFIS approach. The performance of the ANFIS-BSA was compared with different artificial intelligence approaches. Riahi-Madvar et al. [8] trained the ANFIS by using the FA, GA, GWO, PSO, and DE for short- to long-term forecasting of river flows.

Ahmed et al. [9] presented a hybrid approach based on the KH optimization algorithm and the ANFIS for wind speed forecasting and performed the parameter optimization of the ANFIS with KH. Liu et al. [10] presented an observer-based adaptive fuzzy finite-time attitude control strategy for quadrotor unpiloted aerial vehicles (UAVs). The performance of their proposed controller was compared with approaches such as PD, finite-time DSC, DO–NTSMC, adaptive fuzzy finite-time control, FTC–NFTSMC, and adaptive NFTSMC. It was stated that the proposed control strategy was effective.

Although the MPA was developed in 2020, it quickly became popular. The main reason for its popularity is that it gives successful results. Since its development, the MPA has been used to solve many real-world problems [11]. Abd Elminaam et al. [12] proposed an approach named MPA-KNN by hybridizing the MPA with k-Nearest Neighbors (k-NN) to evaluate dimension reduction in feature selection. The performance of the proposed algorithm was compared with the results of eight metaheuristic algorithms by using 18 UCI medical datasets. Abdel-Basset et al. [13] offered an improved MPA to provide the optimal values of photovoltaic parameters. They reported that their proposed algorithm was connected with the measured current–voltage data and their algorithm could be beneficial in estimating the parameters of photovoltaic models. Soliman et al. [14] presented a modified MPA to obtain the electrical parameters of the triple-diode photovoltaic (TDPV) model of a photovoltaic (PV) panel. Abd Elaziz et al. [15] proposed an enhanced MPA (EMPA) for determining the unknown parameters of different PV substances. For the purpose of determining the effect of the improved algorithm, they compared the MPA and heterogeneous comprehensive learning PSO (HCLPSO) algorithms with the improved algorithm by using three real datasets. It has been indicated that the data fitting, convergence rate, stability, and consistency of the proposed algorithm were better than the other algorithms. Zhong et al. [16] improved a new version of the MPA. In the improved algorithm, they created an archive that stored the non-dominated Pareto optimal solutions to select the effective solutions. They used the CEC2019 multi-modal multi-objective benchmark functions. The improved algorithm was compared with nine metaheuristics algorithms. They reported that the improved algorithm has effective results that are better than the other nine algorithms. Fan et al. [17] proposed a modified MPA (MMPA). The performance of the MMPA was evaluated on different problems. It has been indicated that the MMPA gave successful results. Al-Qaness et al. [18] presented the MPAmu, which utilized additional mutation operators, and they optimized the ANFIS with the MPAmu to estimate wind power using time series datasets from wind turbines located in France. They compared the proposed model with the ANFIS, the ANFIS that was modified with different metaheuristic algorithms, and an SVM, FFNN, and LSTM. It has been indicated that the proposed model increases the prediction accuracy of the traditional ANFIS. Al-Qaness et al. [19] used a method based on the MPA and ANFIS to estimate the number of infected people in four countries, Italy, Iran, Korea, and the USA. They compared several methods to the performance of their proposed method. It has been indicated that the performance of their proposed method was better. Shaheen et al. [20] proposed an algorithm improved with the MPA and PSO (IMPAPSO) to implement on the non-linearity of the optimal reactive power dispatch (ORPD) problem. They utilized IEEE 30 bus, IEEE 57 bus, and IEEE 118 bus systems to evaluate the effect of IMPAPSO. Moreover, IMPAPSO was compared with other optimization methods. They reported that IMPAPSO was shown to be effective in the electric power networks' behavior. Houssein et al. [21] presented a hybrid algorithm (MPA-CNN) by using the MPA and a CNN. The MPA-CNN included a combination of heavy feature extraction and classification techniques. Ikram et al. [22] proposed a hybrid model with the ANFIS and MPA to estimate short-term significant wave heights. The results were compared with two different models, the ANFIS with the GA (ANFIS-GA) and the ANFIS with PSO (ANFIS-PSO). They reported that the proposed model generally was better than the other models. Al-Qaness et al. [23] used a model, the ANFIS that was optimized with the chaotic MPA (CMPA), to estimate COVID-19 cases in hotspot regions. They compared the proposed model with three artificial intelligence-based models including the original ANFIS and two modified versions of the ANFIS model using both the original MPA and PSO, to determine the effect of the proposed model. They reported that the results of their model were better than those of the other models.

In light of the above information, it is apparent that meta-heuristic algorithms are used in ANFIS training. Especially in some studies, it is apparent that the performance of meta-heuristic algorithms in ANFIS training has been examined in detail. In these studies, mostly nonlinear systems are used as test functions. Ghomsheh et al. [24] proposed a PSO-based hybrid method for ANFIS training. Karaboğa and Kaya [25] evaluated the performance of the ABC algorithm in ANFIS training on the basis of nonlinear systems. Zangeneh et al. [26] conducted ANFIS training with DE-based algorithms. Haznedar and Kalinli [27] optimized the parameters of the ANFIS using a simulated annealing algorithm. Haznedar and Kalinli [28] trained the ANFIS model with a genetic algorithm for systems identification. Marzi et al. [29] proposed a hybrid training approach based on the bees algorithm for ANFIS training. Canayaz [30] carried out ANFIS training with MFO. As can be observed, it is used in ANFIS training due to the strengths of metaheuristic algorithms. From the above information, it is clear that the MPA is a powerful and successful algorithm. Although the MPA is used to solve many real-world problems, there are limited studies on MPA-based ANFIS training. In particular, there are hardly any studies involving direct performance analysis. Therefore, there is a need to examine the performance of the MPA in ANFIS training. In particular, revealing the superiorities according to different algorithms will shed light on future studies. Therefore, in this study, the aim was to evaluate the performance of the MPA in ANFIS training. When the content, problem structure, and scope of the study are evaluated together, it is clear that it is an innovative study. This study makes important contributions to the literature. These are listed below:

- The performance of the MPA in ANFIS training is examined in detail. The effect of the number of parameters of the ANFIS on the result was analyzed. The advantages and disadvantages of the proposed method were evaluated.
- ANFIS training was carried out using the MPA for the identification of nonlinear systems. Analyses were performed on eight systems with different characteristics.
- The performance of the MPA is compared with eight different meta-heuristic algorithms. These algorithm are the BOA [31], FPA [32], MFO [33], MVO [34], SSA [35], CryStAl [36], DO [37], and RIME algorithm [38]. The success of the MPA according to these algorithms was evaluated.
- The use case of the proposed method for different problems other than system identification was evaluated.

The organizational structure of this study is as follows: the MPA and ANFIS are introduced in Section 2. Section 3 presents simulation results. In Section 4, Discussion is located. In the last section, Conclusions are given.

2. Materials and Methods

2.1. Marine Predators Algorithm (MPA)

Faramarzi et al. proposed the MPA, whose development was inspired by the foraging strategies of ocean predators [39]. Lévy and Brownian motions are considered in the MPA. The optimal encounter rate of prey and predators is taken into account.

The MPA is a population-based meta-heuristic algorithm. The optimization process of the MPA starts with a random solution and (1) is used to start. In (1), X_{min} symbolizes the lower bound, X_{max} the upper bound, and *rand* is a random number between [0, 1].

$$X_0 = X_{min} + rand(X_{max} - X_{min}) \tag{1}$$

There are two matrices named *Prey* and *Elite* in the MPA. They have the same dimensions. When constructing the *Elite* matrix, the best solution is appointed as the best predator.

