International Journal of
Intelligent Systems and Applications in Engineering

ISSN:2147-6799 www.ijisae.org Original Research Paper

Training Anfis System with Moth-Flame Optimization Algorithm

Murat Canayaz[1*]

Submitted: 05/05/2019 Accepted : 13/09/2019

Abstract: Adaptive Neuro Fuzzy Inference System (ANFIS) is an adaptive network that can use the computation and learning abilities of artificial neural network together with the inference feature of fuzzy logic. The ANFIS system, which is used in the solution of many problems such as classification and estimation of deep learning applications, meets the needs in many different areas such as modeling, control, and parameter estimation. In recent years, heuristic methods have been used for the training of this network, which requires initial and result parameters by its structure. Moth-Flame Optimization Algorithm (MFO) is one of the current heuristic methods modeled by the influence of the spiral movement of the moths towards the light source. In this study, the MFO algorithm was used for the first time for the optimization of initial and result parameters in the ANFIS system. In the determination of parameters, nonlinear system identification, time series estimation, classification problems were tried to be solved. When the results obtained for the ANFIS trained with the known heuristic methods such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA) and Whale Optimization Algorithm (WOA) and the results of ANFIS trained by the MFO were examined, it was observed that the MFO had lower error values.

Keywords: *ANFIS, classification, machine learning, moth-flame optimization, neural computing*

1. Introduction

Nowadays during which high-dimensional data are obtained, data mining methods are used to remove unnecessary data and obtain meaningful data. In recent years, the concepts such as machine learning, deep learning, and artificial intelligence have become frequently encountered terms in our lives. Artificial neural networks are the computing systems created by being inspired by the human nervous system, which are included among these concepts. Artificial neural networks are included in the literature as computing mechanisms used in many fields such as classification, estimation and modelling. The computing process that starts by the weighting of the given input values sends a value to the output if the processing elements called neurons transmit to each other via activation functions.

The concept of fuzzy logic modelling was first introduced by Lütfi A. Zadeh [1]. The relations between the concepts in fuzzy logic are represented by quantitative expressions in verbal or numerical structure [2]. In the principles of fuzzy logic, a) approximate values should be used instead of exact values, b) everything should be expressed by the values between 0-1, and c) data should be processed as intermediate values. System inputs are converted to desired outputs with the rules determined by fuzzy sets created based on these principles. Inputs go through the fuzzification, control and defuzzification phases, respectively.

ANFIS are decision-making mechanisms that integrate both neural networks and fuzzy logic inference methods. A data set based on input-output is needed for the applicability of the ANFIS method. The model created using a learning algorithm depends on the type and number of the membership function selected. ANFIS uses the set of fuzzy if-then rules created by it. In building the ANFIS architecture, the determination of parameters is based on in a way to minimize the difference between the output of the entire network and the target value, in other words, the error. In recent years, heuristic methods have been used in the determination of appropriate parameter values. In these methods, parameter values are initially taken as candidate solutions. Candidate solutions are updated in a way to provide the minimum error according to each method's unique update process. At the end of the method, ANFIS training is completed and the problem is solved with the best values.

In this study, the use of Moth-Flame Optimization (MFO) method [3], which is one of the current heuristic methods, in the solution of various problems, such as classification, nonlinear system identification, and estimation, using in ANFIS training will be presented. The rest of this study was organized as follows: Literature information about both ANFIS and MFO method will be given in the second section. In the third and fourth sections, MFO and ANFIS structures will be introduced, respectively. In the final section, the data sets and problems used, and the results obtained from ANFIS trained with MFO will be comparatively given with the PSO [4], GA [5], WOA [6] algorithms.

2. Related Work

MFO has been used in many fields such as medicine, economy, engineering, and agriculture due to its simple structure that does not require more parameters. In their study, Yamanny Waled et al. (2015) [7] used MFO in investigating the weight and bias values of Multilayer Perceptron to obtain the minimum error and high classification ratio. Hassanien, AboulElla, et al. (2017) [8] used MFO based on rough sets for tomato disease detection. Sayed (2016) [9] also included MFO in swarm intelligence algorithms he used for biomedical segmentation problems. El Aziz (2017) [10] used MFO to find the optimum threshold value in order to remove the time consumption problem in multilevel thresholding problems. Zawbaa (2016) [11] performed the attribute selection process, which plays an important role in machine learning, by means of MFO. Lal (2016) [12] used MFO to determine the

[1] *Computer Engineering Department, Engineering Faculty, Van Yuzuncu Yil University, Van, Turkey*
ORCID ID : 0000-0001-8120-5101
** Corresponding Author Email: mcanayaz@yyu.edu.tr*

controller parameters. In the study of Jangir (2016) [13], used MFO in the solution of the emission constrained economic dispatch

(ECED) problem, which is called as an allowable deviation in fuel cost and feasible tolerance in fuel cost. In the study in which MFO and Simulated Angeling were used together, Sayed (2018) [14] tried to solve the benchmark functions and the known engineering problems.

It is possible to say that the PSO algorithm is generally used in the publications related to the training of ANFIS. Only parameter learning was performed in the publications in which ANFIS based estimation models were developed for the estimation of electricity costs, wind energy and customer happiness for a new product [15-19]. Turki (2012) [20] used the ANFIS trained by PSO for nonlinear system adaptive control. Haznedar (2016) [21] performed the classification of Liver microarray cancer data with the ANFIS trained using GA. Furthermore, in another study of Haznedar et al. (2016) [22], the ANFIS trained by GA was also used in dynamic system identification. In their study, Karaboga et al. (2014,2016) [23, 24] used the ANFIS trained with Artificial Bee Colony algorithm in nonlinear system identification. Thaganvel et al. (2016) [25] used the ANFIS architecture with parameter learning by using the Ant Colony Optimizer, for mammogram classification. Canayaz et al. (2018) [26] used the ANFIS trained with WOA for nonlinear system identification with ANFIS and time series estimation problem.

ANFIS training with MFO was not found in the literature search. For this reason, since this study carried out by us is the first study, it was aimed to give an idea about the performance of the ANFIS trained with MFO.

3. Moth-Flame Optimization Algorithm

It is a meta-heuristic method developed by Mirjalili (2016) [3] based on the navigation in nature approach of moths which is called transverse orientation. In this method, candidate solutions are considered as the moths and their locations in space are considered as problem variables.

In this method, a moth flies by providing a constant angle relative to the moon. Since there are many artificial light sources outside the moon in nature, the moths turn towards all light sources they see at the same angle [27]. Although the flames in this method are candidate solutions like moths, unlike them, flames are considered as the best solution.

Flames can be thought of as flags or pins that are dropped by moths when searching the search area. Therefore, each moth searches for a flag (flame) and updates it in case of a better solution [3]. The logarithmic spiral movement equation is used while candidate solutions are updated. The movement of the moths in MFO around the light source is represented as in Figure 1.

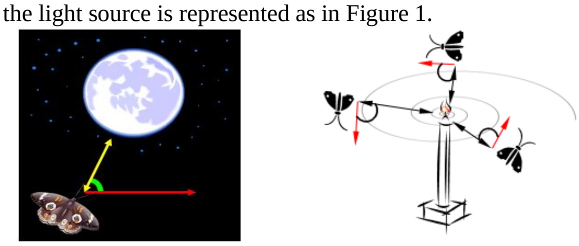


Fig. 1: Moths movement

The moths in the method are represented in the following matrix form.

