Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 47 (2026), No. 2, pp. 487–506
DOI : 10.47974/JIOS-1571

Switching regression analysis for data with outlier using ANFIS trained by GA and PSO

Aynur Yonar
Department of Statistics
Selçuk University
Faculty of Science
Konya
Turkey

Abstract

Classical regression analysis assumes that the data belong to a single class. However, in many application studies, data are typically gathered from various mixed classes without prior knowledge of each class's membership. Switching regression analysis, which creates suitable sub-models for each different class, is used to model such data. As the quantity of independent variables and classes within the data increases, the number of sub-models to be generated simultaneously in switching regression also increases. In such scenarios, it is recommended to utilize the Adaptive Network-based Fuzzy Inference System (ANFIS), which proves to be an effective tool for addressing mixed problems and systems.

Achieving successful training is essential for obtaining effective results with ANFIS. This can be achieved by determining its structure's premise and consequence parameters with practical optimization algorithms. This study focuses on training ANFIS with Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). The effectiveness of the GA-trained ANFIS and PSO-trained ANFIS are tested with a numerical example where the dependent variable has an outlier. The results show that the proposed methods, especially PSO-trained ANFIS, provide accurate predictions and are not affected by the outlier in the dependent variable.

Subject Classification (2010): *62J99, 68T05, 68T20, 62A86.*

Keywords: *ANFIS, GA, PSO, Machine learning, Artificial intelligence, Switching regression.*

ORCID-ID: **0000-0003-1681-9398**

E-mail: **aynursahin@selcuk.edu.tr**

1. Introduction

In regression analysis, it is typically presumed the data originate from a single class. The functional relationship between the independent (input) variables and the dependent (output) variables are mathematically defined as:

$$Y = f(X;\theta) + \varepsilon \tag{1}$$

where Y is the dependent variable, $f(\boldsymbol{X};\boldsymbol{\theta})$ is a model function, $\boldsymbol{X} = (X_1, X_2, \ldots X_k)$ is a vector of independent variables; $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_p)'$, $\boldsymbol{\theta} \in \Theta \subset \Re^p$ is an unknown parameter vector; $\boldsymbol{\varepsilon}$ is the random error term with $\varepsilon(0,\sigma^2)$. The aim is to estimate the $\boldsymbol{\theta}$ with considered data set and to predict dependent variable values.

In many application studies, independent variables in the data come from different and mixed classes, i.e. they do not have the same distribution or the same distribution parameters. In such cases, the estimates obtained with a single regression model for all data will likely have high errors. The "switching regression model" has been proposed to deal with this problem [1-6]. In switching regression, an appropriate sub-model is created for each different class and the estimated outcomes of the dependent variable are derived by integrating the predictions from these sub-models. A switching regression model is expressed as follows:

$$Y_L = f_L(\boldsymbol{X};\boldsymbol{\theta}_L) + \varepsilon_L, \quad L = \prod_{k=1}^{p} l_k \tag{2}$$

Here, $\boldsymbol{\theta}_L$ is the parameter set for f_L, l_k is the class number for each independent variable and p is the number of independent variables. In this model, it is evident that when the number of classes and the number of independent variables increases, the number of sub-models to be created also increases. In such cases, it can be suggested to use the neuro-fuzzy, which is highly effective in solving mixed problems and systems, for establishing the switching regression model [1, 7, 8]

Neuro-fuzzy is an advanced artificial intelligence model that integrates the principles of fuzzy logic and artificial neural networks (ANN) to create systems capable of adaptive learning and human-like reasoning. Fuzzy logic, developed by Zadeh [9], is a mathematical framework designed to handle uncertainty, imprecision, and vagueness in data. Unlike traditional binary logic, which operates on absolute true or false values (0 or 1), fuzzy logic introduces the concept of partial truths, where values can range between 0 and 1. This allows fuzzy logic to model and process information in a way that closely resembles human reasoning, especially in situations where definitive answers are not possible. ANN, on the other hand, is computational model inspired by the structure and functioning of the human brain. It excels in tasks like pattern recognition, learning from data, and making predictions by adjusting its internal parameters through training. By combining these two paradigms, neuro-fuzzy models bring together the interpretability and reasoning strengths of fuzzy logic with the adaptive learning and self-optimization capabilities of neural networks. This hybrid approach enables the creation of systems that can handle complex, nonlinear problems while providing transparent and explainable decision-making processes.

The Adaptive Network-based Fuzzy Inference System (ANFIS), introduced by Jang [10], is considered one of the most prominent and widely recognized neuro-fuzzy models [11]. This is a non-hierarchical hybrid network structure that embodies the Sugeno fuzzy inference system [12]. In recent years, there have been many studies in fuzzy regression analysis with ANFIS. Cheng and Lee [13] applied a fuzzy adaptive network to fuzzy regression analysis using numerical examples. Then, Cheng and Lee [1] introduced a nonparametric switching regression method utilizing the Sugeno fuzzy inference system and fuzzy adaptive network. Dalkılıç and Apaydın [8] employed ANFIS for the analysis of switching regression in scenarios where the independent variables are drawn from an exponential distribution. Dalkılıç, Kula [7] performed a switching regression analysis where the dependent variable included an outlier, comparing the results obtained through ANFIS with those from robust methods which are Huber, Hampel, Andrews, and Tukey. They concluded that ANFIS is the best among these methods. Danesh, Farnoosh [14] used ANFIS for fuzzy nonparametric regression with crisp input and fuzzy output. Danesh [15] proposed two hybrid algorithms based on ANFIS to predict the fuzzy regression model, and they verified that the proposed algorithms have better performance than linear programming and quadratic programming. Razzaghnia [16] employed ANFIS to obtain the predictions in regression models where the data set has outliers and showed that the ANFIS is better than linear programming and fuzzy weights-based linear programming for outlier cases.