The search for and finding of prey is checked via this matrix. $\overrightarrow{X^I}$ denotes the top predator vector, n search agents, d dimensions. The prey and predator both are the search agents.

$$Elite = \begin{bmatrix} X_{1,1}^I & X_{1,2}^I & \cdots & X_{1,d}^I \\ X_{2,1}^I & X_{2,2}^I & \cdots & X_{2,d}^I \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ X_{n,1}^I & X_{n,2}^I & \cdots & X_{n,d}^I \end{bmatrix}_{n\ x\ d} \tag{2}$$

In the *Prey* matrix, the jth dimension of the ith prey is depicted with $X_{i,j}$. In the MPA, the whole optimization process is directly relevant to these two matrices. Predators use this matrix to update their positions.

$$Prey = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,d} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,d} \\ X_{3,1} & X_{3,2} & \cdots & X_{3,d} \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ X_{n,1} & X_{n,2} & \cdots & X_{n,d} \end{bmatrix}_{n\ x\ d} \tag{3}$$

There are three phases in the MPA. Details of the phases are explained below.

If *Iter* < ((*Max_Iter*)/3), phase 1 is realized. *Iter* symbolizes the current iteration number and *Max_Iter* the maximum iteration number. The best strategy is one wherein the predator must stop. In mathematical Formula (4) of phase 1, R_B depicts a vector characterizing the Brownian motion and including random numbers based on a normal distribution. P denotes a constant number with a value of 0.5. R is a random number between 0 and 1.

$$\begin{gathered} \overrightarrow{stepsize}_i = \vec{R}_B \otimes \left(\overrightarrow{Elite}_i - \vec{R}_B \otimes \overrightarrow{Prey}_i \right) \ i = 1, \ldots n \\ \overrightarrow{Prey}_i = \overrightarrow{Prey}_i + P.\vec{R} \otimes \overrightarrow{stepsize}_i \end{gathered} \tag{4}$$

If *((Max_Iter)/3)* < *Iter* < *((2Max_Iter)/3*, phase 2 occurs. When the prey's move is Lévy, the predator's move must be Brownian. The responsibility of the prey is exploitation, which is formulated in (5), and the responsibility of the predator is also exploration, which is depicted in (6). In (5), $\vec{R}_L$ is a random-numbers vector denoting Lévy movement. The multiplication of $\vec{R}_L$ and *Prey* denotes the prey move, and the prey move is represented by adding the step size to the prey location. In (6), CF depicts an adaptive parameter. The step size for the predator move is controlled via CF. The multiplication of $\vec{R}_B$ and *Elite* symbolizes the predator move.

$$\begin{gathered} \overrightarrow{stepsize}_i = \vec{R}_L \otimes \left(\overrightarrow{Elite}_i - \vec{R}_L \otimes \overrightarrow{Prey}_i \right) \ i = 1, \ldots n/2 \\ \overrightarrow{Prey}_i = \overrightarrow{Prey}_i + P.\vec{R} \otimes \overrightarrow{stepsize}_i \end{gathered} \tag{5}$$

$$\begin{gathered} \overrightarrow{stepsize}_i = \vec{R}_B \otimes \left(\vec{R}_B \otimes \overrightarrow{Elite}_i - \overrightarrow{Prey}_i \right) \ i = n/2, \ldots n \\ \overrightarrow{Prey}_i = \overrightarrow{Elite}_i + P.CF \otimes \overrightarrow{stepsize}_i \\ CF = \left(1 - \frac{Iter}{Max\_Iter}\right)^{(2\frac{Iter}{Max\_Iter})} \end{gathered} \tag{6}$$

If *Iter* > ((2*Max_Iter*)/3), phase 3 is realized. As the best strategy, the predator's move is Lévy. In (7), the multiplication of $\vec{R}_L$ and *Elite* symbolizes the predator move.

$$\begin{gathered}\overrightarrow{stepsize}_i = \vec{R}_L \otimes \left(\vec{R}_L \otimes \overrightarrow{Elite}_i - \overrightarrow{Prey}_i\right) i = 1, \ldots n \\ \overrightarrow{Prey}_i = \overrightarrow{Elite}_i + P.CF \otimes \overrightarrow{stepsize}_i\end{gathered} \tag{7}$$

The factors like eddy formation or Fish Aggregating Devices (FADs) can affect the marine predator's course of action. In the MPA, this is called the FADs effect. The FADs effect is formulized by using (8) with a value of 0.2. $\vec{U}$ depicts the binary vector with arrays containing 0 and 1. r denotes a random number between 0 and 1. $\vec{X}_{min}$ is a vector symbolizing the lower bounds of the dimensions. $\vec{X}_{max}$ is a vector symbolizing the upper bounds of the dimensions. $r1$ and $r2$ depict the prey matrix's random indexes.

$$\overrightarrow{Prey}_i = \begin{cases} \overrightarrow{Prey}_i + CF\left[\vec{X}_{min} + \vec{R} \otimes \left(\vec{X}_{max} - \vec{X}_{min}\right)\right] \otimes \vec{U}, \; r \le FADs \\ \overrightarrow{Prey}_i + [FADs(1-r) + r]\left(\overrightarrow{Prey}_{r1} - \overrightarrow{Prey}_{r2}\right), \; r > FADs \end{cases} \tag{8}$$

2.2. Adaptive Network Fuzzy Inference System (ANFIS)

The ANFIS, one of the neuro-fuzzy models, uses the inference feature of fuzzy logic and the learning ability of an ANN [40]. It has significant advantages as it takes the strengths of both fuzzy logic and ANNs. The ANFIS consists of two parts, the antecedent and the conclusion. IF-THEN fuzzy rules determining that these parts are connected with each other are obtained. As seen in Figure 1, the ANFIS consists of five layers.

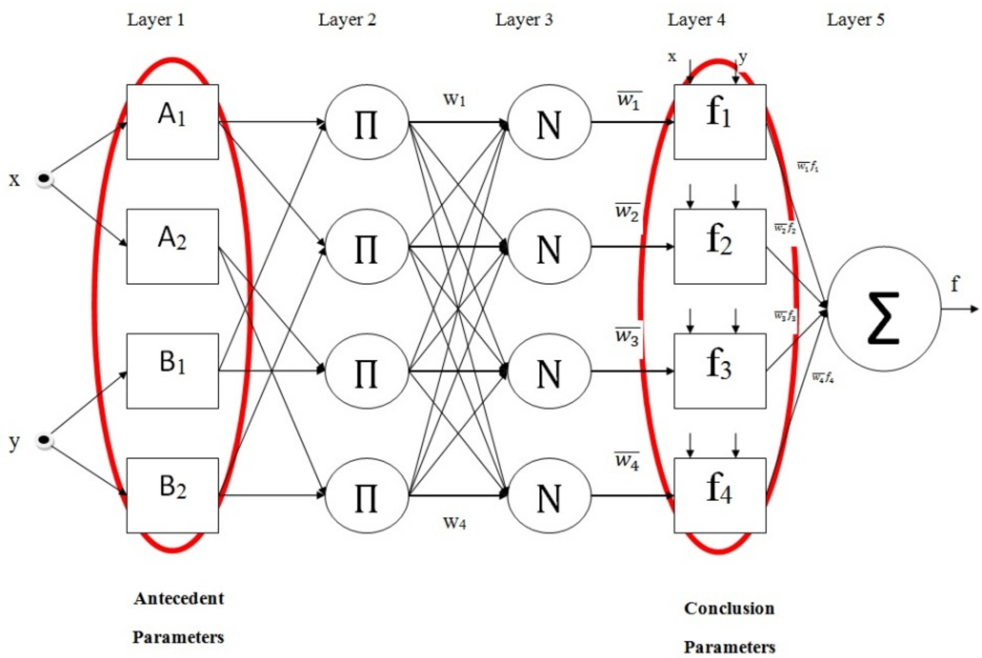

Figure 1. General structure of ANFIS [1].