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & \cdots & m_{1,d} \\ m_{2,1} & m_{2,2} & \cdots & \cdots & m_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{n,1} & m_{n,2} & \cdots & \cdots & m_{n,d} \end{bmatrix}$$

where n is the number of moths and d is the number of variables (dimension).

The fitness values are stored in a matrix for all the moths (OM).

$$OM = \begin{bmatrix} OM_1 \\ OM_2 \\ \vdots \\ OM_n \end{bmatrix}$$

where n is the number of moths.

The same matrix form also applies to the flames.

$$F = \begin{bmatrix} F_{1,1} & F_{1,2} & \cdots & \cdots & F_{1,d} \\ F_{2,1} & F_{2,2} & \cdots & \cdots & F_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ F_{n,1} & m_{n,2} & \cdots & \cdots & F_{n,d} \end{bmatrix}$$

where n is the number of moths and d is the number of variables (dimension).

The fitness matrix for flames is represented as follows (OF).

$$OF = \begin{bmatrix} OF_1 \\ OF_2 \\ \vdots \\ OF_n \end{bmatrix}$$

where n is the number of flames

MFO has a structure specified by three main functions such as I, P, T. When these are explained briefly;

3.1. I function

The task of this function is to ensure that the initial population is produced by complying with the given lower and upper limits. The first fitness function values for these population values are also calculated here.

$$S(M_i, F_j) = D_i.e^{bt}.\cos(2\pi t) + F_j \tag{1}$$

3.2. P function

This function is the main function where the logarithmic spiral movement that is required for moths to update their positions while moving around the search space is performed. The equation used for the spiral movement is given in 1. The movement pattern is shown in Figure 2.

$$S(M_i, F_j) = D_i.e^{bt}.\cos(2\pi t) + F_j \tag{1}$$

D is calculated as follows:

$$D_i = \left| F_j - M_i \right| \tag{2}$$

where, M_i shows the i-th moth, F_j shows the j-th flame. D_i is the distance between the i-th moth and j-th flame, b is a constant for defining the shape of the logarithmic spiral, t is a random number between -1 and 1. For better numerical results, this range is changed from r to 1 and during iterative operation, r is linearly reduced from -1 to -2.

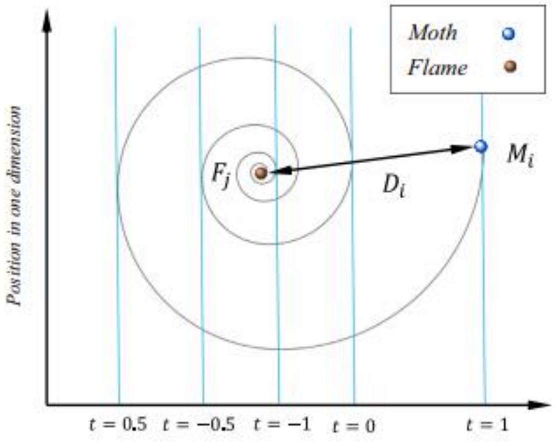


Fig. 2: Spiral Movement

3.2.1. T function

This function controls whether the stopping criterion for the algorithm is met. As it is known, there are many options for stopping criterion such as maximum iteration, tolerance value, and the minimum changes in numerical values [28].

4. ANFIS

ANFIS has a wide usage area in the literature as special network structures that integrate the learning ability of artificial neural networks with the fuzzy systems' ability to make inferences. The purpose of ANFIS is to integrate the best features of fuzzy systems and neural networks. Since "if-then" rule structure uses the input and output values, it is frequently used in estimation problems that require decision making mechanisms. ANFIS has a five-layer

Assume - two inputs X and Y and one output Z

Rule 1: If x is A1 and y is B1,
then f1 = p1x + q1y +r1

Rule 2: If x is A2 and y is B2,
then f2 = p2x + q2y +r2

structure [29-32].

The most known models like Sugeno and Mamdani are used for fuzzy systems in ANFIS. Mamdani FIS uses the fuzzification technique of a fuzzy output while Sugeno FIS uses weighted average to calculate net output. Another important difference is that Sugeno FIS has a MISO (multiple input, single output) structure while Mamdani FIS has a MIMO (multiple input, multiple output) structure. Sugeno FIS is observed to be more advantageous in terms of computational efficiency and accuracy[33].

Layer 1

Each node in this layer represents a fuzzy set like A_i and B_i. The membership degrees that depend on the input samples and the membership function used are used as the output of the nodes. In other words, it is the layer where the input values are fuzzificated. The node outputs are shown in equation (5).

$$u_i^2 = \mu_{A_i}(x) \quad \text{i=1,2}$$
$$u_{i+2}^2 = \mu_{B_i}(y) \tag{5}$$

Here, x, y and i represent the deterministic input values in the node, Ai and Bi represent the fuzzy terms, and µAi and µBi represent the membership functions. In the network which has two different inputs like x and y, there are 2 different output values and 4 nodes in total as it is shown in equation (5). Membership functions of maximum 1 and a minimum 0 are used for each node. In general, any membership function, such as triangle, Guassian, or generalized Bell, can be used for membership functions of the fuzzy set. The parameters in this layer are called Premise Parameters.

Layer 2

This layer is called a rule layer. The nodes in this layer symbolizes the rules and their number created using the fuzzy logic inference system.µi, output of the rule nodes, is the product of the membership degrees from the previous layer. Node output shows the firing strength of a rule. µi values, where (i=1,….,n), are obtained as follows.

$$\omega_i = \mu_{A_i}(x) * \mu_{B_i}(y) \tag{6}$$

Layer 3

In this layer, the firing levels from the rule layer are normalized. Therefore, this layer is called as the normalization layer. The firing level normalized for *ith* node is calculated as follows.

$$\varpi_i = \frac{\omega_i}{\omega_1 + \omega_2} \tag{7}$$

Layer 4

In this layer which is called defuzzification layer, the weighted output values of each rule are calculated. This calculation is obtained by the multiplication of firing levels that are normalized as follows with the $\{p_i, r_i, q_i\}$ values, the output parameters of the fuzzy inference system.

$$\varpi_i f_i = \varpi_i (p_i x + q_i y + r_i) \tag{8}$$

Layer 5

This layer is called as the output layer. The output of the ANFIS is obtained by the sum of outputs obtained for each rule in the previous layer. In this layer, a single number is generated by the defuzzification of fuzzy rules.

$$f(x,y) = \sum_i^4 \varpi_i f_i = \frac{\sum_i^4 \omega_i f_i}{\sum_i^4 \omega_i} \tag{9}$$

The learning algorithm of ANFIS is a hybrid learning algorithm consisting of the combined use of least squares method and backpropagation learning algorithm. There are 2 steps in the hybrid learning algorithm. These are 1. Forwardpass 2. Backwardpass

In the forwardpass process, premise parameters are constant while result parameters are updated using the least squares estimation. In the backwardpass process, result parameters are constant, and backpropagation gradient descent method is used to update the premise parameters. Forward and backward pass processes are repeated with a certain number of epochs and training is completed[34,35]. The aim of the training process in ANFIS is to adjust the parameter sets to adapt ANFIS output to training data. In this study, the MFO algorithm was used to find global optimal parameter values.

