Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı 28, S. 1083-1092, Kasım 2021
© Telif hakkı EJOSAT'a aittir
Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue 28, pp. 1083-1092, November 2021
Copyright © 2021 EJOSAT
Research Article

Dynamic Optimal ANFIS Parameters Tuning with Particle Swarm Optimization

Mahmut Dirik[1]*, Mehmet Gül[2]

[1*] Şırnak University, Faculty of Engineering, Departmant of Computer Engineering, Şırnak, Turkey, (ORCID: 0000-0003-1718-5075), mhmd.dirik@gmail.com
[2] Şırnak University, Faculty of Engineering, Departmant of Computer Engineering, Şırnak, Turkey, (ORCID: 0000-0002-4819-4743), gul.mehmet.21@gmail.com

(1st International Conference on Applied Engineering and Natural Sciences ICAENS 2021, November 1-3, 2021)

(**DOI:** 10.31590/ejosat.1012888)

ATIF/REFERENCE: Dirik, M. & Gül, M. (2021). Dynamic Optimal ANFIS Parameters Tuning with Particle Swarm Optimization. *European Journal of Science and Technology*, (28), 1083-1092.

Abstract

This paper presents dynamic modification parameters of the Adaptive Neuro-Fuzzy Inference System (ANFIS) using the Particle Swarm Optimization (PSO) algorithm. In the proposed ANFIS_PSO, each particle dynamically adjusts its weight to the optimal states of the particles using a nonlinear fuzzy model. Tests of the model were performed using the "Signal-Time Series". The methods are tested simultaneously until the best method to solve the problem is found. The proposed model takes advantage of PSO to tune ANFIS parameters by minimizing mean square error (MSE), root mean square error (RMSE), R-Squared (R2) and Mean Absolute Error (MEA) metrics. The main contribution is a strategy for dynamically finding the best result, which identifies methods for solving a given problem using different performance metrics depending on the problem. The proposed structure's results were compared with several machine learning algorithms. Simulation results show the effectiveness of the proposed algorithm.

Keywords: Adaptive Neuro-Fuzzy Inference System; Parameter estimation; Particle Swarm Optimization; Intelligent control.

Parçacık Sürüsü Optimizasyonu ile Dinamik Optimal ANFIS Parametrelerinin Uyarlaması

Öz

Bu çalışmada, Parçacık Sürü Optimizasyonu (PSO) algoritması kullanarak Uyarlamalı Nöro-Bulanık Çıkarım Sisteminin (ANFIS) dinamik modifikasyon parametreleri işlenmiştir. Önerilen ANFIS_PSO'da, her parçacık, doğrusal olmayan bir bulanık model kullanarak ağırlığını parçacıkların optimal durumlarına dinamik olarak ayarlanır. Modelin testleri "Signal-Time Series" kullanılarak gerçekleştirilmiştir. Yöntemler, sorunu çözmek için en iyi yöntem bulununa kadar aynı anda test edilir. Önerilen model, ortalama kare hata (MSE), R^2, Ortalama Mutlak Hata (MEA) ve kök ortalama kare hata (RMSE) ölçümlerini en aza indirerek ANFIS parametrelerini ayarlamak için PSO'dan yararlanır. Çalışmanın esas katkısı, soruna bağlı olarak farklı performans ölçütlerini kullanarak belirli bir sorunu çözme yöntemlerini tanımlayan en iyi sonucu dinamik olarak bulmaya yönelik bir stratejinin geliştirilmesidir. Önerilen yapının sonuçları, çeşitli makine öğrenme algoritmaları ile karşılaştırılmıştır. Simülasyon sonuçları, önerilen algoritmanın etkinliğini göstermektedir.

Anahtar Kelimeler: Uyarlanabilir Nöro-Bulanık Çıkarım Sistemi; Parametre tahmini; Parçacık Sürü Optimizasyonu; Akıllı kontrol.

* Corresponding Author: mhmd.dirik@gmail.com

1. Introduction

In recent years, successful results have been obtained in solving various problems using intelligent systems developed as alternatives to traditional methods. Application methods such as fuzzy logic and neural networks have been used as complementary tools and have contributed to the development of intelligent systems (Basser et al. 2015; Hodzic 2016; Rini, Shamsuddin, and Yuhaniz 2016; Shamshirband et al. 2019). Fuzzy logic is a powerful and easy to design method that is created by applying expert knowledge about inference rules without requiring model knowledge (Guillaume 2001). The fuzzy inference system mainly consists of three conceptual components: the rule base, which consists of the sum of fuzzy rules, the database, which is used to define the degrees of membership, and the inference mechanism, which is used to collect rules from the inputs and outputs of the system and produce corresponding results (Hodzic 2016)(Sada and Ikpeseni 2021). The superiority of artificial neural networks over demand forecasting methods was first demonstrated in (Zhang, Eddy Patuwo, and Y. Hu 1998). An evaluation study conducted in early 2000 by Zhang et al (Zhang, Eddy Patuwo, and Y. Hu 1998) addressed 23 problems. The study found that ANN is quite successful compared to traditional demand forecasting methods. The reason for this is that, compared to statistical models, the ANN method does not require a functional structure that defines the relationships of decision makers between variables. Thus, although ANN does not require assumptions, it has the ability to learn the relationship between variables. While performing linear regression analysis, assumptions such as the absence of correlation between independent variables, the occurrence of estimation errors independently of each trial, and the distribution of errors with constant variance and mean should be made. On the other hand, in ANN the superiority of eliminating the ambiguity between variables, not requiring assumptions and learning the relationships of the model is recognized.