Layer 1 is called the fuzzification layer. In this layer, MFs are used and fuzzy sets are obtained from input values using MFs. The parameters in the MFs structure are called antecedent parameters. The shapes of MFs are formed according to the value of the antecedent parameters. Membership functions have membership degrees. The membership degrees take a value between 0 and 1. If the generalized Bell function is used as an MF, the membership degrees are calculated using (9) and (10).

$$\mu_{A_i}(x) = gbellmf(x; a,\ b,\ c) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}} \tag{9}$$

$$O_1^1 = \mu_{A_i}(x) \tag{10}$$

Layer 2 is called the rule layer. Firing strengths are found for each rule. Membership values found in the previous layer are used in the calculation of firing strengths. Firing strength (w_i) values are found by multiplying the membership values as in (11):

$$O_1^2 = w_i = \mu_{A_i}(x) * \mu_{B_i}(y),\ \ i = 1,\ 2. \tag{11}$$

Layer 3 is named the normalization layer. Normalized firing strengths are calculated for each rule. Normalized firing strengths $\overline{w_i}$ are calculated as in (12).

$$O_1^3 = \overline{w_i} = \frac{w_i}{w_1 + w_2 + w_3 + w_4},\ \ i \in \{1,\ 2, 3, 4\} \tag{12}$$

Layer 4 is named as defuzzification layer. In this layer, the output value of each rule is calculated. Normalized firing strengths and a first-order polynomial are used as in (13) to calculate the output for each rule.

$$O_1^4 = \overline{w_i} f_i = \overline{w_i}(p_i x + q_i y + r_i) \tag{13}$$

Layer 5 is called the summation layer. The actual output of the ANFIS is obtained by summing the outputs obtained for each rule in the defuzzification layer.

$$O_1^5 = \sum_i \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{14}$$

3. Simulation Results

In this study, the performance of the MPA was evaluated in the ANFIS training carried out for the identification of nonlinear systems. Eight systems with different characteristics were used in the applications. Information about the equations, inputs, and outputs of the systems used, and the data numbers of the training and testing process, are given in Table 1. The S1 system consists of one input. In other systems except S1, there are two inputs. In addition, all systems have one input. In other words, an ANFIS model was created for S1 with an input and an output. For other systems, models consisting of two inputs and one output were used. As seen in Table 1, the dataset of all systems consists of 100 input–output data pairs. 80 data pairs were used for the training process, and 20 data pairs were utilized for the testing process. All results are given as mean squared error (MSE). A generalized Bell function was used as a membership function (MF). The results were obtained by choosing two, three, and four MFs for each input in all systems. Each application was run at least 30 times. In each application, the population size is 20 and the maximum number of generations is 2500.

The training results of the ANFIS training carried out with the MPA to identify the relevant systems are presented in Table 2. Effective results were achieved with 2, 3, and 4 MFs for S1. The best mean error value in S1 was found using three MFs. Its value is 1.6×10^{-6}. Error values at the level of 10^{-3} were reached in S2. If 2, 3, and 4 MFs are used, the mean training error values obtained are 6.1×10^{-3}, 3.9×10^{-3} and 3.2×10^{-3}, respectively. In fact, this shows that more effective results are achieved with four MFs in S2. Successful results were achieved in S3. Mean error values at the level of 10^{-5} were reached. Its best mean error value is 1.5×10^{-5}. The best mean error value of S4 is 9.2×10^{-6}. It was found with four MFs. The effective mean error value was reached with three MFs in

S5. Its mean error value is 3.2×10^{-5}. The mean error values of S6 are at the 10^{-3} level, as in S2. As the number of MFs increased in S6, the quality of the solution increased. If 2, 3, and 4 MFs are used, the mean training error values obtained are 3.0×10^{-3}, 2.8×10^{-3} and 2.3×10^{-3}, respectively. S7 and S8 are in the same situation as S6 in terms of the number of MFs. The mean error values of S7 and S8 are 1.7×10^{-5} and 8.7×10^{-6}, respectively.

Table 1. Nonlinear systems used.

System	Equation	Inputs	Output	Number of Training/Test Data
S1	$y = 4.26\left(e^{-x_1} - 4e^{-2x_1} + 3e^{-3x_1}\right)$	x_1	y	80/20
S2	$y = e^{-x_1{}^2 - 3x_2{}^2 - 4x_1x_2}$	x_1, x_2	y	80/20
S3	$y = \frac{1}{9}\left\{64 - 81\left(\left(x_1 - \frac{1}{2}\right)^2 + \left(x_2 - \frac{1}{2}\right)^2\right)\right\}^{\frac{1}{2}} - \frac{1}{2}$	x_1, x_2	y	80/20
S4	$y = \frac{1}{3}e^{-\frac{81}{16}\{(x_1-0.5)^2+(x_2-0.5)^2\}}$	x_1, x_2	y	80/20
S5	$y = 0.075e^{-\frac{(x_1-0.3)^2}{0.3^2}-(x_2-0.3)^2} + 0.094e^{-\frac{(x_1-0.8)^2}{0.3^2}-\frac{(x_2-0.8)^2}{0.4^2}}$	x_1, x_2	y	80/20
S6	$y = 0.75e^{-\frac{(9x_1-2)^2+(9x_2-2)^2}{4}} + 0.75e^{-\frac{(9x_1+1)^2}{49}-\frac{9x_2+1}{10}} + 0.5e^{-\frac{(9x_1-7)^2+(9x_2-3)^2}{4}} - 0.2e^{-(9x_1-4)^2-(9x_2-7)^2}$	x_1, x_2	y	80/20
S7	$y = \frac{1.25+\cos(5.4x_2)}{6\{1+(3x_1-1)^2\}}$	x_1, x_2	y	80/20
S8	$y = 0.7e^{-\{(-3x_1+3)^2+0.7(3x_2-3)^2\}/5}$	x_1, x_2	y	80/20

Table 2. Training process results of ANFIS training carried out with MPA for identification of nonlinear systems.

System	Number of MFs	Best (MSE)	Mean (MSE)	Worst (MSE)	Standard Deviation
S1	2	5.1×10^{-7}	8.3×10^{-6}	2.3×10^{-4}	4.2×10^{-5}
	3	1.1×10^{-7}	1.6×10^{-6}	5.3×10^{-6}	1.3×10^{-6}
	4	1.6×10^{-7}	3.7×10^{-6}	4.5×10^{-5}	8.1×10^{-6}
S2	2	4.4×10^{-3}	6.1×10^{-3}	9.1×10^{-3}	1.1×10^{-3}
	3	2.1×10^{-3}	3.9×10^{-3}	1.1×10^{-2}	1.6×10^{-3}
	4	1.3×10^{-3}	3.2×10^{-3}	8.4×10^{-3}	1.4×10^{-3}
S3	2	5.6×10^{-6}	2.4×10^{-5}	2.0×10^{-4}	3.5×10^{-5}
	3	3.3×10^{-6}	1.5×10^{-5}	3.5×10^{-5}	8.2×10^{-6}
	4	6.9×10^{-6}	1.8×10^{-5}	5.3×10^{-5}	1.1×10^{-5}
S4	2	1.3×10^{-6}	2.3×10^{-5}	1.9×10^{-4}	3.8×10^{-5}
	3	1.2×10^{-6}	1.8×10^{-5}	1.8×10^{-4}	3.2×10^{-5}
	4	1.8×10^{-6}	9.2×10^{-6}	2.5×10^{-5}	6.1×10^{-6}
S5	2	3.3×10^{-5}	5.5×10^{-5}	7.5×10^{-5}	1.0×10^{-5}
	3	1.1×10^{-5}	3.2×10^{-5}	6.0×10^{-5}	1.3×10^{-5}
	4	7.0×10^{-6}	3.4×10^{-5}	6.1×10^{-5}	1.1×10^{-5}
S6	2	1.9×10^{-3}	3.0×10^{-3}	4.1×10^{-3}	4.9×10^{-4}
	3	6.0×10^{-4}	2.8×10^{-3}	6.1×10^{-3}	1.2×10^{-3}
	4	9.0×10^{-4}	2.3×10^{-3}	4.6×10^{-3}	7.8×10^{-4}