The effectiveness of ANFIS largely depends on the optimization algorithm utilized in the learning phase. In the structure of ANFIS, both premise and consequent parameters are optimized through an optimization algorithm throughout the training phase. In the standard ANFIS model, premise parameters are typically established through the gradient descent method, while consequence parameters are evaluated through the least square estimation method. However, a reliable optimization algorithm is necessary to ensure optimal performance in ANFIS. Hence, there is a growing tendency to utilize heuristic algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Differential Evolution (DE), Simulated Annealing (SA), and others for training ANFIS in recent years [17-23].

This study aims to perform a switching regression analysis in a case where there is an outlier in the dependent variable, by ANFIS algorithms trained with GA and PSO, which are well-known artificial intelligence techniques, and to evaluate the effectiveness of the GA-trained ANFIS and the PSO-trained ANFIS, which are proposed in this study, with the standard ANFIS.

The remainder of the paper is structured as follows: In Section 2, artificial intelligence techniques: ANFIS, GA, and PSO algorithms are presented. Section 3 introduces the training process of ANFIS with GA and PSO. In Section 4, proposed GA-trained ANFIS and PSO-trained ANFIS are conducted for switching regression modeling in a numerical example. In the final section, conclusions are presented.

2. Artificial Intelligence Techniques

This section offers a concise summary of the artificial intelligence techniques utilized in this study, including the ANFIS, GA, and PSO. Furthermore, it outlines the training process of ANFIS using GA and PSO.

2.1 *Adaptive Network Fuzzy Inference System (ANFIS)*

ANFIS, proposed by Jang [10], is widely recognized as a prominent artificial intelligence methodologies. It originated from the concept that combines the strengths of ANN and fuzzy logic. ANN derives decision-making capabilities from fuzzy logic, while fuzzy logic gains learning abilities and relational structure from ANN. Various types of ANFIS have been developed based on the specific fuzzy inference systems including Mamdani, Sugeno, and Tsumoto. In this study, the ANFIS structure adopts the Sugeno fuzzy inference system proposed by Takagi and Sugeno [24], making it highly suitable for modeling regression with fuzzy inputs derived from diverse distributions. The fuzzy rule-based Sugeno system is defined as follows:

$$R^i : \text{if } (x_1 \text{ is } F_1^i, \text{and } x_2 \; is \; F_2^i, ..., \text{and } x_p \; is \; F_p^i) \text{ then } (y = f^i = a_0^i + a_1^i x_1 + ... + a_p^i x_p)$$

Here, $\boldsymbol{x} = (x_1, x_2, ..., x_p)'$ is the input vector, y is the output, F_i^i is a fuzzy set associated with the input x_i in the i th rule, f^i is the system output for R^i rule, $i = 1, 2, ..., k$, and k is the fuzzy rule number. The parameters which characterize the membership functions for the input variables are called premise parameters, whereas the parameters that define the consequent part of the fuzzy rules are known as consequent parameters.

The parameters used to define the membership function for F_j^i are termed premise parameters, while those used for a_j^i are called consequence parameters.

During the training of ANFIS, both the premise and consequence parameters are optimized using an algorithm. This process relies on the available input-output relationship and if-then fuzzy conditions. The discrepancy in the result from training and the actual system result creates an error signal. For reducing these errors, the parameters of ANFIS are iteratively adjusted, resulting in the development of an optimal structure.

The ANFIS structure depicted in Figure 1 consists of two inputs (x_1, x_2) and one output.

The ANFIS structure comprises 5 layers. Below is the explanation of the layer structure depicted in Figure 1.

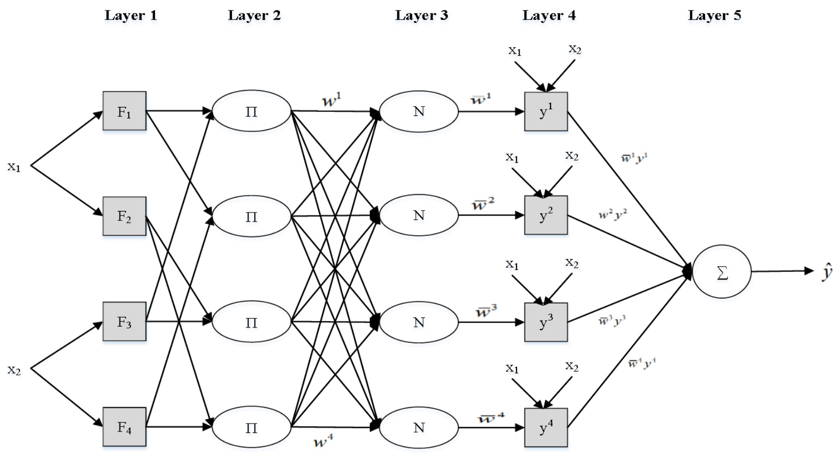

Figure 1

The ANFIS structure with two inputs and one output

Layer 1: In the fuzzification layer, various membership functions, including the generalized bell function, triangle, trapezium, Gaussian, sigmoid, etc., are employed to generate fuzzy clusters from input values. For this study, the Gaussian membership function is utilized. Each node's output in this layer represents a membership degree, which is linked to the input values and membership functions as follows:

$$O_{1,i} = \begin{cases} \mu_{F_i}(x_1) = e^{-\frac{1}{2}\left(\frac{x_1 - c_i}{\sigma_i}\right)^2}, \ i = 1,2 \\ \mu_{F_i}(x_2) = e^{-\frac{1}{2}\left(\frac{x_2 - c_i}{\sigma_i}\right)^2}, \ i = 3,4 \end{cases} \tag{3}$$

Here, μ_{F_i} is the membership degrees for the fuzzy set parameters F_i. c_i and σ_i are the parameters which determine the shapes of Gaussian membership function. These parameters are referred to as the premise or antecedent parameters.

Layer 2: In the rule layer, the firing strengths (w^i) are determined by utilizing the membership values obtained from the fuzzification layer. Every node in this layer represents the rules generated by the fuzzy logic inference system. The firing strengths w^i of each rule are calculated by multiplying the associated membership values, as demonstrated below:

$$\begin{aligned} O_{2,1} &= w^1 = \mu_{F_1}(x_1) \times \mu_{F_3}(x_2) \\ O_{2,2} &= w^2 = \mu_{F_1}(x_1) \times \mu_{F_4}(x_2) \\ O_{2,3} &= w^3 = \mu_{F_2}(x_1) \times \mu_{F_3}(x_2) \\ O_{2,4} &= w^4 = \mu_{F_2}(x_1) \times \mu_{F_4}(x_2) \end{aligned} \tag{4}$$

Layer 3: Within the normalization layer, the normalized firing strengths $(\overline{w}^i)$ of all rules are computed based on the firing strengths obtained from the preceding layer. The scaled firing strength of the i-th rule is calculated through normalizing its firing strength with respect to the sum of all firing strengths, as expressed in the following formula:

$$O_{3,i} = \overline{w}^i = \frac{w^i}{\sum_{i=1}^{4} w^i}, \ i = 1,2,3,4 \tag{5}$$

Layer 4: In the defuzzification layer, the output of every rule is computed through multiplying the normalized firing strengths with a linear function. The calculation of the outputs of the nodes in this layer is described as follows:

$$O_{4,i} = \overline{w}^i y^i = \overline{w}^i (a_0^i + a_1^i x_1 + a_2^i x_2),\ i = 1, 2, 3, 4 \tag{6}$$

Here, $a_0^i, a_1^i,$ and a_2^i are the parameters in the linear function. The above-mentioned parameters are known as the conclusion or consequent parameters.

Layer 5: This layer, referred to as the summation layer, determines the actual output of ANFIS by aggregating the signals received from the defuzzification layer. This layer consists of only one node, and its output is calculated as follows:

$$O_{5,1} = \hat{y} = \sum_{i=1}^{4} \overline{w}^i y^i = \frac{\sum_{i=1}^{4} w^i y^i}{\sum_{i=1}^{4} w^i} \tag{7}$$

2.2 *Genetic Algorithm (GA)*

GA was initially introduced by Holland [25] to gain insight into the process of evolution in natural systems, inspired by Darwin's theory of natural selection. Subsequently, Goldberg and Holland [26] adapted it for applications in optimization and machine learning. As one of the most widely recognized population-based artificial intelligence methods, the GA is designed to find approximate solutions to complex optimization problems, particularly when exact solutions are computationally expensive or infeasible to obtain.

The GA begins by randomly generating an initial population within the solution space, where each individual in the population represents a potential solution to the problem. The algorithm then iteratively evolves this population through processes inspired by biological evolution, including selection, mutation, and crossover, to enhance the quality of the solutions over successive generations [27, 28]. These operations allow the GA to explore the solution space efficiently and converge towards the best possible solutions.

At each iteration, the individuals in the population are assessed using a fitness function, which measures their quality or how well they solve the optimization problem. Selection mechanisms then determine which individuals will be used to create the next generation. These selected individuals are combined and modified through crossover and mutation to produce new candidate solutions, with the goal of enhancing the population's overall fitness.

The pseudocode for the GA algorithm, which outlines its step-by-step procedure, is provided in Table 1.