The Artificial Intelligence (AI) method is known for its flexible structure and its ability to produce extremely success-oriented results in nonlinear systems. When properly trained, it achieves results with high accuracy using the methods of interpolation and extrapolation of random data. With this approach, AI functions like a black box. Adaptive Network Based Fuzzy Inference System method is a hybrid method that consists of the combination of the above two powerful methods. While ANFIS is mainly used for task estimation or estimation of nonlinear functions with an output, the main purpose of this method is to optimize the parameters used. Objects may not belong to any particular class, as it can observe in many real-life events. The approach of classification by assigning degrees of membership to objects that do not belong was first proposed by Zadeh (Zadeh 1994). Adaptive Network-based Fuzzy Inference System (ANFIS) classification is an artificial neural network based method, which is particularly used to solve pattern recognition problems that require rule-based process control (Sada and Ikpeseni 2021). These approaches are obtained after training fuzzy inference rules with artificial neural networks, which even experts in the field have difficulty in drawing conclusions (Kundapura and Hegde 2021; Yaseen et al. 2019). The proposed method presents dynamic modification parameters of the ANFIS using the PSO algorithm. the ANFIS_PSO dynamically adjusts its weight to the optimal states of the particles using a nonlinear fuzzy model. The proposed model takes advantage of PSO to tune ANFIS parameters by minimizing mean square error (MSE) and root means square error (RMSE) metrics. The general architectural structure of the proposed system is shown in Fig. 1.

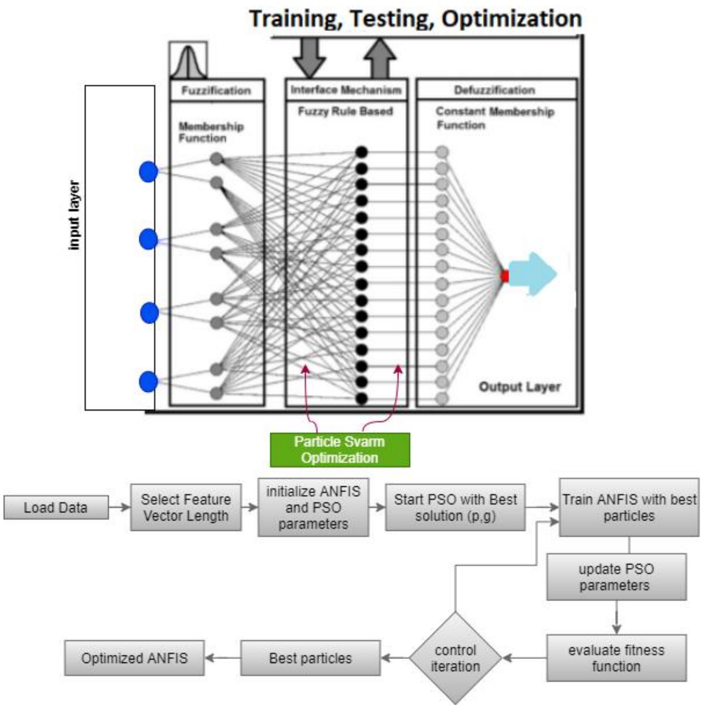

Training, Testing, Optimization
input layer
Fuzzification
Membership Function
Interface Mechanism
Fuzzy Rule Based
Defuzzification
Constant Membership Function
Output Layer
Particle Svarm Optimization
Load Data
Select Feature Vector Length
initialize ANFIS and PSO parameters
Start PSO with Best solution (p,g)
Train ANFIS with best particles
update PSO parameters
evaluate fitness function
control iteration
Best particles
Optimized ANFIS

Fig. 1. Flow chart of the proposed ANFIS+PSO

The remainder of this paper proceeds to organize as follows: Section 2 summarizes the materials and methods. This section covers the basic concepts of a standard PSO and ANFIS structure. Section 3 illustrates in detail the methodology of the proposed algorithm integration and structure. The analyses and the corresponding results presented in Section 4. Finally, in Section 5, general considerations and other proposals for the future were proposed.

2. Material and Method

2.1. Adaptive Neuro-Fuzzy Inference System (Anfis)

The ANFIS architecture consists of 6 layers; the first layer is used for data inputs. While the incoming data is fuzzy in the second layer, the fuzzy data is processed according to a specific rule defined in the third layer. The fourth layer is defined as normalization layer while the fifth layer is defined as defuzzification layer. In the sixth and final layer, the values from the previous layer are collected and given as the actual output value to the ANFIS system (El-Hasnony, Barakat, and Mostafa 2020). It includes a pilot zone and four separate zone to support a typical workflow in the ANFIS design window. In the design window, tasks such as loading and cleaning data, creating or loading the initial FIS structure, FIS training, and validating the trained FIS are performed. Figure 2 shows the ANFIS architecture.