5. Training ANFIS with MFO

In ANFIS, the parameter values in the fuzzification process in layer 1 and the defuzzification process in layer 4 are provided by MFO. These parameter values are expressed by the moth-flame position values in the algorithm. These parameter values are updated as the position of the moths is updated according to the flames, and ANFIS is trained with the updated parameter values. The RMSE error function shown in (10) is used for the fitness function that determines the suitability of solution in the optimization algorithms.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \overline{y_i})}{N}} \tag{10}$$

$\overline{y_i}$ used in (10) represents the output obtained by ANFIS at time I, and y_i represents the real output of the system. N represents the number of samples used in practice.

For MFO, initial parameters are 30 moth-flame, maximum number of iterations is 1000.
The steps to be followed in training the ANFIS with MFO:
Step 1: Initial parameters for MFO
Step 2: Creating MFO initial population
Step 3: Creating initial FIS
Step 4: Calculation of initial fitness values with ANFIS initial parameters
Step 5: Updating Moth-flame values
Step 6: Trying to find the minimum error value until the end of the maximum iteration
Step 7: Showing the best parameter values with minimum error rate

6. Experiments

The application performed for the study was developed in the MATLAB environment on a computer with 8 GB, I7 processor. The algorithms used in the application were run independently for 30 times. The results show average RMSEAvg values and RMSEStd standard deviation values after 30 runs. The parameter values used for the algorithms and ANFIS are given Table 1.

Table 1: Parameters of ANFIS and Algorithm

Algorithms	Parameters of Algorithm	Parameters of ANFIS
MFO	Max Iteration:1000 Population Size:30	fcm_U=2 fcm_MaxIter=100 fcm_MinImp=1e-5 fis=genfis3(sugeno) input mf=gaussmf output mf=linear
PSO	Max Iteration:1000 Population Size:25 C_1=1,c_2=2,	
GA	Max Iteration:1000 Population Size:25 Crossover=0.4 Mutation=0.7	
WOA	Max Iteration:1000 Population Size:30	

6.1. Data Sets

HTRU2[36] and Banknote[37] which are the current data sets loaded into UCI, as well as the data sets such as
Wisconsin Breast Cancer[38], which is frequently encountered in the literature were used for classification problems. Data set properties are given in Table 2.

6.2. Classification Problem

Classification problems are among the problems that have to be solved commonly in the field of machine learning. In these problems, the aim is to group similar data according to their predetermined attributes. There is a significant need for tools to interpret and classify big data in decision making mechanisms that are used in the fields such as economy and medical. The fact that ANFIS has a structure that meets this need has made it widely used in classification problems. In order to increase the performance in the problems using ANFIS,
the most significant attributes in the data sets are trained. Here, feature extraction from data sets has an important place. The Principal Component Analysis(PCA) method will be used in the extraction of the attributes to be trained. 3 attributes were selected for the Breast Cancer dataset with PCA, while 4 attributes were selected for HTRU2 and Banknote dataset. After PCA, all attributes are normalized by z-score method. The results obtained from the algorithms for the data sets used are given in Table 3. The accuracy graphs obtained for the classification problems are shown in Figures 3,4,5 respectively.

Table 2: Properties of Data Sets

Data Set Name	Description	Type	Dimensions	Inputs	Output
Wisconsin Breast Cancer	Predict whether a cancer is malignant or benign from biopsy details.	Binary Classification	699 instances, 10 attributes	Integer (Nominal)	Integer, 2 class labels
HTRU2	Pulsar candidates collected during the HTRU survey. Pulsars is a scientifically interesting star species. Candidates should be classified into pulsar and non-pulsar classes to aid exploration.	Classification, Clustering	17898 instances, 9 attributes	Real	Integer, 2 class labels
Banknote Authentication	Data were obtained from images taken for evaluation of an authentication procedure for banknotes.	Classification	1372 instances, 5 attributes	Real	Integer, 2 class labels

Table 3: Results of Classification Problems

Wisconsin Breast Cancer									
Algorithms	**Train Data**					**Test Data**			
	Class	**Accuracy**	**Recall**	**Precision**	**Fscore**	**Accuracy**	**Recall**	**Precision**	**Fscore**
GA	2	0.97237	0.95945	0.99828	0.97847	0.975	0.9748	0.98372	0.9792
	4	0.94463	0.85755	0.97983	0.91313	0.925	0.8111	1	0.89422
PSO	2	0.97552	0.96492	0.99752	0.98094	0.9951	0.99444	0.99767	0.99601
	4	0.97204	0.93975	0.97895	0.95879	0.95931	0.89359	1	0.94314
WOA	2	0.97438	0.96253	0.99755	0.97973	0.92745	0.89722	1	0.94582
	4	0.91737	0.78295	0.98171	0.87097	0.94902	0.87667	0.94582	0.90975
MFO	2	0.97834	0.96949	0.99736	0.98322	0.97206	0.97561	0.97855	0.97694
	4	0.97568	0.94623	0.98279	0.96398	0.94118	0.88765	0.9614	0.92259
Bank Data Set									
Algorithms	**Train Data**					**Test Data**			
	Class	Accuracy	Recall	Precision	**Fscore**	**Accuracy**	**Recall**	**Precision**	**Fscore**
GA	0	0.98397	0.97109	1	0.98531	0.99367	0.98874	1	0.9943
	1	0.9897	1	0.9654	0.98236	0.99392	0.99944	0.9871	0.99316
PSO	0	0.99968	0.99942	1	0.99971	0.9983	0.99701	0.99951	0.99825
	1	0.99968	1	0.99927	0.99963	0.9983	0.99952	0.99718	0.99834
WOA	0	0.99026	0.98374	0.99884	0.9912	0.98808	0.977	1	0.9883
	1	0.99026	0.99853	0.97989	0.98908	0.98808	1	0.97613	0.98785
MFO	0	0.99914	0.99845	1	0.99922	1	1	1	1
	1	0.99914	1	0.99807	0.99903	1	1	1	1
HTRU2 Data Set									
Algorithms	**Train Data**					**Test Data**			
	Class	Accuracy	Recall	Precision	**Fscore**	**Accuracy**	**Recall**	**Precision**	**Fscore**
GA	0	0.97308	0.99397	0.9768	0.98531	0.97145	0.98903	0.97985	0.98442
	1	0.9732	0.76678	0.92963	0.84022	0.97158	0.78966	0.87656	0.8306
PSO	0	0.97267	0.99288	0.9774	0.98507	0.97955	0.99216	0.98547	0.9888
	1	0.97272	0.7726	0.91745	0.83861	0.97957	0.85197	0.91541	0.88237
WOA	0	0.97151	0.99326	0.97575	0.98443	0.98248	0.99758	0.98369	0.99059
	1	0.97199	0.76028	0.92662	0.83518	0.98291	0.79728	0.9795	0.87554
MFO	0	0.97361	0.99306	0.97824	0.9856	0.97201	0.99034	0.97883	0.98455
	1	0.97361	0.77855	0.91828	0.84254	0.97201	0.80599	0.90236	0.85133

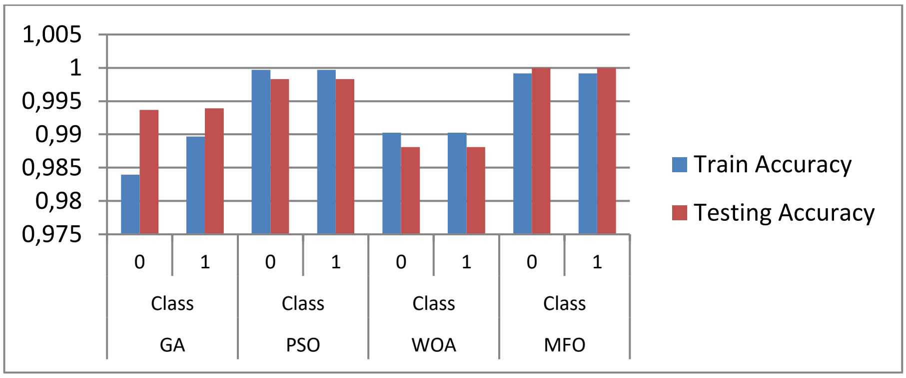


Fig. 4: Accuracy graph for Banknote Dataset

Fig. 5: Accuracy graph for HTRU2 Dataset

For the Breast Cancer data set, the highest accuracy rate was obtained by MFO algorithm in the train data and PSO algorithm in the test data.