Table 1

Pseudocode for GA

Set the GA parameters: mutation probability (*MP*), crossover probability (*CR*), selection rate (*SR*), particle size (*npop*), and define the stopping criterion such as maximum iteration or minimum error.
Create the initial population.
Assess the fitness of each individual in the population
do
Choose the top-performing individuals to be utilized by genetic operators.
Create new individuals by applying the crossover and mutation operators.
Assess the fitness of newly generated individuals.
Substitute the best newly generated individuals for the worst-performing individuals in the population.
while stopping criterion is not satisfied

2.3 *Particle swarm optimization (PSO)*

PSO was first proposed by Kennedy and Eberhart [29] and later refined by Eberhart and Kennedy [30]. It is an optimization technique inspired by biological systems, specifically drawing principles from the group interaction observed in bird flocking and fish schooling. The PSO is a population-based and self-adaptive artificial intelligence optimization method. Each particle in the population maintains its own personal best position (*pbest*) and is aware of the global best position (*gbest*), which represents the unique best position discovered by all particles in the swarm. In each iteration, they update their velocities and positions considering this knowledge.

The velocity and the updated position of every individual at iteration g+1, are given by the following formulas:

$$v_i^{g+1} = wv_i^g + c_1 r_1 (pbest_i^g - x_i^g) + c_2 r_2 (gbest^g - x_i^g) \tag{8}$$

$$x_i^{g+1} = x_i^g + v_i^g \tag{9}$$

Here, v_i^g represents the velocity of individual i at iteration g, w denotes the inertia weight, while c_1 and c_2 are the acceleration coefficients, r_1 and r_2 are randomized values uniformly distributed between 0 and 1, x_i^g refers to the position of individual i at iteration g, $pbest_i^g$ is the best position of individual i up to the iteration g, $gbest^g$ is the best position of the group until iteration g[27, 28]. The pseudo-code for the PSO algorithm can be found in Table 2.

Table 2

Pseudo-code of the PSO algorithm

```
Set the PSO parameters: inertia weight (w), acceleration coefficients (c1 and c2),
particle size (npop), and define the stopping criterion.
Generate the initial population
do
    for each individual in the population
        Evaluate fitness score
        If the current fitness score exceeds the previous personal best (pbest)
            Update the pbest with the current value
        end
    end
    Identify the individual with the highest fitness as the global best (gbest).
    for every individual in the population
        Compute the velocity based on Equation (8)
        Adjust the position using Equation (9)
    end
while stopping criterion is not satisfied
```

3. Training of ANFIS using GA and PSO

The training process of ANFIS is crucial for achieving effective results with the model. Training of ANFIS means finding out the premise and consequent parameters within its structure via an optimization algorithm. These parameters must be determined to minimize errors, which represent the disparity between the output (prediction) obtained from the network and the actual output of the system. In the first ANFIS model by Jang [10], a derivative-based hybrid learning algorithm used Gradient Descent (GD) method for determining the premise parameters and Least Square

Estimation (LSE) method to detect the consequence parameters, which was suggested as a training algorithm. Indeed, there is a possibility of encountering local optima when using derivative-based optimization algorithms like Gradient Descent (GD) and Least Squares Estimation (LSE) to determine unknown parameters. This risk arises due to the initial value problem associated with these algorithms. So, metaheuristic methods, such as GA, PSO, DE, ABC, etc., have recently been widely used to enhance the effectiveness of ANFIS learning. In this study, GA and PSO algorithms are considered in the training of ANFIS due to the advantages such as ease of use, high convergence speeds, and effective results.

Premise parameters expressed by c_i and σ_i belong to the Gaussian membership functions in the fuzzification layer. Consequent parameters denoted by $a_0^i, a_1^i,$ and a_2^i are those used in the defuzzification layer. The number of premise parameters equals the sum of the parameters in all membership functions. For example, there will be eight premise parameters in total if it is assumed that the gauss membership function is used in the ANFIS structure in Figure 1. There are also 12 consequent parameters in total because each rule has three parameters. Consequently, this structure in Figure 1 has 20 parameters that need to be optimized by GA or PSO.

The method proposed in this study is applied to switching regression analysis. The objective is to minimize the discrepancies between the actual output values and the predicted output values obtained from ANFIS. Therefore, the optimal model is developed by adjusting the ANFIS parameters using the GA and PSO algorithms to minimize the Mean Squared Error (MSE) value. The MSE value is calculated as follows:

$$MSE = \frac{\sum_{j=1}^{n} (y_j - \hat{y}_j)^2}{n} \tag{10}$$

Here, y_j are the actual output values, $\hat{y}_j$ are prediction output values by ANFIS and n represents the sample size.

4. Numerical Example

In this section, the data set sourced from the study conducted by Dalkılıç, Kula [7] is used to show the performance of the proposed trained-ANFIS methods for switching regression. This data set, which has three independent variables, and one dependent variable, is presented in Table 3. They generated the values of the data set from the normal distribution such that $X_1 \sim N$ $(\mu = 20,\ \sigma^2 = 9)$, $X_2 \sim N$ $(\mu = 50,\ \sigma^2 = 144)$ $X_3 \sim N$

$(\mu = 32,\ \sigma^2 = 169)$ and dependent variable values Y depend on these variables. Furthermore, an outlier was added to the data by replacing the 15th observation of the dependent variable with $y_{15} + 50$. They also gave the regression models, and predictions for this data set in comparison with ANFIS, LSM, and some robust regression methods: Huber, Hampel, Tukey, and Andrews, and concluded that ANFIS was the best among them.