Table 2. *Cont.*

System	Number of MFs	Best (MSE)	Mean (MSE)	Worst (MSE)	Standard Deviation
S7	2	1.2×10^{-6}	4.9×10^{-5}	4.2×10^{-4}	1.0×10^{-4}
	3	2.0×10^{-6}	2.4×10^{-5}	1.4×10^{-4}	2.9×10^{-5}
	4	2.7×10^{-6}	1.7×10^{-5}	1.4×10^{-4}	2.8×10^{-5}
S8	2	1.0×10^{-6}	4.0×10^{-5}	1.2×10^{-4}	3.2×10^{-5}
	3	1.3×10^{-6}	2.3×10^{-5}	1.5×10^{-4}	2.9×10^{-5}
	4	5.4×10^{-7}	8.7×10^{-6}	3.1×10^{-5}	8.7×10^{-6}

The testing results of the ANFIS training carried out with the MPA to identify the relevant systems are presented in Table 3. Effective results were achieved with three MFs for S1. The best mean error value is 2.2×10^{-6}. At the same time, the best error value in S1 is at 10^{-8} levels. The best mean value found for S2 is 3.0×10^{-2}. This error value was at the level of 10^{-3} in the training process. In S3, mean error values were reached at the level of 10^{-4} for two MFs and three MFs, and it is at the level of 10^{-3} for four MFs. In S4, two MFs are more efficient. Its mean error value is 1.5×10^{-4}. Generally effective test error results were achieved in S5. The mean error value obtained for two MFs and three MFs is the same, and it is 2.3×10^{-4}. For four MFs, the error is 1.6×10^{-4}. The decrease in the number of MFs for S6 increased the solution quality of the test process. For S6, this value is 3.4×10^{-3}. When two MFs and four MFs were used in S7, an error level of 10^{-4} was obtained. The best mean error value for S7 is 1.3×10^{-4}. Three MFs are more effective in S8. The mean error value obtained is 3.0×10^{-4}.

Table 4 provides information based on the best training error values. In other words, the results of the best training values found for each system are included. While creating the table, success in mean training errors was taken into account. In five of the eight systems, the best mean training error values were obtained with four MFs. In the other three systems, three MFs were more effective. The best mean error values obtained for S1, S2, S3, S4, S5, S6, S7, and S8 were 1.6×10^{-6}, 3.2×10^{-3}, 1.5×10^{-5}, 9.2×10^{-6}, 3.2×10^{-5}, 2.3×10^{-3}, 1.7×10^{-5}, and 8.7×10^{-6}, respectively. The corresponding test error values are 2.2×10^{-6}, 4.9×10^{-2}, 8.5×10^{-4}, 7.8×10^{-4}, 2.3×10^{-4}, 4.8×10^{-3}, 1.3×10^{-4}, and 3.6×10^{-4}, respectively. The training error values obtained for S2 and S6 are at the 10^{-3} level. For other systems, this value is 10^{-5} and above.

The performance of the MPA in ANFIS training to solve the relevant problem is compared with eight meta-heuristic algorithms including BSO, FPA, MFO, MVO, SSA, CryStAl, DO, and RIME, and training results are presented in Table 5. In all systems, more successful mean training error values were achieved with the MPA. In S1, the best training error value after the MPA was found with the DO. The error values found with the MPA and DO are 1.6×10^{-6} and 1.3×10^{-4}, respectively. The only result at the 10^{-6} level in S1 belongs to the MPA. Mean error values at the 10^{-4} level were found with the FPA, MVO, DO, and the RIME algorithm. The second successful algorithm in S2 is the CryStAl. Its error value is 2.5×10^{-3}. MFO is the second successful algorithm in S3. The values of MFO and the MPA in S3 are 5.6×10^{-5} and 1.5×10^{-5}, respectively. In S4, the MPA achieved a very successful mean error value compared to other algorithms. The value of the MPA is 9.2×10^{-6}. The MVO is the second successful algorithm in S4. Except for BSO, average error values at 10^{-5} level were reached in S5. The two most successful algorithms on S5 are the MPA and MFO, respectively. Except for BSO and the CryStAl, error values at 10^{-3} level were reached in S6. In S7, the MPA was very effective compared to other algorithms. The error value of the MPA is 1.7×10^{-5}. The most successful algorithm after the MPA is RIME for S7. The mean error value of the MPA in S8 is 8.7×10^{-6}. MFO ranks second in S8.

Table 3. Testing process results of ANFIS training carried out with MPA for identification of nonlinear systems.

System	Number of MFs	Best (MSE)	Mean (MSE)	Worst (MSE)	Standard Deviation
S1	2	7.9×10^{-7}	6.5×10^{-6}	1.7×10^{-4}	3.0×10^{-5}
	3	9.4×10^{-8}	2.2×10^{-6}	1.1×10^{-5}	2.3×10^{-6}
	4	1.7×10^{-7}	6.4×10^{-5}	1.2×10^{-3}	2.4×10^{-4}
S2	2	2.0×10^{-2}	6.6×10^{-2}	5.1×10^{-1}	1.1×10^{-1}
	3	7.0×10^{-3}	3.0×10^{-2}	2.3×10^{-1}	4.2×10^{-2}
	4	4.7×10^{-3}	4.9×10^{-2}	2.2×10^{-1}	6.2×10^{-2}
S3	2	8.6×10^{-5}	4.7×10^{-4}	3.2×10^{-3}	6.5×10^{-4}
	3	4.3×10^{-5}	8.5×10^{-4}	6.0×10^{-3}	1.5×10^{-3}
	4	1.3×10^{-4}	1.7×10^{-3}	1.7×10^{-2}	3.2×10^{-3}
S4	2	3.8×10^{-6}	1.5×10^{-4}	1.3×10^{-3}	2.4×10^{-4}
	3	5.6×10^{-6}	1.8×10^{-3}	4.7×10^{-2}	8.5×10^{-3}
	4	5.3×10^{-6}	7.8×10^{-4}	1.2×10^{-2}	2.3×10^{-3}
S5	2	3.3×10^{-5}	2.3×10^{-4}	1.4×10^{-3}	2.5×10^{-4}
	3	7.3×10^{-5}	2.3×10^{-4}	1.1×10^{-3}	1.8×10^{-4}
	4	2.9×10^{-5}	1.6×10^{-4}	4.4×10^{-4}	8.1×10^{-5}
S6	2	2.0×10^{-3}	3.4×10^{-3}	4.5×10^{-3}	6.9×10^{-4}
	3	1.0×10^{-3}	4.0×10^{-3}	1.1×10^{-2}	2.3×10^{-3}
	4	8.7×10^{-4}	4.8×10^{-3}	2.8×10^{-2}	5.4×10^{-3}
S7	2	6.5×10^{-6}	2.2×10^{-4}	2.0×10^{-3}	4.7×10^{-4}
	3	6.2×10^{-6}	2.4×10^{-3}	2.0×10^{-2}	5.5×10^{-3}
	4	8.6×10^{-6}	1.3×10^{-4}	7.4×10^{-4}	1.6×10^{-4}
S8	2	5.2×10^{-6}	4.6×10^{-4}	3.4×10^{-3}	6.2×10^{-4}
	3	4.7×10^{-6}	3.0×10^{-4}	3.6×10^{-3}	6.4×10^{-4}
	4	5.7×10^{-6}	3.6×10^{-4}	4.7×10^{-3}	8.6×10^{-4}

Table 4. Information belonging to the best training mean errors.