When looking at the results for Banknote dataset, PSO has the highest accuracy rate in train data. In the test data, the MFO algorithm achieved a better result than the other algorithms with a success rate of 100%.

Finally, in HTRU2 dataset, MFO algorithm has reached the highest accuracy rate in train data, while the test data shows the superiority of WOA algorithm this time. The Accuracy, Recall, Precision and Fscore values obtained from the algorithms are given in Table 3.

6.3. Nonlinear system identification

A dynamic system consisting of an input and an output (SISO) was used in the problem given for the identification of non-liner systems. Furthermore, for the ANFIS structure, 2 input values and 1 output value are used in problem 1 and 2. 3 input values and 1 output value are used in problem 3. Gaussian functions are used as the membership function for inputs in the ANFIS structure. Two inputs for problem 1 and 2 of ANFIS are defined as the previous input u(k) and the previous output y(k) of the dynamic systems. The previous input u(k) and the previous output y(k) and y(k-1) of the dynamic systems are used as the input values for problem 3. y(k+1) is accepted as the output value for all problems. The input value u(k) for the sample 1 is obtained using equation 11. Equation 12 is used for the u(k) value of the samples 2 and 3. The use of the ANFIS and MFO pair in dynamic system identification is represented as in Figure 6.

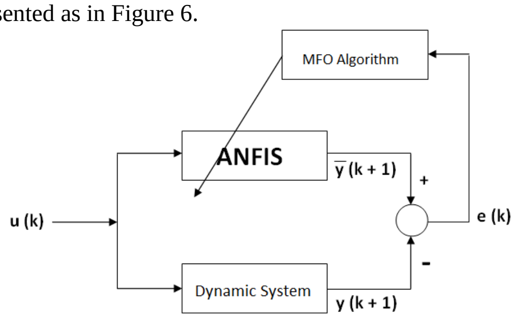


Fig. 6: MFO-ANFIS dynamic system identification

$$u(k)=\begin{cases}\sin(\pi k/25) & k<250\\ 1 & 250\le k\le 500\\ -1 & 500<k\le 750\\ 0.3\sin(\pi k/25)+ & 750<k\le 1000\\ 0.1\sin(\pi k/32)+0.6\sin(\pi k/10) & \end{cases} \quad (11)$$

$$u(k)=\sin(\pi k/25) \quad (12)$$

For the example 1 and 2, 1000 input and output values are generated using the nonlinear system shown in Table 4. These are divided into 90% training and 10% test data and used in training the ANFIS. For the example 3, 300 data were generated using the nonlinear system shown in table 1. These were divided into 80% training and 20% test data. The results of these problems obtained from PSO, GA, WOA, and the results of the proposed method are given in Table 5.

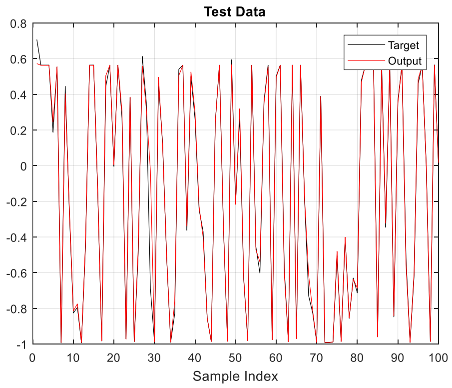
Table 4: Nonlinear System

Example	Inputs of ANFIS	Number of training/testing data	Nonlinear System
1	u(k), yp(k)	900/100	$y_p(k+1)=\dfrac{y_p(k)y_p(k-1)y_p(k-2)u(k-1)(y_p(k-2)-1)+u(k)}{1+y_p(k-1)^2+y_p(k-2)^2}$
2	u(k), yp(k)	900/100	$y_p(k+1)=0.72y_p(k)+0.025y_p(k-1)u(k-1)$ $+0.01u^2(k-2)+0.2u(k-3)$
3	u(k), yp(k), yp(k-1)	200/50	$y_p(k+1)=\dfrac{y_p(k)y_p(k-1)(y_p(k)+2.5)}{1+y_p(k)^2+y_p(k-1)^2}+u(k)$

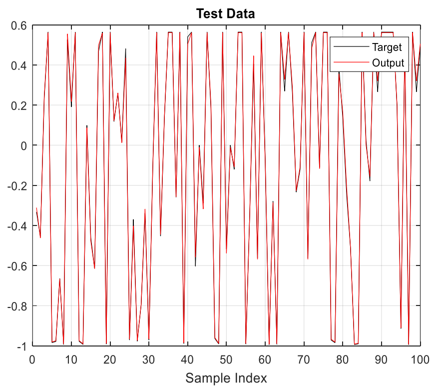
Table 5: Results of Nonlinear System Identification Problems

Example 1						
Algorithms	**Train Data**			**Test Data**		
	Min RMSE	**RMSEAvg**	**RMSEStd**	**Min RMSE**	**RMSEAvg**	**RMSEStd**
GA	0.028317	0.034497	0.0049359	0.017587	0.025575	0.0058389
PSO	0.023806	0.025845	0.0010108	0.024223	0.061103	0.022772
WOA	0.04433	0.053142	0.0037386	0.042928	0.051326	0.0032938
MFO	0.023945	0.025226	0.00067538	0.025832	0.029011	0.0012295
Example 2						
Algorithms	**Train Data**			**Test Data**		
	Min RMSE	**RMSEAvg**	**RMSEStd**	**Min RMSE**	**RMSEAvg**	**RMSEStd**
GA	0.0022614	0.0030242	0.00024334	0.00050072	0.00094621	0.00035333
PSO	0.00077615	0.001624	0.00047679	0.0045633	0.0084168	0.012737
WOA	0.0034549	0.0037649	0.00021664	0.0014751	0.002287	0.00042669
MFO	0.0010912	0.0020344	0.00063359	0.00026673	0.00062601	0.00023767
Example 3						
Algorithms	**Train Data**			**Test Data**		
	Min RMSE	**RMSEAvg**	**RMSEStd**	**Min RMSE**	**RMSEAvg**	**RMSEStd**
GA	0.023749	0.060524	0.025537	0.017643	0.045377	0.023948
PSO	0.0026569	0.013998	0.010851	0.047931	0.087383	0.01809
WOA	0.081743	0.10718	0.013713	0.065314	0.089369	0.013235
MFO	0.0096924	0.025316	0.0099627	0.0061654	0.016596	0.0049912

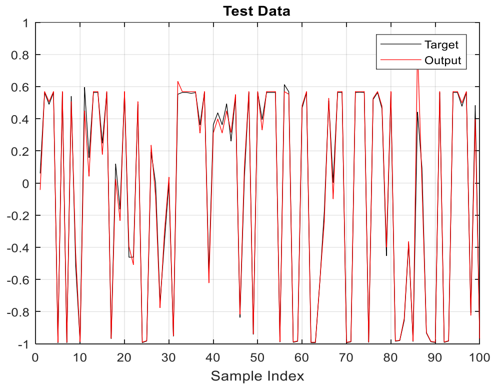
As shown in the above table, MFO has reached low error rates in RMSEtrain(Average) value of example 1 and RMSEtest(Average) value of example 2 and 3. The small standard deviations(RMSEStd) indicate that the results are consistent. For examples the target and output values obtained from all the algorithms are shown in Figures 7, 8, 9 respectively.