Table 3

Data set

No	x_1	x_2	x_3	y	No	x_1	x_2	x_3	y
1	21.8101	50.5397	49.8319	125.4057	**16**	25.2815	50.2143	54.2714	128.9194
2	19.8248	78.9993	35.1925	137.4526	**17**	20.2663	30.6749	40.9755	60.9541
3	16.6740	46.2813	33.5450	97.0719	**18**	27.7867	64.8650	33.4679	130.0444
4	26.4327	52.2510	37.0010	116.3851	**19**	17.9736	58.2030	17.8756	87.9184
5	15.9415	61.3724	31.0880	107.0015	**20**	28.3604	40.6314	11.7420	78.4568
6	21.3711	43.6916	24.4820	92.0224	**21**	19.9495	56.3718	40.2863	116.6304
7	21.1735	36.6127	38.1010	100.600	**22**	20.8150	75.6140	26.7405	118.1328
8	26.2190	30.8922	48.8959	90.8950	**23**	17.2577	54.2523	26.7568	103.9698
9	19.0300	64.0981	53.2524	136.546	**24**	14.1459	52.7804	33.0930	101.7193
10	24.4044	55.8217	22.8635	104.741	**25**	19.0477	65.4558	26.3405	113.7832
11	18.4928	69.7458	42.4943	133.625	**26**	21.7650	49.8381	24.6859	93.9008
12	20.6288	44.5492	18.6431	83.1755	**27**	22.4870	33.9999	43.4148	101.4928
13	22.2644	62.1052	48.8284	134.887	**28**	14.9754	43.3230	21.4096	85.7995
14	17.1554	74.5928	32.1941	126.0083	**29**	14.2331	59.0672	28.6413	105.1197
15	21.8395	57.2242	34.8432	166.4707	**30**	18.6900	39.0578	38.4129	99.1197

x_1, x_2 and x_3 given in Table 3 are taken as inputs of the proposed ANFIS structures. Gaussian membership functions are used to achieve fuzzy clusters from the values of inputs. The number of fuzzy classes for each cluster and the number of fuzzy inference rules are determined as 8. Thus, in this study, a total of 72 parameters, consisting of 48 premises and 24 consequences parameters, are optimized by GA or PSO.

By the GA-trained ANFIS and the PSO-trained ANFIS, regression models and predictions for this data set are achieved separately. The proposed algorithms: GA-trained ANFIS and PSO-trained ANFIS are implemented using a program written in MATLAB. In GA and PSO,

Table 4

Parameter values for the GA and PSO algorithms

Algorithm	Parameters
GA	$MP = 0.7$, $CR = 0.4$, $SR = 0.15$, $npop = 50$ and $Maxit = 1000$
PSO	$w = 1$, $c_1 = 1$ and $c_2 = 2$, $npop = 50$ and $Maxit = 1000$

search spaces are selected as [-25,25] for all unknown parameters, and unique parameter values of the algorithms are determined based on previous studies in the literature and trials, as shown in Table 4.

Models obtained via GA-trained ANFIS and PSO-trained ANFIS structures are given in Equation (11) and Equation (12), respectively.

$$\begin{aligned}
\hat{y}_1 &= 232.4729 + 12.9824x_1 + 27.4832x_2 - 23.5151x_3 \\
\hat{y}_2 &= -242.7042 + 24.8479x_1 + 29.7926x_2 + 23.5151x_3 \\
\hat{y}_3 &= 196.1690 - 26.0893x_1 - 26.4049x_2 + 20.7355x_3 \\
\hat{y}_4 &= -73.0630 + 11.7479x_1 + 1.1341x_2 - 2.4368x_3 \\
\hat{y}_5 &= -262.0264 - 26.0893x_1 + 11.3735x_2 + 20.1876x_3 \\
\hat{y}_6 &= -13.5312 + 1.0436x_1 + 1.2434x_2 + 0.9406x_3 \\
\hat{y}_7 &= 264.2421 - 6.1209x_1 + 2.0338x_2 - 9.2491x_3 \\
\hat{y}_8 &= 264.2421 + 1.1880x_1 + 17.0850x_2 + 18.3293x_3
\end{aligned} \tag{11}$$

and

$$\begin{aligned}
\hat{y}_1 &= -4.4102 + 1.1949x_1 + 1.9069x_2 + 1.1666x_3 \\
\hat{y}_2 &= 3.8587 + 0.8839x_1 + 0.8910x_2 + 1.1436x_3 \\
\hat{y}_3 &= -6.8429 + 1.4008x_1 + 0.8054x_2 + 1.3781x_3 \\
\hat{y}_4 &= 16.3552 + 0.7971x_1 + 1.3477x_2 + 0.0925x_3 \\
\hat{y}_5 &= -14.1715 + 1.0913x_1 - 0.1439x_2 - 3.0761x_3 \\
\hat{y}_6 &= 16.7945 - 0.0615x_1 + 1.2261x_2 + 0.7718x_3 \\
\hat{y}_7 &= -11.5324 + 1.0576x_1 + 1.7858x_2 + 0.7697x_3 \\
\hat{y}_8 &= -11.7834 + 0.9884x_1 + 1.2259x_2 + 0.3680x_3
\end{aligned} \tag{12}$$

The weights of the observation obtained from the network belonging to the models in Equation (11), and Equation (12) are given in Table 5 and 6, respectively. These weights, also called membership degrees of observation, represent the effects of the observations on more than one model.