System	Number of MFs	Train Mean	Test Mean
S1	3	1.6×10^{-6}	2.2×10^{-6}
S2	4	3.2×10^{-3}	4.9×10^{-2}
S3	3	1.5×10^{-5}	8.5×10^{-4}
S4	4	9.2×10^{-6}	7.8×10^{-4}
S5	3	3.2×10^{-5}	2.3×10^{-4}
S6	4	2.3×10^{-3}	4.8×10^{-3}
S7	4	1.7×10^{-5}	1.3×10^{-4}
S8	4	8.7×10^{-6}	3.6×10^{-4}

Table 5. Comparison of training error values obtained with MPA and different meta-heuristic algorithms (best results are given in bold).

System	BSO	FPA	MFO	MVO	SSA	CryStAl	DO	RIME	Proposed (MPA)
S1	1.7×10^{-2}	3.2×10^{-4}	5.5×10^{-3}	4.5×10^{-4}	1.7×10^{-3}	2.5×10^{-3}	1.3×10^{-4}	7.7×10^{-4}	$\mathbf{1.6 \times 10^{-6}}$
S2	1.2×10^{-1}	8.2×10^{-3}	1.1×10^{-2}	5.7×10^{-3}	7.7×10^{-3}	5.4×10^{-2}	6.9×10^{-3}	5.6×10^{-3}	$\mathbf{3.2 \times 10^{-3}}$
S3	1.6×10^{-3}	1.0×10^{-4}	5.6×10^{-5}	6.1×10^{-5}	8.0×10^{-5}	4.0×10^{-4}	1.1×10^{-4}	6.9×10^{-5}	$\mathbf{1.5 \times 10^{-5}}$
S4	1.2×10^{-3}	1.4×10^{-4}	9.2×10^{-5}	8.0×10^{-5}	1.0×10^{-4}	4.2×10^{-4}	1.2×10^{-4}	1.1×10^{-4}	$\mathbf{9.2 \times 10^{-6}}$
S5	1.6×10^{-4}	6.2×10^{-5}	5.9×10^{-5}	6.6×10^{-5}	6.6×10^{-5}	7.3×10^{-5}	6.8×10^{-5}	6.7×10^{-5}	$\mathbf{3.2 \times 10^{-5}}$
S6	3.3×10^{-2}	4.7×10^{-3}	5.0×10^{-3}	4.2×10^{-3}	5.5×10^{-3}	1.7×10^{-2}	4.6×10^{-3}	3.0×10^{-3}	$\mathbf{2.3 \times 10^{-3}}$
S7	2.2×10^{-3}	1.9×10^{-4}	1.8×10^{-4}	2.4×10^{-4}	2.6×10^{-4}	6.5×10^{-4}	2.6×10^{-4}	1.4×10^{-4}	$\mathbf{1.7 \times 10^{-5}}$
S8	1.1×10^{-3}	4.5×10^{-5}	2.5×10^{-5}	5.5×10^{-5}	3.0×10^{-5}	1.4×10^{-4}	4.2×10^{-5}	4.6×10^{-5}	$\mathbf{8.7 \times 10^{-6}}$

In Table 6, the testing performance of the MPA is compared with BSO, FPA, MFO, MVO SSA, CryStAl, DO and RIME. The population size of all algorithms was taken as 20. The maximum number of iterations is 2500. For each algorithm, the MF information given in Table 4 was used. The best mean test results in S1, S3, S6, and S7 belong to the MPA. The error value of the MPA in S1 is at the 10^{-6} level and high success has been achieved. The second successful algorithm for S1 is the FPA. Results found with the MPA and FPA are 2.2×10^{-6} and 2.6×10^{-4}, respectively. The best mean error value with S2 is 3.1×10^{-2} and it was found with MFO. The most effective algorithm after MFO in S2 is RIME. In S3, the MPA was more successful than other algorithms. The error value of the MPA is 8.5×10^{-4}. The second successful algorithm for S3 is MFO. The error value of MFO is 1.1×10^{-3}. The two most successful algorithms in S4 are MFO and the MPA. The mean error values of these algorithms are 5.0×10^{-4} and 7.8×10^{-4}, respectively. The most successful algorithm in S5 is MFO. Its error value is 2.1×10^{-4}. Three algorithms share the second place in S5. These algorithms are the DO, RIME, and the MPA. The error value of the MPA in S6 is 4.8×10^{-3}. After the MPA for S6, the next most successful algorithm is RIME, and the mean error value was found to be 5.8×10^{-3} using RIME. The top two most successful algorithms on S7 are the MPA and FPA. Their error values are 1.3×10^{-4} and 1.8×10^{-3}, respectively. While the most successful algorithm in S8 was MFO, the MPA ranked third.

Table 6. Comparison of testing error values obtained with MPA and different meta-heuristic algorithms (best results are given in bold).

System	BSO	FPA	MFO	MVO	SSA	CryStAl	DO	RIME	Proposed (MPA)
S1	1.5×10^{-2}	2.6×10^{-4}	3.5×10^{-3}	5.0×10^{-4}	9.2×10^{-4}	1.5×10^{-3}	1.7×10^{-4}	5.8×10^{-4}	$\mathbf{2.2 \times 10^{-6}}$
S2	9.8×10^{-2}	6.1×10^{-2}	$\mathbf{3.1 \times 10^{-2}}$	7.6×10^{-2}	3.7×10^{-2}	8.1×10^{-2}	4.1×10^{-2}	3.4×10^{-2}	4.9×10^{-2}
S3	6.8×10^{-3}	1.6×10^{-3}	1.1×10^{-3}	2.6×10^{-3}	3.6×10^{-3}	2.6×10^{-3}	2.1×10^{-3}	2.9×10^{-3}	$\mathbf{8.5 \times 10^{-4}}$
S4	3.1×10^{-3}	1.4×10^{-3}	$\mathbf{5.0 \times 10^{-4}}$	1.1×10^{-2}	1.3×10^{-3}	5.2×10^{-3}	2.4×10^{-3}	3.6×10^{-3}	7.8×10^{-4}
S5	2.7×10^{-4}	2.5×10^{-4}	$\mathbf{2.1 \times 10^{-4}}$	6.3×10^{-4}	2.9×10^{-4}	2.9×10^{-4}	2.3×10^{-4}	2.3×10^{-4}	2.3×10^{-4}
S6	7.9×10^{-3}	6.5×10^{-3}	1.2×10^{-2}	7.0×10^{-3}	7.2×10^{-3}	1.7×10^{-2}	1.4×10^{-2}	5.8×10^{-3}	$\mathbf{4.8 \times 10^{-3}}$
S7	8.6×10^{-3}	1.8×10^{-3}	4.7×10^{-3}	2.2×10^{-3}	2.0×10^{-3}	2.6×10^{-3}	2.1×10^{-3}	2.5×10^{-3}	$\mathbf{1.3 \times 10^{-4}}$
S8	1.7×10^{-3}	1.2×10^{-3}	$\mathbf{2.2 \times 10^{-4}}$	5.1×10^{-4}	3.4×10^{-4}	8.4×10^{-4}	4.3×10^{-4}	3.1×10^{-3}	3.6×10^{-4}

The performances of the algorithms are compared in Tables 5 and 6. The comparison of the success scores obtained according to these performances is presented in Table 7. The MPA's success ranking score is 24. According to the training and testing processes, it is generally more successful than other algorithms. Algorithms other than BSO and the

CryStAl achieved a total success score in the range of 60–80. After the MPA, the most successful algorithm according to total score is MFO. MFO is followed by the FPA and RIME with a success score of 72. The DO's rank is very close to the FPA and the RIME. Its value is 73. The total score of the SSA is 77. The MVO follows SSA with a total score of 79. BSO is the most unsuccessful algorithm in both training and testing processes.