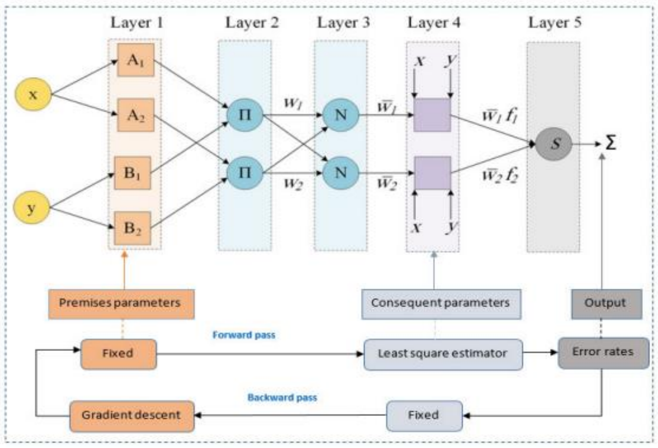


Fig. 2 ANFIS architecture [11]

Layer 1: The first layer of the architecture is defined as the input layer and the input signals from each node in this layer are forwarded to the other layers. Layer 2: It is also known as fuzziness layer because the data from the first layer is fuzzy. The output of each node in the fuzzification layer consists of membership degrees depending on the input values and the membership functions used. The membership degrees obtained from the 2nd layer are defined as $\mu_{A_j}(x)$ and $\mu_{B_j}(y)$. Layer 3: defined as the rule layer. Each node is re-evaluated according to the rules created using the Sugeno fuzzy logic inference system. The output of each rule node μ_i is obtained by multiplying the membership degrees from the 2. The μ_i values are (j=1, 2) and (i = 1, ..., n).

$$y_i^3 = \Pi i = \mu_{A_j}(x)\ x\ \mu_{B_i}(y) = \mu_i \qquad (1)$$

The value y_i^3 represents the output values of the 3rd layer and n represents the number of nodes in this layer. Layer 4: In this layer, defined as the normalization layer, the firing level is calculated by accepting the input value of all nodes sent by the rule layer and normalizing each rule. Calculation of the normalized firing using Eq. 2.

$$y_i^4 = Ni = \frac{\mu_i}{\sum_{i=1}^{n} \mu_i} = \overline{\mu_i}, \qquad (i = 1, \dots, n) \qquad (2)$$

Layer 5: The weighted output values of a given rule are computed for each node in this layer, which is defined as the purification layer. Output value of node i in layer 5 using Eq. 3.

$$y_i^5 = \overline{\mu_i}\,[p_i x_i + q_i x_2 + r_i]\,, \qquad (i = 1, \dots, n) \qquad (3)$$

Layer 6: The last level layer, defined as the sum layer, has only one node and each node is labeled with Σ. The output value of each node in layer 5 is summed. The output value is determined as the actual value of the ANFIS system. Equation 4 is used to calculate the output value y of the system.

$$y = \sum_{i=1}^{n} \overline{\mu_i}\,[p_i x_i + q_i x_2 + r_i] \qquad (4)$$

2.2. Particle Swarm Optimization (PSO)

Animal herds in nature have unique structures with scattered structures that interact with each other. In herds, each individual exhibits a specific purposeful behavior. The task that each individual in the herd undertakes to achieve the goal reveals the tendency to act collectively. Collective intelligence, viewed from the outside, results from the actions that individuals repeat among themselves. Each member of the herd intuitively applies simple rules to the product of the collective intelligence of his actions. Thus the herd eventually achieves its goal.

Community in itself leads to self-organization in a sense in all group activities. When each individual, which we can call a representative or particle, performs an action, it calculates the fitness value of its position separately from the other individuals of the herd. It is important that the individual computes its position, keeps track of the speed at which it is progressing in each dimension of the solution set, the optimal fitness value it has achieved so far, the best positions to its neighbors, and finally the position at which it can catch up with this value. Only in this case can the herd reach the goal (Blum, National, and Li 2008). As a population-based probabilistic optimization method, the PSO technique is generally used to solve multivariate optimization problems.

The system starts by generating random solutions and then searches for the optimal solution by updating the generations of the populations. Unlike classical optimization techniques, this technique does not require any inferred knowledge. This has the advantage of minimizing the complex and difficult operations involved in solving many optimization problems. On the other hand, the fact that the number of parameters is smaller compared to many classical optimization methods makes the method easier to use (Ozkaya and Seyfi, 2018). The PSO formula is in Eq. 5

$$V_{id} = W * V_{id} + c_1 * rand_1 * (P_{id} - X_{id}) + c_2 * rand_2 * \left(P_{gd} - X_{id}\right) \qquad (5)$$

$$X_{id} = X_{id} + V_{id} \qquad (6)$$

V_{id} indicates the position value of the particle and V_{id} indicates the velocity values of the particle, while the values $rand_1$ and $rand_2$ are randomly derived numbers. W is the inertia weight value and c_1 and c_2 are the scaling factors.

Table 1. Basic PSO parameters

Parameters	**Descriptions**
Number of particles (swarm size; n)	Generally a value between 20-40 is taken.
Size of particles (d)	This parameter indicates the number of variables in the problem.
Learning factors	c1 and c2 in the formula stand for learning factors. The use of two learning factors contributes to generally good results.
Termination condition (termination criteria)	Adding a criterion to the algorithm that indicates at which point in the process of finding a solution it should stop
Number of iterations	Achieving the optimal solution in solving the problem depends on the number of iterations.
Particle Distance	The optimal result in solving the problem is achieved by identifying particles of different sizes and distances.
V_max	Determines the maximum velocity of the particles as a result of iteration in the process of solving the problem.