Table 5

The degree of membership of the observation to each model in Equation (11)

No	w^1	w^2	w^3	w^4	w^5	w^6	w^7	w^8
1	0.9939	0.9984	0.9513	0.3663	0.1264	0.3446	0.9253	0.9737
2	0.9854	0.9992	0.9920	0.2162	0.1477	0.2998	0.9070	0.9637
3	0.9943	0.9984	0.9427	0.4950	0.1854	0.4953	0.9204	0.9766
4	0.9938	0.9982	0.9541	0.2730	0.0866	0.2364	0.9290	0.9717
5	0.9906	0.9989	0.9722	0.3850	0.1955	0.4598	0.9112	0.9716
6	0.9953	0.9981	0.9360	0.4189	0.1307	0.3716	0.9283	0.9761
7	0.9965	0.9979	0.9180	0.4676	0.1326	0.3909	0.9317	0.9783
8	0.9977	0.9974	0.9012	0.3780	0.0879	0.2702	0.9398	0.9783
9	0.9901	0.9989	0.9759	0.3207	0.1566	0.3722	0.9143	0.9698
10	0.9928	0.9984	0.9614	0.2888	0.1027	0.2718	0.9253	0.9712
11	0.9884	0.9991	0.9834	0.2890	0.1630	0.3637	0.9104	0.9678
12	0.9950	0.9982	0.9382	0.4287	0.1384	0.3895	0.9269	0.9760
13	0.9910	0.9987	0.9724	0.2863	0.1221	0.3033	0.9196	0.9696
14	0.9867	0.9992	0.9888	0.2715	0.1797	0.3740	0.9057	0.9662
15	0.9923	0.9986	0.9643	0.3232	0.1262	0.3266	0.9217	0.9715
16	0.9942	0.9982	0.9501	0.3037	0.0954	0.2642	0.9291	0.9728
17	0.9973	0.9977	0.9015	0.5251	0.1420	0.4260	0.9334	0.9802
18	0.9905	0.9985	0.9756	0.1945	0.0772	0.1885	0.9232	0.9668
19	0.9916	0.9988	0.9666	0.3793	0.1691	0.4196	0.9161	0.9722
20	0.9963	0.9977	0.9273	0.2904	0.0731	0.2147	0.9366	0.9748
21	0.9923	0.9986	0.9631	0.3604	0.1460	0.3752	0.9198	0.9723
22	0.9867	0.9991	0.9891	0.2252	0.1370	0.2920	0.9102	0.9648
23	0.9926	0.9987	0.9595	0.4213	0.1780	0.4525	0.9171	0.9738
24	0.9926	0.9987	0.9570	0.4840	0.2208	0.5434	0.9127	0.9751
25	0.9897	0.9989	0.9778	0.3111	0.1564	0.3669	0.9136	0.9693
26	0.9940	0.9983	0.9499	0.3716	0.1269	0.3474	0.9256	0.9740
27	0.9970	0.9977	0.9107	0.4519	0.1196	0.3594	0.9346	0.9786
28	0.9947	0.9983	0.9360	0.5512	0.2085	0.5541	0.9191	0.9780
29	0.9910	0.9989	0.9686	0.4279	0.2197	0.5141	0.9095	0.9729
30	0.9959	0.9981	0.9248	0.5079	0.1601	0.4564	0.9271	0.9783

Table 6

The degree of membership of the observation to each model in Equation (12)