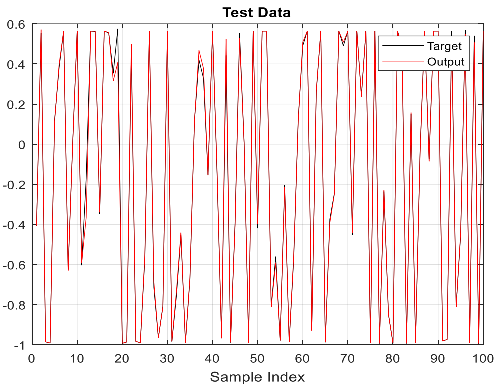
(a)GA



(b)PSO

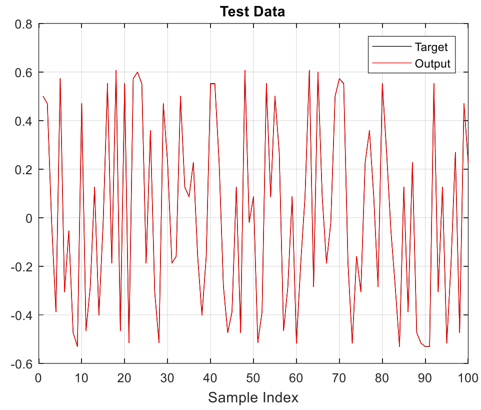


(c) WOA

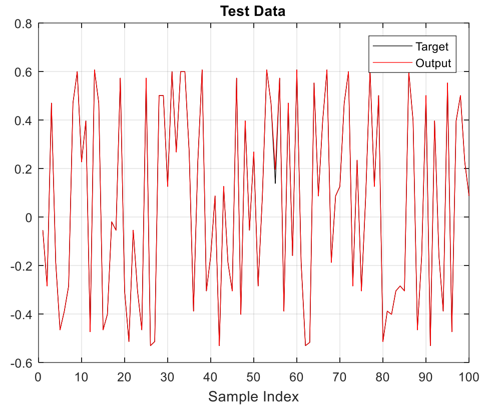


(d)MFO

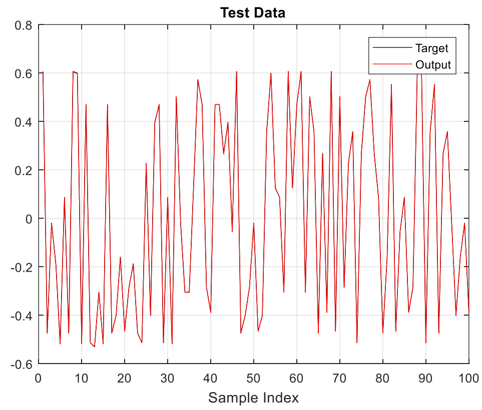
Fig. 7: Targets and outputs values for example 1



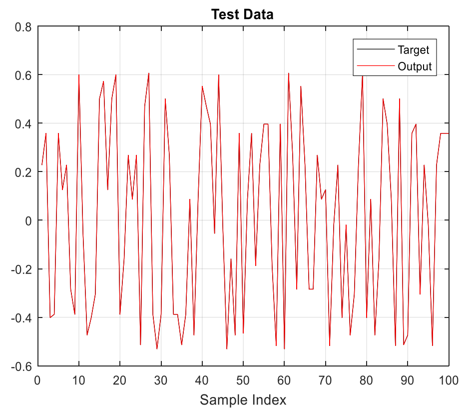
(a)GA



(b)PSO

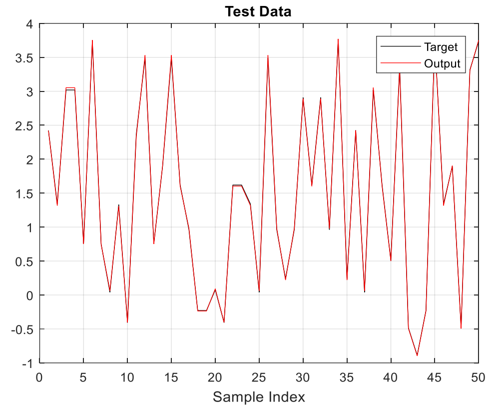


(c) WOA

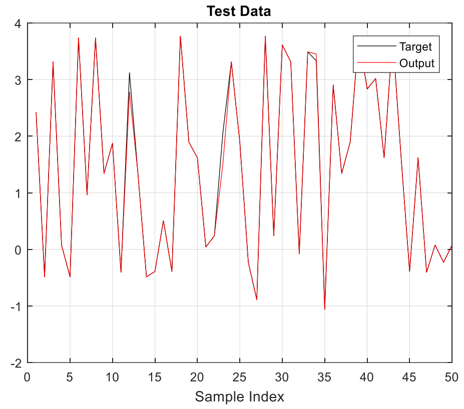


(d)MFO

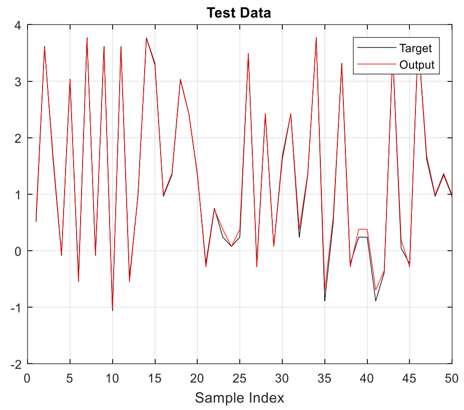
Fig. 8. Targets and outputs values for example 2



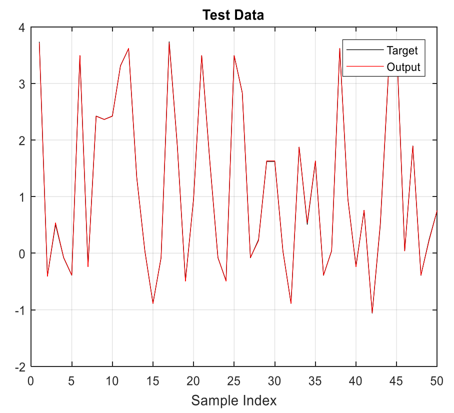
(a)GA



(b)PSO



(c) WOA



(d)MFO

Fig. 9. Targets and outputs values for example 3

6.4. Time Series Prediction

6.4.1. Mackey-Glass

Mackey-Glass[39] is the Chaotic time series estimation problem. The aim of the problem is to estimate the x(t+6) value using the x(t-18), x(t-12), x(t-6), and x(t) values. The data set was divided into 500 training and 500 test data.The time series graph used for Mackey-Glass is given in Figure 10. The data used were obtained from the time-delay equation shown by equation 13 by Mackey-Glass.

$$\frac{dx(t)}{dt}=\frac{0.2x(t-\tau)}{1+(x(t-\tau))^{10}}-0.1x(t) \qquad (13)$$

When x(0) = 1.2 and τ = 17, a non-periodic and non-convergent time series is obtained which is very sensitive to the initial conditions (assuming that x (t) = 0 when T <0).