Table 7. General success scores of the metaheuristic algorithms.

System	BSO		FPA		MFO		MVO		SSA		CryStAl		DO		RIME		Proposed (MPA)	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
S1	9	9	3	3	8	8	4	4	6	6	7	7	2	2	5	5	1	1
S2	9	9	6	6	7	1	3	7	5	3	8	8	4	4	2	2	1	5
S3	9	9	6	3	2	2	3	5	5	8	8	5	7	4	4	7	1	1
S4	9	6	7	4	3	1	2	9	4	3	8	8	6	5	5	7	1	2
S5	9	6	3	5	2	1	4	9	4	7	8	7	7	2	6	2	1	2
S6	9	6	5	3	6	7	3	4	7	5	8	9	4	8	2	2	1	1
S7	9	9	4	2	3	8	5	5	6	3	8	7	6	4	2	6	1	1
S8	9	8	5	7	2	1	7	5	3	2	8	6	4	4	6	9	1	3
TOTAL	72	62	39	33	33	29	31	48	40	37	63	57	40	33	32	40	8	16
	134		72		62		79		77		120		73		72		24	

The graphs of the actual output and the predicted output were compared to evaluate the success of the training process, and this is presented in Figure 2. These graphics were created taking into account the best results of the MPA. In S1, S3, S4, S7, and S8, the real and predicted outputs overlap exactly. In other systems, there is little difference. This shows that the training process of the MPA is successful.

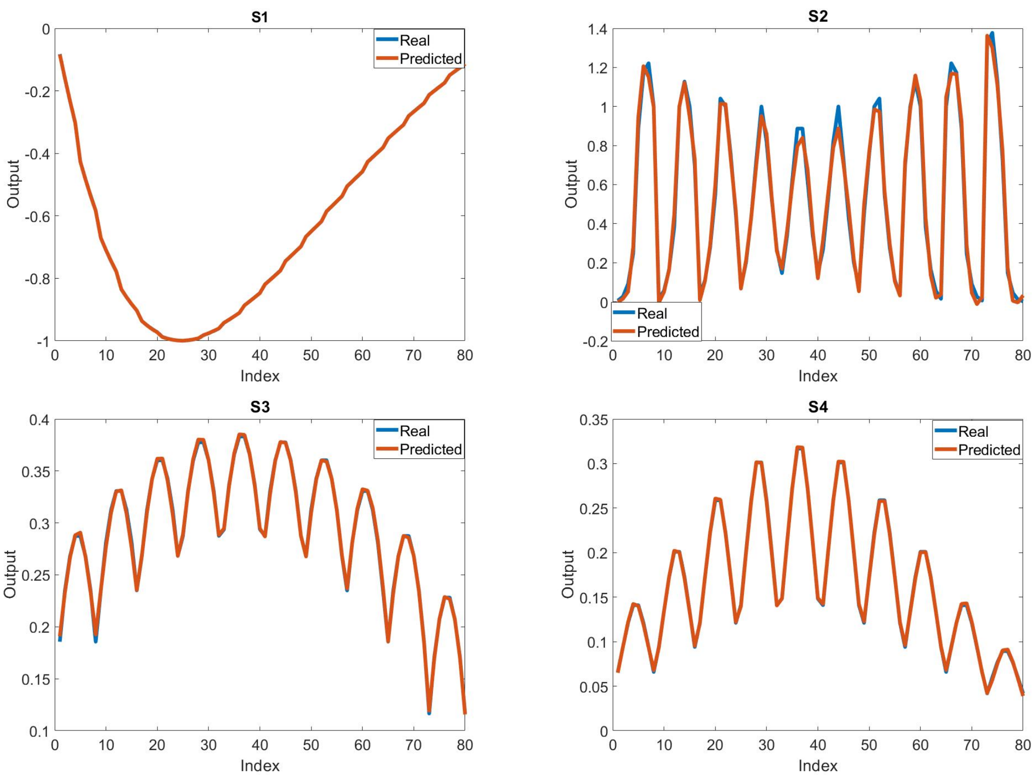

Figure 2. *Cont.*

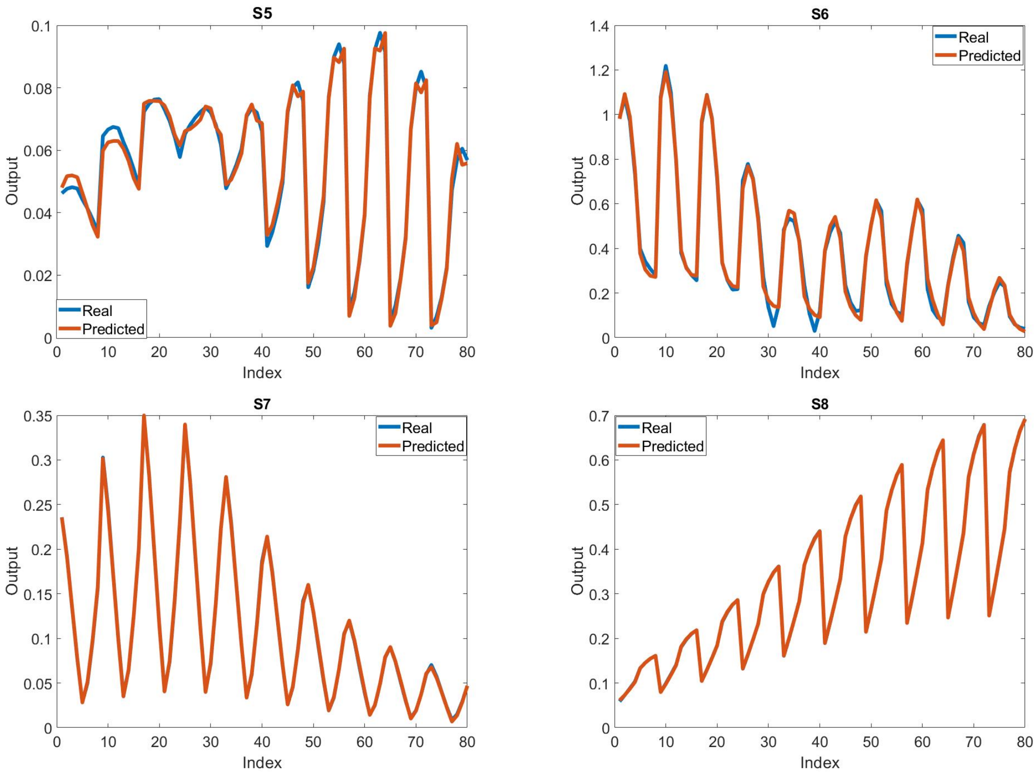

Figure 2. Comparison of graphs belonging to real and predicted outputs for training process.

4. Discussion

MF type, number of MFs, control parameters of the training algorithm, and the system to be modeled are important factors affecting performance in ANFIS training. In this study, the generalized Bell function, which is used extensively in the literature, was used. The results were obtained by using 2, 3, and 4 MFs for each input in each system. The effect of increasing or decreasing the number of MFs varies according to the systems. It also affected training and test results differently. Mostly, the increase in the number of MFs improved the performance in the training process. This situation is clearly seen in the S2, S4, S6, S7, and S8 systems. Especially when looking at the best mean error values, it can be seen that it is not found with two MFs.

In ANFIS training, the testing process is as important as the training process. As seen in Table 4, training error values and test error values show parallelism with each other. In other words, both the test results and the training results are successful.