Particle swarm optimization is an optimization method first developed by Kenedy and Eberhart in 1995, inspired by fish and insects that move in swarms (James Kennedy and Russell Eberhart 1995). Basically, the algorithm based on herd intelligence allows animals moving in a herd to more easily achieve their goals in situations such as food and safety, which they often show randomly. In the PSO method, the exchange of information between individuals is crucial. The swarm is made up of individuals, in other words, particles. Each particle in the swarm adjusts its position according to the best position, using its previous experience. The positions of the particles in the swarm, which are fundamental in PSO, are based on the approach of the swarm to the individual with the best position. The rate of approach of each particle in the swarm is a random situation, and most of the time the individuals in the swarm are in a better position in their new movements. This process is an ongoing process until the goal is reached (James Kennedy and Russell Eberhart 1995). Fig. 3 shows the flow chart of particle swarm optimization.

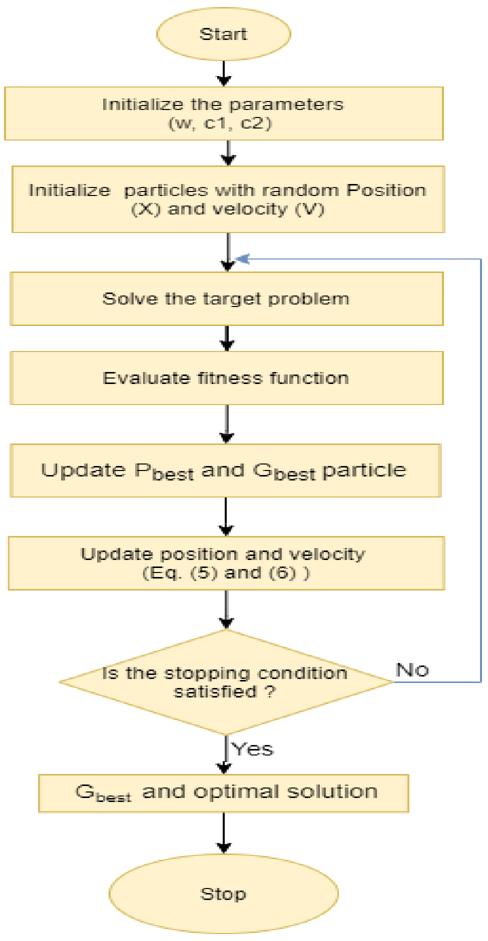

Start
Initialize the parameters (w, c1, c2)
Initialize particles with random Position (X) and velocity (V)
Solve the target problem
Evaluate fitness function
Update P_{best} and G_{best} particle
Update position and velocity (Eq. (5) and (6))
Is the stopping condition satisfied ?
No
Yes
G_{best} and optimal solution
Stop

Fig. 3. Flowchart of particle swarm optimization

The algorithm essentially consists of the following steps;

(i) Randomly generated starting positions and velocities form the initial swarm

(ii) The degree of fulfilment of all particles in the swarm is computed

(iii) For each particle, the best (pbest) from the current generation to the ground is found. The number of bests in the swarm is equal to the number of particles.

iv) The global best (gbest) is selected among the best of the current generation.

v) The positions and velocities are renewed as follows.

Steps 2,...,5 are repeated until the termination criterion is met.

In the PSO method, particles are adaptively directed to the most meaningful region in the search space by using the social interaction of the individuals in the swarm, i.e., the particles themselves. Initially, a swarm of particles is started with a random solution and updates are used to try to find an optimal solution. In each iteration, the position of the particles is updated relative to the two best particles. The particle with the best fitness value so far is called the local best (pbest) and its information should be stored in memory. The other particle is the particle that provides the best fitness value of all particles in the swarm and is called the global best (gbest). The particle matrix of the population looks as follows.

$$P = \begin{bmatrix} P_{11} & P_{12} & \dots & \dots & P_{1D} \\ P_{21} & P_{22} & \dots & \dots & P_{2D} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ P_{N1} & P_{N2} & \dots & \dots & P_{ND} \end{bmatrix}_{NxD}$$

In the above matrix, particle i is given as $P_i = [P_{i1}, P_{i2}, \dots\dots, P_{iD}]$, while in previous iterations, the position of particle i that gives the best fitness value is $P_{ye} = \left[P_{ye_{i1}}, P_{ye_{i2}}, \dots\dots, P_{ye_{iD}}\right]$. The position of the ith particle is renamed to the name of the local best particle. Gbest is unique for all particles in the repetition, and the velocity of the particle is called $V_i = [V_{i1}, V_{i2}, \dots\dots, V_{iD}]$. After the value of the two best particles in the swarm is determined, the velocities and positions of the particles are updated according to Equations 7 and 8 below.

$$V_i^{k+1} = V_i^k + c_1 r_1^k\left(p_{ye_i}^k - p_i^k\right) + c_2 r_2^k\left(g_{best}^k - p_i^k\right) \quad (7)$$

$$p_i^{k+1} = p_i^k + V_i^{k+1} \quad (8)$$

The values c1 and c2 in the 4th equation are the learning factors as well as constants expressing the acceleration terms pulling each particle towards its local best (pbest) and the best of the swarm (gbest). c1 allows the particle to move according to its own experience, while c2 allows the particle to move according to the experience of the other particles in the swarm. Choosing lower values allows particles to move far away from the target area before being pulled in its direction. This may lengthen the time it takes the herd to reach the goal. On the other hand, if the values are high, the herd may reach the goal sooner, and in this case unexpected movements are inevitable. Skipping the target area can be given as an example. The numbers r1 and r2 in the equations are random numbers with a uniform distribution between 0 and 1. K indicates the number of iterations. Each row in the matrix is called a particle and each particle has its own solution. For n particles, n solutions are obtained and the particles move in the research space according to two important parameters. Before each repetition, the best solution of the individual pbest and the best solution of the herd gbest are determined. In this case, n pbest and gbest values are determined. In the initial phase, the position and velocity values of each particle in the swarm are available. With each iteration, the particles get a little closer to the optimal solution.