The results of Mackey-Glass time series prediction are given Table 6. Errors values which are obtained from algorithms are demonstrated Figure 11.

As shown in the below table, PSO algorithm gives the best result for this problem. The results of the MFO are close to the PSO.

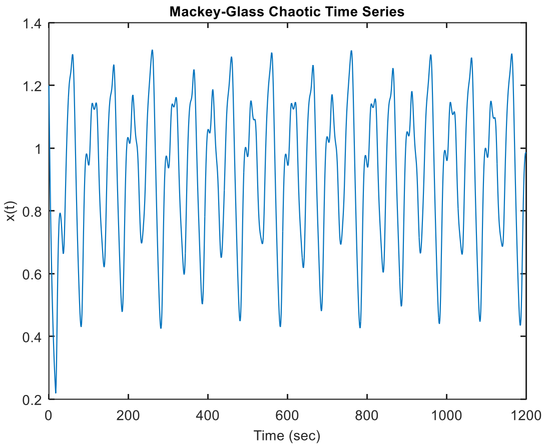
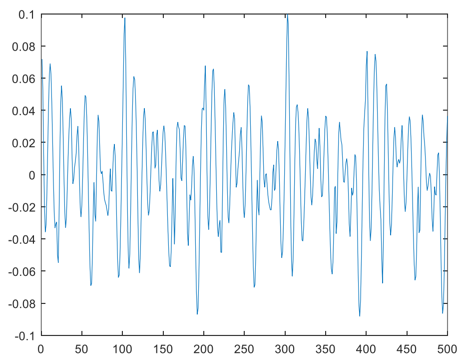


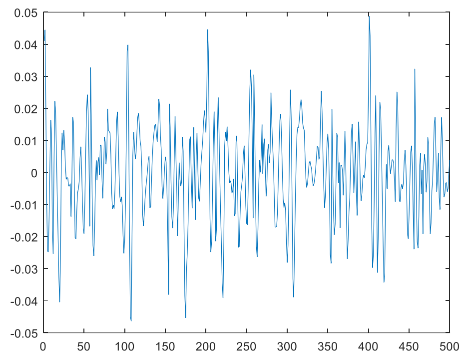
Figure 10: Mackey-Glass time series

Table 6: Mackey-Glass Results

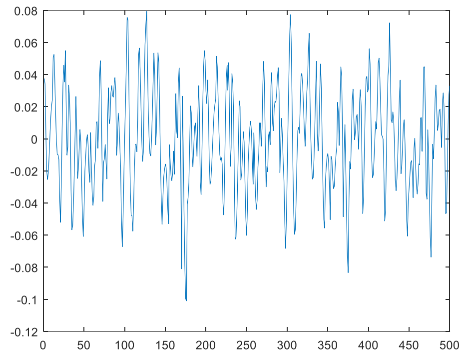
Mackey-Glass Time Series						
Algorithms	**Train Data**			**Test Data**		
	Min RMSE	**RMSE Avg**	**RMSE Std**	**Min RMSE**	**RMSE Avg**	**RMSE Std**
GA	0.0346	0.0597	0.0167	0.0343	0.0591	0.0167
PSO	0.0150	0.0214	0.0035	0.0150	0.0212	0.0037
WOA	0.0316	0.0417	0.0039	0.0312	0.0411	0.0038
MFO	0.0187	0.0227	0.0026	0.0187	0.0225	0.0029



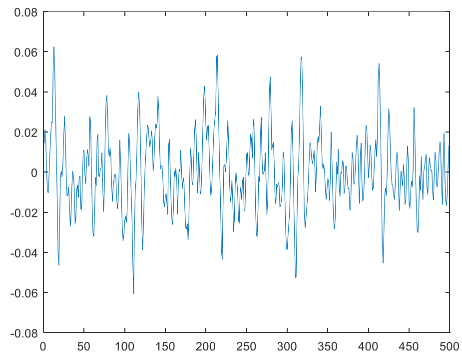
(a)GA



(b)PSO



(c) WOA



(d)MFO

Figure 11: Errors of Mackey-Glass

6.4.2. Lorenz Attractor

Lorenz attractor, one of the chaotic continuous-time dynamic systems, is the time series resulting from the mathematical model of three-dimensional fluid convection proposed by Lorenz for the estimation of meteorological events. It is mathematically expressed as follows [40].

$$\frac{dx}{dt}=\sigma(y-x)$$

$$\frac{dy}{dt}=rx-y-xz \qquad (14)$$

$$\frac{dz}{dt}=xy-bz$$

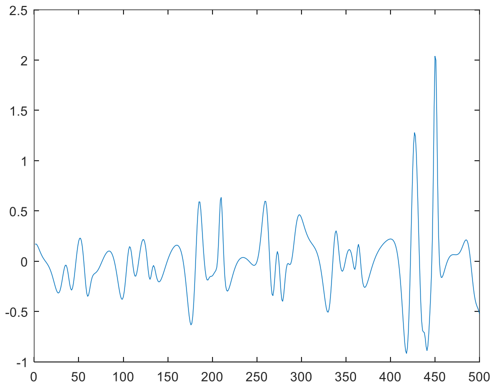
Here, σ, b, r are state variables.

1200 sample data set was used by the use of σ =10.0, r = 28.0, b = 8/3. values under initial conditions [41]. This data set was divided

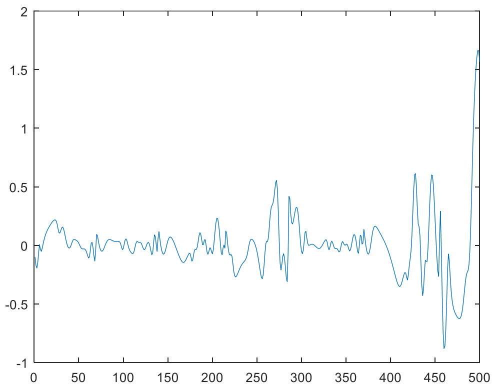
into 500 training and 500 test data sets.
The results of Lorenz attractor are given Table 7. Errors values which are obtained from algorithms are demonstrated Figure 12.
As shown in the below table, MFO algorithm gives the best result for RMSEtrain, while PSO algorithm gives the best result for RMSEtest.