The general structure of nonlinear systems directly affects the results. In some systems, mean training error values at the level of 10^{-3} were reached, whereas in some systems, mean training error values at the level of 10^{-6} were reached. It is also important in this context whether nonlinear systems exhibit symmetric behavior or not. When we look at the systems in general, it can be observed that they have an asymmetric structure. However, some systems exhibit regional symmetry.

Standard deviation is one of the important indicators. Each application was run 30 times and standard deviation values were calculated. As can be seen in Tables 2 and 3, low standard deviation values were obtained in parallel with the error values. This shows that the results are reproducible. Continuously achieving successful results in a process that starts with random solutions increases the confidence in the training algorithm. The standard deviation values obtained are an indication that the MPA is effective.

One of the most important ways to determine whether a training algorithm is successful is to compare it with different training algorithms. For this reason, the performance of the MPA has been compared with meta-heuristic algorithms such as BSO, FPA, MFO, MVO, SSA, CryStAl, DO, and RIME. Especially in the training process of all systems, the MPA is more successful than other algorithms. This supports the conclusion that the MPA is a powerful training algorithm. Likewise, the MPA achieved significant success in the test results.

It is apparent that the proposed method is generally successful in solving the relevant problem. When all the results are evaluated, the proposed method has some advantages and disadvantages.

- The results show that the proposed method is successful on different systems. The characteristics of each system are different. Some of them can be very difficult to work with, and some of them present easier problems. In contrast, the MPA has produced successful results on all types of systems. To analyze this better, the results obtained with other meta-heuristic algorithms should be looked at. For example, while MFO is very successful in identifying some systems in both training and testing processes, it is also very unsuccessful in some systems. In contrast, the MPA has generally consistent success across all systems. This is an important advantage.
- Another important advantage of the proposed method is that it is also successful in the training and testing process. In the training process, the training algorithm learns by seeing the dataset. But during the test process, it gives results according to the data that it does not know at all. As seen in Table 7, the MPA is successful in both training and testing processes and ranks first. The success of the MPA parallels the training and testing process. In other algorithms, this stability cannot be observed in general. The behavior of other algorithms may vary depending on the system. In fact, this shows that the MPA can be successful in solving different problems.
- A disadvantage of the proposed method is that the best results were achieved by trial and error. In fact, this disadvantage arises from the ANFIS model. The MFs and number of MFs used in the ANFIS model affect the result. As can be seen in the results, more successful results were achieved with three MFs in some systems, whereas using four MFs was more successful in some systems. Furthermore, while increasing the number of MFs improves performance in some systems, this is not observed in others. The best successful model for each system must be found by trial and error. This appears as a disadvantage. Despite this disadvantage, successful results can be achieved in solving problems with the strong structure of the ANFIS model.

5. Conclusions

In this study, ANFIS training was carried out using the MPA for the identification of nonlinear systems. Applications were performed on eight nonlinear systems. Performance analysis was performed on different ANFIS models for each system. The performance of the MPA is compared with different meta-heuristic algorithms. The main conclusions of this study are as follows:

- It was observed that the performance of the MPA was successful in the ANFIS training carried out for the identification of nonlinear systems.
- It has been observed that the performance of the MPA changes according to the structure of the systems used.
- Low standard deviation values were obtained. This indicates that the results are reproducible.
- It has been seen that the performance of the MPA is better than meta-heuristic algorithms such as BSO, FPA, MFO, MVO, SSA, CryStAl, DO, and RIME for solving related problems.
- The change in the number of MFs affects performance. The impact rate varies according to the system used and the training algorithm.

In this study, the success, reproducibility, and strong structure of the MPA show that it can also provide successful results in solving different problems in the fields of economics, education, social sciences, health sciences, engineering, and energy in the future.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The author declares no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MPA	Marine predators algorithm
BOA	Butterfly optimization algorithm
FPA	Flower pollination algorithm
MFO	Moth–flame optimization
MVO	Multi-verse optimizer
SSA	Salp swarm algorithm
FA	Firefly algorithm
CA	Cultural algorithm
BA	Bees algorithm
IWO	Invasive weed optimization
SBO	Satin bowerbird optimizer
TLBO	Teaching–learning-based optimization
AO	Aquila optimizer
ABC	Artificial bee colony
PSO	Particle swarm optimization
HS	Harmony search
CS	Cuckoo search
DE	Differential evolution
BSA	Backtracking search algorithm
GA	Genetic algorithm
GWO	Gray wolf optimization
KH	Krill Herd
SVM	Support vector machine
FFNN	Feed-forward neural network
LSTM	Long short-term memory
GOA	Grasshopper optimization algorithm
EO	Equilibrium optimizer
WOA	Whale optimization algorithm
DSA	Differential search algorithm
LCA	League championship algorithm
CryStAl	Crystal structure algorithm
DO	Dandelion optimizer
RIME	RIME algorithm
CNN	Convolutional neural network
ANFIS	Adaptive Network Fuzzy Inference System

References

1. Karaboga, D.; Kaya, E. Adaptive network based fuzzy inference system (ANFIS) training approaches: A comprehensive survey. *Artif. Intell. Rev.* **2019**, *52*, 2263–2293. [CrossRef]
2. Yaseen, Z.M.; Ebtehaj, I.; Bonakdari, H.; Deo, R.C.; Mehr, A.D.; Mohtar, W.H.M.W.; Diop, L.; El-Shafie, A.; Singh, V.P. Novel approach for streamflow forecasting using a hybrid ANFIS-FFA model. *J. Hydrol.* **2017**, *554*, 263–276. [CrossRef]
3. Tien Bui, D.; Khosravi, K.; Li, S.; Shahabi, H.; Panahi, M.; Singh, V.P.; Chapi, K.; Shirzadi, A.; Panahi, S.; Chen, W. New hybrids of anfis with several optimization algorithms for flood susceptibility modeling. *Water* **2018**, *10*, 1210. [CrossRef]
4. Chen, W.; Chen, X.; Peng, J.; Panahi, M.; Lee, S. Landslide susceptibility modeling based on ANFIS with teaching-learning-based optimization and Satin bowerbird optimizer. *Geosci. Front.* **2021**, *12*, 93–107. [CrossRef]