No	w^1	w^2	w^3	w^4	w^5	w^6	w^7	w^8
1	0.4481	0.3710	0.4047	0.3148	0.7208	0.6084	0.3204	0.3153
2	0.2664	0.1478	0.3077	0.3356	0.5615	0.5563	0.3929	0.1882
3	0.5810	0.4855	0.5674	0.5160	0.8015	0.7589	0.5242	0.4527
4	0.3488	0.2945	0.2826	0.1805	0.6521	0.4755	0.1828	0.2222
5	0.4589	0.3216	0.4937	0.5159	0.7173	0.7344	0.5553	0.3581
6	0.5051	0.4465	0.4460	0.3370	0.7606	0.6309	0.3361	0.3599
7	0.5555	0.5211	0.4795	0.3501	0.7935	0.6402	0.3434	0.3977
8	0.4632	0.4720	0.3468	0.1963	0.7411	0.4930	0.1883	0.2978
9	0.3904	0.2691	0.4060	0.3926	0.6705	0.6452	0.4250	0.2867
10	0.3636	0.2896	0.3176	0.2298	0.6600	0.5244	0.2370	0.2414
11	0.3519	0.2233	0.3855	0.4007	0.6376	0.6345	0.4465	0.2591
12	0.5149	0.4491	0.4637	0.3628	0.7658	0.6513	0.3634	0.3722
13	0.3563	0.2581	0.3414	0.2860	0.6490	0.5660	0.3045	0.2459
14	0.3285	0.1922	0.3833	0.4376	0.6150	0.6423	0.5023	0.2461
15	0.3989	0.3062	0.3740	0.3059	0.6831	0.5921	0.3192	0.2791
16	0.3827	0.3270	0.3167	0.2107	0.6778	0.5106	0.2124	0.2503
17	0.6114	0.5951	0.5288	0.3881	0.8263	0.6640	0.3773	0.4454
18	0.2553	0.1868	0.2153	0.1425	0.5670	0.4128	0.1517	0.1558
19	0.4561	0.3367	0.4650	0.4432	0.7189	0.6944	0.4685	0.3439
20	0.3713	0.3579	0.2692	0.1439	0.6758	0.4348	0.1400	0.2276
21	0.4381	0.3351	0.4255	0.3724	0.7095	0.6467	0.3889	0.3186
22	0.2793	0.1648	0.3067	0.3105	0.5760	0.5494	0.3555	0.1946
23	0.5020	0.3867	0.5061	0.4786	0.7503	0.7254	0.4989	0.3851
24	0.5662	0.4357	0.5927	0.6077	0.7879	0.8076	0.6344	0.4597
25	0.3791	0.2566	0.3980	0.3894	0.6614	0.6393	0.4242	0.2778
26	0.4540	0.3786	0.4090	0.3171	0.7251	0.6109	0.3220	0.3199
27	0.5393	0.5213	0.4483	0.3058	0.7855	0.6016	0.2975	0.3759
28	0.6380	0.5431	0.6308	0.5918	0.8336	0.8070	0.5984	0.5131
29	0.5046	0.3607	0.5477	0.5891	0.7469	0.7841	0.6302	0.4069
30	0.5957	0.5385	0.5453	0.4431	0.8143	0.7109	0.4395	0.4478

Table 7

Comparison results of for ANFIS, GA-trained ANFIS and PSO-trained ANFIS

No	ANFIS Predictions	ANFIS Residuals	GA-trained ANFIS Predic-tions	GA-trained ANFIS Residuals	PSO-trained ANFIS Predic-tions	PSO-trained ANFIS Residuals
1	143.0816	-17.6759	118.9459	6.4598	125.4191	-0.0134
2	145.278	-7.8254	138.4881	-1.0355	137.4454	0.0072
3	98.6598	-1.5879	93.1207	3.9512	97.0742	-0.0023
4	127.9457	-11.5606	115.1861	1.1990	116.3741	0.0110
5	109.26754	-2.26604	108.5475	-1.5460	107.9977	-0.9962
6	90.9193	1.1031	85.4363	6.5861	89.6803	2.3421
7	108.2847	-7.6847	99.0900	1.5100	100.4708	0.1292
8	119.8321	-28.9371	89.8000	1.0950	90.8192	0.0758
9	133.7845	2.7615	136.0170	0.5290	136.6069	-0.0609
10	110.9347	-6.1937	102.4591	2.2819	104.7126	0.0284
11	134.6763	-1.0513	130.2349	3.3901	133.7941	-0.1691
12	81.9083	1.2672	79.8267	3.3488	83.1070	0.0685
13	148.258	-13.3710	133.7292	1.1578	134.8800	0.0070
14	130.3885	-4.3802	127.3917	-1.3834	126.0470	-0.0387
15	176.6868	-10.2161	161.3711	5.0996	166.4929	-0.0222
16	158.6018	-29.6824	126.2937	2.6257	128.9255	-0.0061
17	70.2880	-9.3339	78.5075	-17.5534	60.9505	0.0036
18	141.4112	-11.3668	127.4083	2.6361	130.0507	-0.0063
19	93.0661	-5.1477	94.5057	-6.5873	89.8522	-1.9338
20	67.4576	10.9992	67.2022	11.2546	78.5576	-0.1008
21	124.3031	-7.6727	126.8776	-10.2472	116.6369	-0.0065
22	129.6829	-11.5501	127.3390	-9.2062	120.0854	-1.9526
23	107.2308	-3.2610	99.8413	4.1285	103.8315	0.1383
24	101.6517	0.0676	97.2942	4.4251	101.8252	-0.1059
25	121.7088	-7.9256	113.6494	0.1338	108.8576	4.9256
26	96.8150	-2.9142	97.2723	-3.3715	95.9536	-2.0528
27	117.8637	-16.3709	91.2149	10.2779	101.4816	0.0112
28	84.9727	0.8268	75.0824	10.7171	85.7034	0.0961
29	106.4569	-1.3372	101.5546	3.5651	105.1321	-0.0124
30	104.2006	-5.0809	94.1036	5.0161	99.5886	-0.4689
MSE		119.3650		38.4903		1.4279

The predictions and residual results for considered data obtained using the PSO-trained ANFIS and GA-trained ANFIS are demonstrated in Table 7. For comparison, the results for ANFIS reported by Dalkılıç, Kula [7] are quoted. It is clear from Table 7 that GA and PSO-trained ANFIS methods provide much better prediction results than the standard ANFIS method concerning the MSE criterion. In particular, PSO-trained ANFIS displays a superior performance with prediction values very close to real values and the smallest MSE value. Dalkılıç, Kula [7] showed that ANFIS is better than LSE and some robust regression methods: Huber, Hampel, Tukey, and Andrews in their study. Consequently, it can be concluded that the PSO-trained ANFIS is the most effective and robust method compared to all the other methods. Furthermore, the graphs of the dependent variables and the residuals' actual and predicted values separately for the GA-trained ANFIS and the PSO-trained ANFIS are shown in Figure 2 and 3. These graphs also support the above results. Therefore, using PSO-trained ANFIS for such cases with outliers can be highly recommended.