Table 7: Lorentz Attractor Results

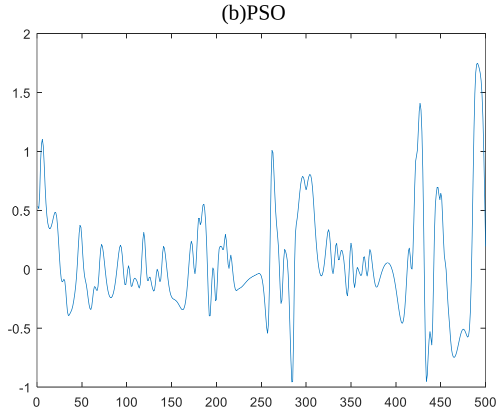
Lorentz Time Series						
Algorithms	Train Data			Test Data		
	Min RMSE	RMSE Avg	RMSE Std	Min RMSE	RMSE Avg	RMSE Std
GA	0.2218	0.4363	0.1229	0.3288	1.3304	3.8890
PSO	0.0887	0.1150	0.0188	0.2320	0.3011	0.0532
WOA	0.2878	0.3540	0.0268	0.3743	0.4159	0.0191
MFO	0.0796	0.1045	0.0153	0.2462	0.3421	0.2232



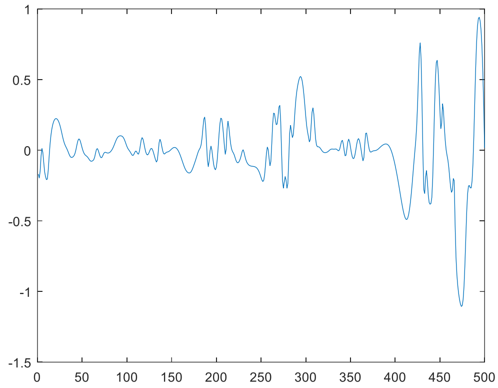
(a)GA



(b)PSO



(c) WOA



(d)MFO

Figure 12: Errors of Lorentz Attractor

6.4.3. Rossler Attractor

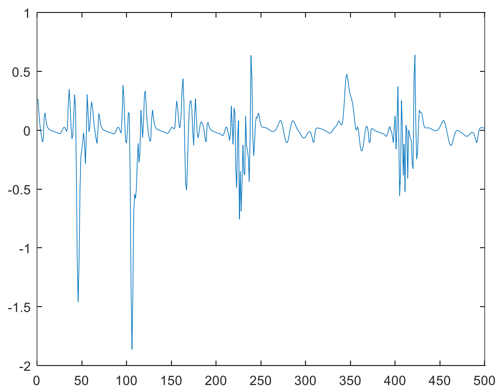
This attractor which was created by Otto Rössler was revealed by modeling the stability in chemical reactions [42]. The mathematical expressions of the attractor are given in (15).

$$\frac{dx}{dt} = -z - y$$
$$\frac{dy}{dt} = x + ay \tag{15}$$
$$\frac{dz}{dt} = b + z(x - c)$$

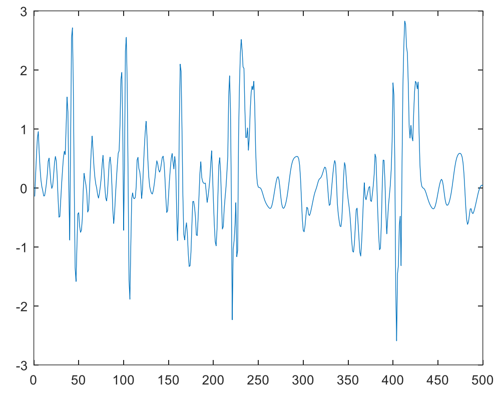
1000 sample data set was used by the use of a=0.15, b=0.20, c=10 values under initial conditions. This data set was divided into 500 training and 500 test data sets. The results of Rossler attractor are given Table 8. Errors values which are obtained from algorithms are demonstrated Figure 13.
As shown in the below table, MFO algorithm gives the best results than the others.

Table 8: Results of Rossler Attractor

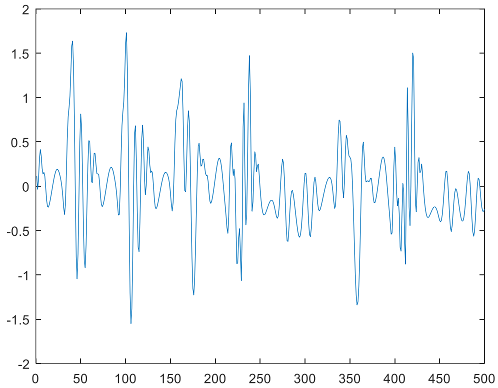
Rossler Time Series						
Algorithms	Train Data			Test Data		
	Min RMSE	RMSE Avg	RMSE Std	Min RMSE	RMSE Avg	RMSE Std
GA	0.4572	0.9028	0.2517	0.4687	1.2103	1.5822
PSO	0.1815	0.3521	0.1382	0.2214	0.4942	0.1654
WOA	0.8165	1.006	0.0940	0.7674	0.9724	0.0925
MFO	0.1422	0.2659	0.0635	0.2113	0.4593	0.4851



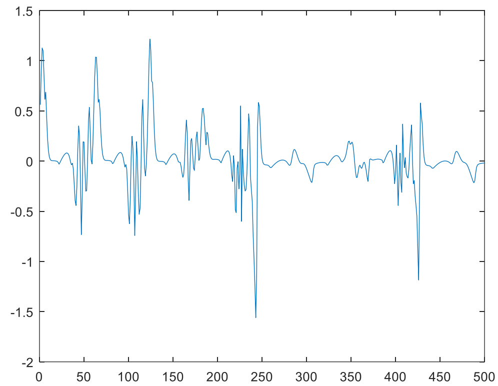
(a) GA



(b)PSO



(c) WOA



(d)MFO

Figure 13: Errors of Rossler Attractor

7. Conclusions

In this study, the ANFIS model, which is a special network type in which fuzzy and artificial neural network are used together, was trained with MFO, which is one of the current heuristic optimization methods. The parameter values providing the minimum error rate in the fitness function were determined to find the parameter values required for ANFIS during the training process. It was tried to solve many problems, such as classification, nonlinear system identification, and time series estimation, to show the performance of MFO on ANFIS. In addition, the results obtained from the known, current heuristic methods such as PSO, GA, and WOA, and the results obtained from MFO are presented in graphics.

When the results obtained at the end of the study were examined, it was observed that MFO achieved successful results for training the ANFIS. Since this study is the first study on training the ANFIS with MFO, it is predicted that researchers will be able to use the MFO-ANFIS pair in their future studies. The performance of this pair will be tested in the fields such as computer vision and medical in future studies.

Acknowledgements

This study is supported Scientific Research Projects Department Project No. FBA-2018-6915 by Van Yuzuncu Yıl University.

References

[1] Zadeh L.A., “Fuzzy Sets”. Elsevier Information and Control, 8, 338-35,1965.

[2] Şen Z., Bulanık Mantık İlkeleri ve Modelleme, Genişletilmiş 3. Baskı, Su Vakfı Yayınları, 2009.

[3] Mirjalili, S., Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowledge-Based Systems,Volume 89, 2015,Pages 228-249,

[4] Kennedy, J., Eberhart, R. "Particle Swarm Optimization". Proceedings of IEEE International Conference on Neural Networks. IV. pp. 1942–1948, 1995.

[5] Winter, G., Periaux J., Galan, M., Genetic Algorithms in Engineering and Computer Science , John Wiley & Son Ltd. 1995.

[6] Mirjalili, S., Lewis, A., The Whale Optimization Algorithm,Advances in Engineering Software,Volume 95,2016,Pages 51-67,

[7] Yamany, W., et al. "Moth-flame optimization for training multi-layer perceptrons." Computer Engineering Conference (ICENCO), 2015 11th International. IEEE, 2015.

[8] Aboul Ella, H., et al. "An improved moth flame optimization algorithm based on rough sets for tomato diseases detection." Computers and Electronics in Agriculture 136 (2017): 86-96

[9] Sayed, G. I., Mona S., Aboul Ella H., "Bio-inspired Swarm Techniques for Thermogram Breast Cancer Detection." Medical Imaging in Clinical Applications. Springer International Publishing, 2016. 487-506.