5. Al Rassas, A.M.; Al-Qaness, M.A.; Ewees, A.A.; Ren, S.; Abd Elaziz, M.; Damaševičius, R.; Krilavičius, T. Optimized ANFIS model using Aquila Optimizer for oil production forecasting. *Processes* **2021**, *9*, 1194. [CrossRef]
6. Karaboga, D.; Kaya, E. An adaptive and hybrid artificial bee colony algorithm (aABC) for ANFIS training. *Appl. Soft Comput.* **2016**, *49*, 423–436. [CrossRef]
7. Pourdaryaei, A.; Mokhlis, H.; Illias, H.A.; Kaboli, S.H.A.; Ahmad, S. Short-term electricity price forecasting via hybrid backtracking search algorithm and ANFIS approach. *IEEE Access* **2019**, *7*, 77674–77691. [CrossRef]
8. Riahi-Madvar, H.; Dehghani, M.; Memarzadeh, R.; Gharabaghi, B. Short to long-term forecasting of river flows by heuristic optimization algorithms hybridized with ANFIS. *Water Resour. Manag.* **2021**, *35*, 1149–1166. [CrossRef]
9. Ahmed, K.; Ewees, A.A.; El Aziz, M.A.; Hassanien, A.E.; Gaber, T.; Tsai, P.-W.; Pan, J.-S. A hybrid krill-ANFIS model for wind speed forecasting. In Proceedings of the International Conference on Advanced Intelligent Systems and Informatics, Cairo, Egypt, 24–26 November 2016; pp. 365–372. [CrossRef]
10. Liu, K.; Yang, P.; Wang, R.; Jiao, L.; Li, T.; Zhang, J. Observer-Based Adaptive Fuzzy Finite-Time Attitude Control for Quadrotor UAVs. *IEEE Trans. Aerosp. Electron. Syst.* **2023**. [CrossRef]
11. Al-Betar, M.A.; Awadallah, M.A.; Makhadmeh, S.N.; Alyasseri, Z.A.A.; Al-Naymat, G.; Mirjalili, S. Marine Predators Algorithm: A Review. *Arch. Comput. Methods Eng.* **2023**, *30*, 3405–3435. [CrossRef]
12. Abd Elminaam, D.S.; Nabil, A.; Ibraheem, S.A.; Houssein, E.H. An efficient marine predators algorithm for feature selection. *IEEE Access* **2021**, *9*, 60136–60153. [CrossRef]
13. Abdel-Basset, M.; El-Shahat, D.; Chakrabortty, R.K.; Ryan, M. Parameter estimation of photovoltaic models using an improved marine predators algorithm. *Energy Convers. Manag.* **2021**, *227*, 113491. [CrossRef]
14. Soliman, M.A.; Hasanien, H.M.; Alkuhayli, A. Marine predators algorithm for parameters identification of triple-diode photovoltaic models. *IEEE Access* **2020**, *8*, 155832–155842. [CrossRef]
15. Abd Elaziz, M.; Thanikanti, S.B.; Ibrahim, I.A.; Lu, S.; Nastasi, B.; Alotaibi, M.A.; Hossain, M.A.; Yousri, D. Enhanced marine predators algorithm for identifying static and dynamic photovoltaic models parameters. *Energy Convers. Manag.* **2021**, *236*, 113971. [CrossRef]
16. Zhong, K.; Zhou, G.; Deng, W.; Zhou, Y.; Luo, Q. MOMPA: Multi-objective marine predator algorithm. *Comput. Methods Appl. Mech. Eng.* **2021**, *385*, 114029. [CrossRef]
17. Fan, Q.; Huang, H.; Chen, Q.; Yao, L.; Yang, K.; Huang, D. A modified self-adaptive marine predators algorithm: Framework and engineering applications. *Eng. Comput.* **2022**, *38*, 3269–3294. [CrossRef]
18. Al-qaness, M.A.; Ewees, A.A.; Fan, H.; Abualigah, L.; Abd Elaziz, M. Boosted ANFIS model using augmented marine predator algorithm with mutation operators for wind power forecasting. *Appl. Energy* **2022**, *314*, 118851. [CrossRef]
19. Al-Qaness, M.A.; Ewees, A.A.; Fan, H.; Abualigah, L.; Abd Elaziz, M. Marine predators algorithm for forecasting confirmed cases of COVID-19 in Italy, USA, Iran and Korea. *Int. J. Environ. Res. Public Health* **2020**, *17*, 3520. [CrossRef]
20. Shaheen, M.A.; Yousri, D.; Fathy, A.; Hasanien, H.M.; Alkuhayli, A.; Muyeen, S. A novel application of improved marine predators algorithm and particle swarm optimization for solving the ORPD problem. *Energies* **2020**, *13*, 5679. [CrossRef]
21. Houssein, E.H.; Abdelminaam, D.S.; Ibrahim, I.E.; Hassaballah, M.; Wazery, Y.M. A hybrid heartbeats classification approach based on marine predators algorithm and convolution neural networks. *IEEE Access* **2021**, *9*, 86194–86206. [CrossRef]
22. Ikram, R.M.A.; Cao, X.; Sadeghifar, T.; Kuriqi, A.; Kisi, O.; Shahid, S. Improving Significant Wave Height Prediction Using a Neuro-Fuzzy Approach and Marine Predators Algorithm. *J. Mar. Sci. Eng.* **2023**, *11*, 1163. [CrossRef]
23. Al-Qaness, M.A.; Saba, A.I.; Elsheikh, A.H.; Abd Elaziz, M.; Ibrahim, R.A.; Lu, S.; Hemedan, A.A.; Shanmugan, S.; Ewees, A.A. Efficient artificial intelligence forecasting models for COVID-19 outbreak in Russia and Brazil. *Process Saf. Environ. Prot.* **2021**, *149*, 399–409. [CrossRef] [PubMed]
24. Ghomsheh, V.S.; Shoorehdeli, M.A.; Teshnehlab, M. Training ANFIS structure with modified PSO algorithm. In Proceedings of the 2007 Mediterranean Conference on Control & Automation 2007, Athens, Greece, 27–29 June 2007; pp. 1–6. [CrossRef]
25. Karaboğa, D.; Kaya, E. Training ANFIS by using the artificial bee colony algorithm. *Turk. J. Electr. Eng. Comput. Sci.* **2017**, *25*, 1669–1679. [CrossRef]
26. Zangeneh, A.Z.; Mansouri, M.; Teshnehlab, M.; Sedigh, A.K. Training ANFIS system with DE algorithm. In Proceedings of the Fourth International Workshop on Advanced Computational Intelligence, Wuhan, China, 19–21 October 2011; pp. 308–314. [CrossRef]
27. Haznedar, B.; Kalinli, A. Training ANFIS structure using simulated annealing algorithm for dynamic systems identification. *Neurocomputing* **2018**, *302*, 66–74. [CrossRef]
28. Haznedar, B.; Kalinli, A. Training ANFIS using genetic algorithm for dynamic systems identification. *Int. J. Intell. Syst. Appl. Eng.* **2016**, *4*, 44–47. [CrossRef]
29. Marzi, H.; Haj Darwish, A.; Helfawi, H. Training ANFIS using the enhanced Bees Algorithm and least squares estimation. *Intell. Autom. Soft Comput.* **2017**, *23*, 227–234. [CrossRef]
30. Canayaz, M. Training anfis system with moth-flame optimization algorithm. *Int. J. Intell. Syst. Appl. Eng.* **2019**, *7*, 133–144. [CrossRef]
31. Arora, S.; Singh, S. Butterfly optimization algorithm: A novel approach for global optimization. *Soft Comput.* **2019**, *23*, 715–734. [CrossRef]
32. Yang, X.-S. Flower pollination algorithm for global optimization. In Proceedings of the International Conference on Unconventional Computing and Natural Computation, Orléan, France, 3–7 September 2012; pp. 240–249. [CrossRef]

33. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl. Based Syst.* **2015**, *89*, 228–249. [CrossRef]
34. Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **2016**, *27*, 495–513. [CrossRef]
35. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [CrossRef]
36. Talatahari, S.; Azizi, M.; Tolouei, M.; Talatahari, B.; Sareh, P. Crystal structure algorithm (CryStAl): A metaheuristic optimization method. *IEEE Access* **2021**, *9*, 71244–71261. [CrossRef]
37. Zhao, S.; Zhang, T.; Ma, S.; Chen, M. Dandelion Optimizer: A nature-inspired metaheuristic algorithm for engineering applications. *Eng. Appl. Artif. Intell.* **2022**, *114*, 105075. [CrossRef]
38. Su, H.; Zhao, D.; Heidari, A.A.; Liu, L.; Zhang, X.; Mafarja, M.; Chen, H. RIME: A physics-based optimization. *Neurocomputing* **2023**, *532*, 183–214. [CrossRef]
39. Faramarzi, A.; Heidarinejad, M.; Mirjalili, S.; Gandomi, A.H. Marine Predators Algorithm: A nature-inspired metaheuristic. *Expert Syst. Appl.* **2020**, *152*, 113377. [CrossRef]
40. Jang, J.-S. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Trans. Syst. Man Cybern.* **1993**, *23*, 665–685. [CrossRef]