3. The Proposed Methodology

The structure shown in Fig.1 represents the general structure of the proposed system. In this structure, PSO was used to adapt the parameters of the ANFIS model. ANFIS training is a classical hybrid optimization algorithm consisting of a combination of two algorithms, least squares and gradient descent. In this traditional hybrid method, gradient descent is used as a means of backpropagation to change membership settings, while least squares error is supported to change parameters in forward transmission. The proposed method, which is more flexible and faster than gradient-based approaches, can be used to improve ANFIS parameters in terms of discovery and exploitation capacities. The position of each particle in the PSO reflects a complete set of parameters for the ANFIS system. Due to the computational effort required for the fitting process, the total number of modifiable ANFIS parameters is an important factor in ANFIS networking. The overall operating cycle of the proposed system is shown in Fig. 4.

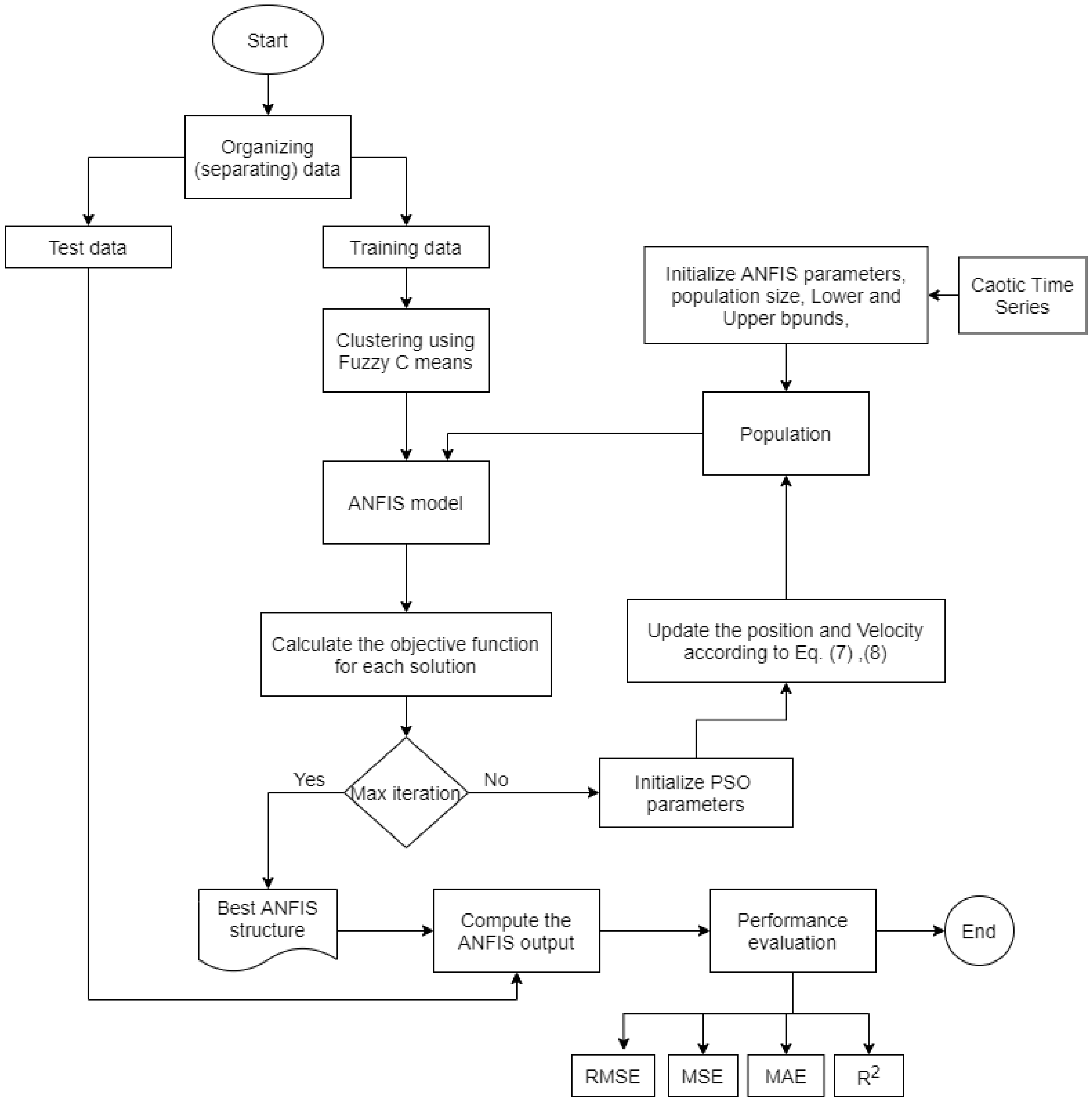

Start
Organizing (separating) data
Test data
Training data
Clustering using Fuzzy C means
Initialize ANFIS parameters, population size, Lower and Upper bpunds,
Caotic Time Series
Population
ANFIS model
Calculate the objective function for each solution
Update the position and Velocity according to Eq. (7) ,(8)
Yes
Max iteration
No
Initialize PSO parameters
Best ANFIS structure
Compute the ANFIS output
Performance evaluation
End
RMSE
MSE
MAE
R2

Fig. 4 The Proposed ANFIS with PSO algorithm

The model starts by splitting the data into training and testing data. In order to maintain the appropriate diversity of the population and improve the global search capacity, the basic properties of chaotic motions such as randomness and regularity are effective so that the algorithms do not fall into the problem of local optimum. It is necessary and important to calculate the accuracy of the system with the training and testing data of the data.