[10] El Aziz, M.A., Ewees, A. A., and Aboul Ella H., "Whale Optimization Algorithm and Moth-Flame Optimization for Multilevel Thresholding Image Segmentation." Expert Systems with Applications 2017.

[11] Zawbaa, H. M., et al. "Feature selection approach based on moth-flame optimization algorithm." Evolutionary Computation (CEC), 2016 IEEE Congress on. IEEE, 2016.

[12] Kumar, L.D., Bhoi, K.K., Barisal A.K., "Performance evaluation of MFO algorithm for AGC of a multi area power system." transfer 1: 4. 2016

[13] Narottam, J., et al. "Price Penalty factors Based Approach for Combined Economic Emission Dispatch Problem Solution using Moth-flame Optimizer Algorithm",2016

[14] Sayed, G.I., Hassanien, A.E. Complex Intell. Syst. A hybrid SA-MFO algorithm for function optimization and engineering design problems, 2018. https://doi.org/10.1007/s40747-018-0066-z,

[15] Catalao, J. P. S., Pousinho H. M. I.,. Mendes, V. M. F., "Hybrid wavelet-PSO-ANFIS approach for short-term electricity prices forecasting", IEEE Transactions on Power Systems, vol. 26, no. 1, pp. 137-144, 2011.

[16] Jiang, H. , Kwong, C. K., Ip W. H., and Wong, T. C.. "Modeling customer satisfaction for new product development using a PSO-based ANFIS approach", Applied Soft Computing, vol. 12, no. 2, pp. 726-734, 2012

[17] Pousinho, H. M. I., Mendes V. M. F., Catalão, J. P. S., "A hybrid PSO–ANFIS approach for short-term wind power prediction in Portugal", Energy Conversion and Management, vol. 52, no. 1, pp. 397-402,2011.

[18] Pousinho, H. M. I., Mendes V. M. F., Catalão, J. P. S., "Short-term electricity prices forecasting in a competitive market by a hybrid

PSO–ANFIS approach", International Journal of Electrical Power & Energy Systems, vol. 39, no. 1, pp. 29-35, 2012.

[19] Salleh, M.N.M., Hussain, K., A Review of Training Methods of ANFIS for Applications in Business and Economics, International Journal of u- and e- Service, Science and Technology,Vol.9, No. 7 ,pp.165-172, 2016.

[20] Turki, M., Bouzaida, S., Sakly A., M'Sahli, F., "Adaptive control of nonlinear system using neuro-fuzzy learning by PSO algorithm. in Electrotechnical Conference (MELECON)", 2012 16th IEEE Mediterranean, 2012.

[21] Haznedar, B., Arslan, M.T., Kalınlı, A., Karaciğer mikrodizi kanser verisinin sınıflandırılması için genetik algoritma kullanarak ANFIS'in eğitilmesi, Sakarya Üniversitesi Fen bilimleri Enstitüsü Dergisi, doi: 10.16984/saufenbilder.41925

[22] Haznedar, B., Kalınlı, A., "Training ANFIS Using Genetic Algorithm for Dynamic Systems Identification," Int. J. of Intell. Sys. and Appl. in Eng., vol. 4, no. 1, pp. 44-47, 2016.

[23] Karaboga, D., Kaya, E., "Training ANFIS using artificial bee colony algorithm for nonlinear dynamic systems identification," in 2014 22nd Signal Processing and Communications Applications Conference (SIU), pp. 493–496.

[24] Karaboga, D., Kaya, E., Training ANFIS by using the artificial bee colony algorithm, Turk J Elec Eng & Comp Sci (2017) 25: 1669 – 1679

[25] Thangavel K., Kaja Mohideen A, Mammogram Classification Using ANFIS with Ant Colony Optimization Based Learning. Communications in Computer and Information Science, vol 679. Springer, Singapore, 2016.

[26] Canayaz M., Özdağ R., "Training ANFIS using The Whale Optimization Algorithm ", International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES 2018), KARABÜK, TÜRKIYE, 11-13 Mayıs 2018, pp.409-414

[27] Sayed, G. I., et al. "Alzheimer's Disease Diagnosis Based on Moth Flame Optimization." International Conference on Genetic and Evolutionary Computing. Springer International Publishing, 2016.

[28] Ceylan, O., "Harmonic elimination of multilevel inverters by moth-flame optimization algorithm," 2016 International Symposium on Industrial Electronics (INDEL), Banja Luka, 2016, pp. 1-5.

[29] Jang, J. S. R., "ANFIS: Adaptive-Network-Based Fuzzy Inference System," IEEE Trans Syst Man Cybern, vol. 23, no. 3, pp. 665–685, 1993.

[30] Jang, J. S. R., Sun, C.T., Mizutani, E., Neurofuzzy and soft computing, Prentice Hall, Upper Saddle River, 1997

[31] Elmas, Ç., Yapay Zeka Uygulamaları, Seçkin Yayıncılık, 2016.

[32] K S, Elhoseny M, S K L, et al. Optimal feature level fusion based ANFIS classifier for brain MRI image classification.Concurrency Computat Pract Exper. 2018;e4887. https://doi.org/10.1002/cpe.4887

[33] Hamam, A., Georganas, N. D., A Comparison of Mamdani and Sugeno Fuzzy Inference Systems for Evaluating the Quality of Experience of Hapto-Audio-Visual Applications, HAVE 2008 – IEEE International Workshop on Haptic Audio Visual Environments and their Applications, Ottawa – Canada, 18-19 October 2008

[34] Vaidhehi, V., The role of Dataset in training ANFIS System for Course Advisor, International Journal of Innovative Research in Advanced Engineering (IJIRAE) ,Volume 1 Issue 6 ,July 2014,pp 249-253

[35] Lin X., Sun J., Palade V., Fang W., Wu X., Xu W. (2012) Training ANFIS Parameters with a Quantum-behaved Particle Swarm Optimization Algorithm. In: Tan Y., Shi Y., Ji Z. (eds) Advances in Swarm Intelligence. ICSI 2012. Lecture Notes in Computer Science, vol 7331. Springer, Berlin, Heidelberg

[36] Lyon, R. J., Stappers, B. W., Cooper, S., Brooke, J. M. ,. Knowles, J. D., Fifty Years of Pulsar Candidate Selection: From simple filters to a new principled real-time classification approach, Monthly Notices of the Royal Astronomical Society 459 (1), 1104-1123, DOI: 10.1093/mnras/stw656

[37] Volker L., Helene D., UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. 2013.

[38] Mangasarian O. L., Wolberg, W. H.. "Cancer diagnosis via linear programming", SIAM News, Volume 23, Number 5, September 1990, pp 1 & 18

[39] Mackey, M.C., Glass, L., Oscillation and chaos in physiological control systems. Science, 197 (1977), pp. 287-289

[40] PAMUK, N., Dinamik Sistemlerde Kaotik Zaman Dizilerinin Tespiti, BAÜ Fen Bil. Enst. Dergisi Cilt 15(1) 77-91 (2013)

[41] Weeks, E.R., My Adventures in Chaotic Time Series Analysis,2018 http://www.physics.emory.edu/faculty/weeks//research/tseries1.html

[42] Rössler, O. E., Chaotic behavior in simple reaction system, Zeitschrift für Naturforsch A, 31, 259-264, 1976.