4. Results and Discussion

The prediction of chaotic time series was estimated using a ANFIS-PSO approach. Matlab software was used to build our model. In this analysis, several performance measures were used to control for ANFIS-PSO effectiveness. The results of mean square error (MSE), root mean square error (RMSE), R-Squared (R^2) and Mean Absolute Error (MEA) were obtained to demonstrate the effectiveness of the proposed method ANFIS-PSO and to verify the efficiency of the solutions' performance. The metrics MSE, MAE, RMSE and R-squared are mainly used to evaluate the prediction error rates and model performance in regression analysis. MAE is used to indicate the difference between the original and predicted values obtained by averaging the absolute difference over the data set. MSE is used to indicate the difference between the original and predicted values obtained by squaring the mean difference over the data set. The error rate obtained from the square root of the MSE is the RMSE. The formula R-squared (coefficient of determination) shows how well the values agree with the original values. A value between 0 and 1 is interpreted as a percentage. The higher the value, the better the model. The formulas used are given below (Eq. 9-12).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2 \quad (9)$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2} \quad (10)$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}| \quad (11)$$

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2} \quad (12)$$

Where n data points denotes, $\hat{y}$ denotes the predicted values of $\bar{y}$ denotes the mean values of y.

In this experiment, ANFIS-PSO was compared to many machine learning algorithms. For all experiments, the dataset was split into 50% for training and the rest for testing. The parameters of all optimization algorithms are: Population size (n) = 80, maximum iteration = 1200, lower bound = -10, upper bound = 10. In this experiment, four metrics, namely, root mean square error (RMSE), R-Squared (R2), Mean Absolute Error (MEA), and mean square error (MSE) were used to optimize the ANFIS parameters to evaluate the proposed ANFIS+PSO model. The tests were performed 10 times and all measurements are shown as average values in Table 6 and Figure 8, respectively. From these tables, we can see that ANFIS-PSO outperforms the other algorithms. This experiment proves the success of our proposal and can be used to improve the parameter optimization process of ANFIS.

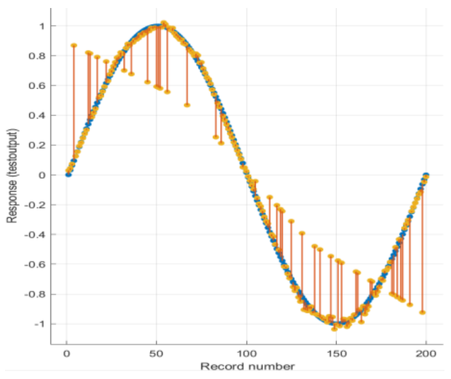
Table 2. Parameter Setting

Algorithm	Parameters	Values
ANFIS	Error goal	0
	Initial step	0,015
	Maximum epochs	100
ANFIS+PSO	C1	1
	C2	2
	Vmax	0,9
	Vmin	0,2
	W	1
	Wdamp	0.99
	maxit	1200
	npop	80

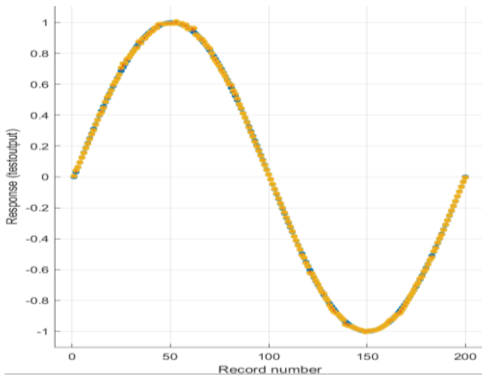
Table 3. The experimental results of MSE, MAE, RMSE and R2 using different algorithms

Algorithms		Training	Testing
Narrow Neural Network	RMSE	0.03085	0.17219
	R2	1.00	0.94
	MSE	0.00095171	0.029651
	MAE	0.02045	0.081603
Medium Neural Network	RMSE	0.0057177	0.0066921
	R2	1.00	1.00
	MSE	3.2692e-05	4.4784e-05
	MAE	0.0042401	0.0048332
Wide Neural Network	RMSE	0.0017565	0.0017477
	R2	1.00	1.00
	MSE	3.0851e-06	3.0545e-06
	MAE	0.0012522	0.0011796
Bilayered Neural Network	RMSE	0.11326	0.014002
	R2	0.97	1.00
	MSE	0.012828	0.00019607
	MAE	0.035032	0.01036
Trilayered Neural Network	RMSE	0.027383	0.086437
	R2	1.00	0.99
	MSE	0.00074983	0.0074713
	MAE	0.013395	0.023972
Linear SVM	RMSE	0.47997	0.46067
	R2	0.54	0.58
	MSE	0.23037	0.21221
	MAE	0.40933	0.39193
Quadratic SVM	RMSE	0.49914	0.4649
	R2	0.50	0.57

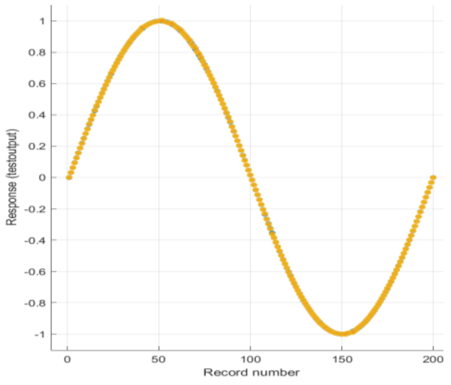
	MSE	0.24914	0.21613
	MAE	0.41771	0.39577
Cubic SVM	RMSE	0.074091	0.076584
	R2	0.99	0.99
	MSE	0.0054895	0.0058652
	MAE	0.066766	0.068535
Fine Gaussian SVM	RMSE	0.095271	0.094911
	R2	0.98	0.98
	MSE	0.0090765	0.0090082
	MAE	0.091844	0.09153
Medium Gaussian SVM	RMSE	0.075573	0.075522
	R2	0.99	0.99
	MSE	0.0057113	0.0057035
	MAE	0.068611	0.068688
Coarse Gaussian SVM	RMSE	0.42828	0.408
	R2	0.63	0.67
	MSE	0.18343	0.16646
	MAE	0.36588	0.35036
ANFIS_PSO	RMSE	0.00020483	0.0002125
	R2	1.00	1.00
	MSE	4.1954e-08	4.5157e-08
	MAE	8.1915e-05	8.3767e-05



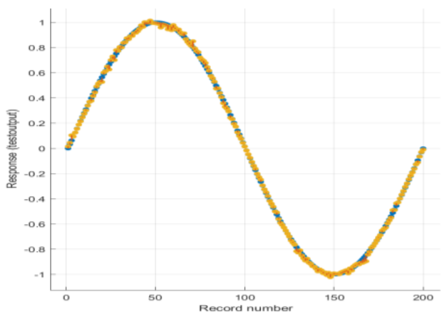
Narrow Neural Network



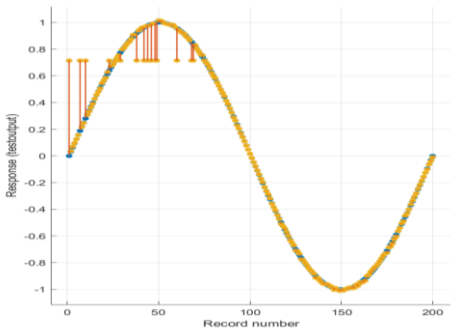
Medium Neural Network



Wide Neural Network



Bilayered Neural Network



Trilayered Neural Network

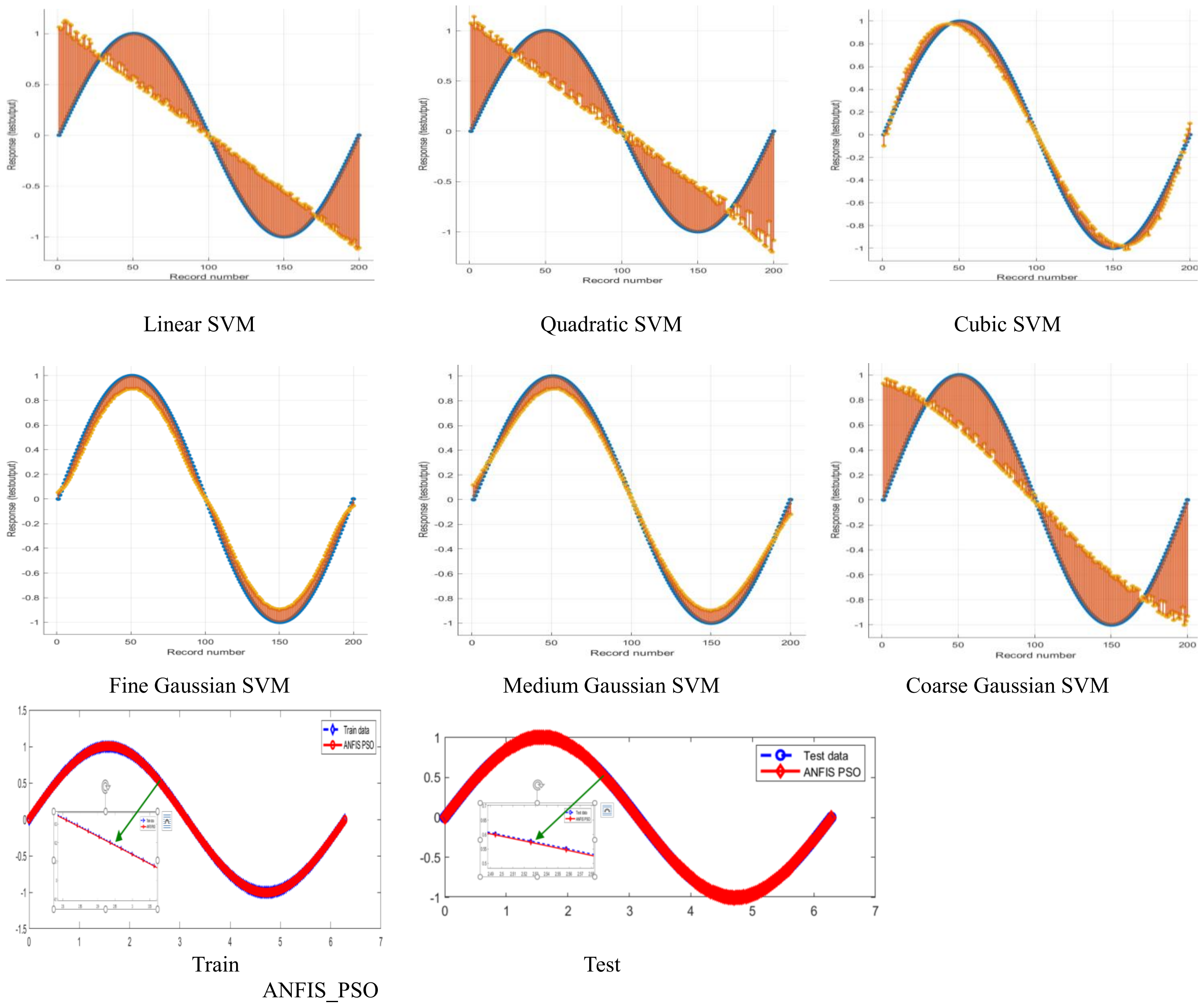


Fig. 5. Illustration of the experimental result for MSE, MAE, RMSE and R2 using different algorithms.

The response of data given above is illustrated in Fig. 5. The proposed model achieved the lowest error rate and the highest accuracy during the training and testing phase. Moreover, the results were compared with many machine learning algorithms. The results are presented using the performance values of the proposed model in terms of RMSE, R2, MEA, and MSE.

5. Conclusion and Future Work

This paper presents a model for using learning algorithms to obtain reliable decision support systems with time series data. ANFIS+PSO model is proposed to use time series feature prediction sets. Using PSO with ANFIS for feature selection and training the entire ANFIS structure provides optimal estimation. The proposed model has been evaluated against various metrics to confirm its effectiveness. The proposed model yielded good results compared to other algorithms. The proposed algorithm provides a way to improve the evolving structure of an ANFIS and achieve the goal of both higher accuracy and minimal computational overhead. The performance evaluation metrics of the proposed algorithm compared to other algorithms show the effectiveness of the proposed method. The proposed model will be extended in the future by applying contemporary metaheuristic algorithms to a wider range of datasets for other machine learning algorithms.

References

Basser, Hossein et al. 2015. “Hybrid ANFIS-PSO Approach for Predicting Optimum Parameters of a Protective Spur Dike.” Applied Soft Computing 30: 642–49. http://dx.doi.org/10.1016/j.asoc.2015.02.011.

Blum, Christian, Spanish National, and Xiaodong Li. 2008. Swarm Intelligence Swarm Intelligence.

El-Hasnony, Ibrahim M., Sherif I. Barakat, and Reham R. Mostafa. 2020. “Optimized ANFIS Model Using Hybrid Metaheuristic Algorithms for Parkinson’s Disease Prediction in IoT Environment.” IEEE Access 8: 119252–70.

Guillaume, S. 2001. “Designing Fuzzy Inference Systems from Data: An Interpretability-Oriented Review.” IEEE Transactions on Fuzzy Systems 9(3): 426–43.

Hodzic, Adnan. 2016. “A Novel Approach for Face Recognition Based on ANN and ANFIS.” (September 2015).

James Kennedy and Russell Eberhart. 1995. "Particle Swarm Optimisation." Proc. of the IEEE Int. Conference on Neural Networks 4: 1942–48.

Kundapura, Suman, and Arkal Vittal Hegde. 2021. "PSO-ANFIS Hybrid Approach for Prediction of Wave Reflection Coefficient for Semicircular Breakwater." ISH Journal of Hydraulic Engineering 27(2): 135–43. https://doi.org/10.1080/09715010.2018.1525688.

Ozkaya, U., and Seyfi, L. 2018. "A comparative study on parameters of leaf-shaped patch antenna using hybrid artificial intelligence network model"s. Neural Computing and Applications, 29(8), 35-45.

Rini, Dian Palupi, Siti Mariyam Shamsuddin, and Siti Sophiayati Yuhaniz. 2016. "Particle Swarm Optimization for ANFIS Interpretability and Accuracy." Soft Computing 20(1): 251–62. http://dx.doi.org/10.1007/s00500-014-1498-z.

Sada, S. O., and S. C. Ikpeseni. 2021. "Evaluation of ANN and ANFIS Modeling Ability in the Prediction of AISI 1050 Steel Machining Performance." Heliyon 7(2): e06136. https://doi.org/10.1016/j.heliyon.2021.e06136.

Shamshirband, Shahaboddin et al. 2019. "Developing an ANFIS-PSO Model to Predict Mercury Emissions in Combustion Flue Gases." Mathematics 7(10).

Yaseen, Zaher Mundher et al. 2019. "Novel Hybrid Data-Intelligence Model for Forecasting Monthly Rainfall with Uncertainty Analysis." Water 2019, Vol. 11, Page 502 11(3): 502. https://www.mdpi.com/2073-4441/11/3/502/htm (October 6, 2021).

Zadeh, L A. 1994. "Soft Computing and Fuzzy Logic." IEEE Software 11(6): 48–56.

Zhang, Guoqiang, B. Eddy Patuwo, and Michael Y. Hu. 1998. "Forecasting with Artificial Neural Networks: The State of the Art." International Journal of Forecasting 14(1): 35–62.