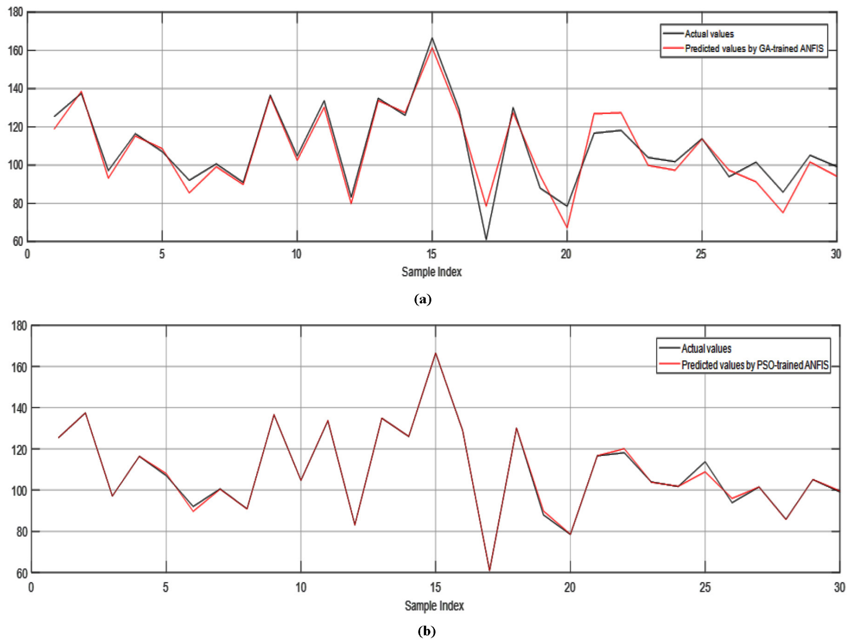

Figure 2

Actual and predicted values of dependent variable by GA-trained ANFIS (a) and PSO-trained ANFIS (b)

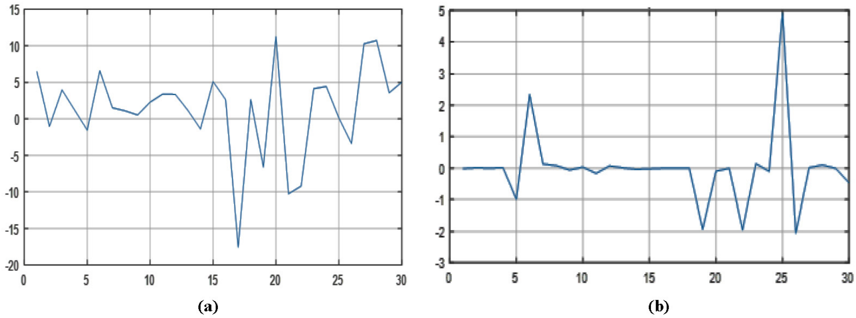

Figure 3

Residuals by GA-trained ANFIS (a) and PSO-trained ANFIS (b)

5. Conclusion

In this study, the switching regression analysis utilized in the cases where data are collected from different and mixed classes is considered by ANFIS. Successful results with ANFIS depend on its training, which means determining the premise and consequence parameters in its structure through an optimization algorithm. Thus, we proposed using GA and PSO algorithms to train the ANFIS to establish the switching regression analysis. A numerical example is analyzed to show the performance of the proposed GA-trained ANFIS and PSO-trained ANFIS structures on switching regression analysis in the dependent variable with an outlier. The results show that GA-trained ANFIS and PSO-trained ANFIS structures provide more accurate predictions with small MSE values than standard ANFIS. The PSO-trained ANFIS structure gives the best predictions with the smallest MSE among them. Moreover, the proposed algorithms are robust since they are not affected by outliers in the dependent variable. Therefore, it can be said that proposed trained ANFIS structures, particularly PSO-trained ANFIS, are effective and satisfactory in establishing the switching regression analysis for the data with an outlier.

In future work, further investigation can be carried out to explore the potential of hybridizing other optimization techniques with ANFIS for more specialized tasks. Additionally, the performance of these models can be assessed on larger datasets and in different domains, such as time series forecasting or classification problems, to evaluate their generalizability and robustness.

References

[1] C.-B. Cheng and E. S. Lee, "Switching regression analysis by fuzzy adaptive network," *European Journal of Operational Research*, vol. 128, no. 3, pp. 647-663 (2001).

[2] M. Menard, "Fuzzy clustering and switching regression models using ambiguity and distance rejects," *Fuzzy Sets and Systems*, vol. 122, no. 3, pp. 363-399 (2001).

[3] R. J. Hathaway and J. C. Bezdek, "Switching regression models and fuzzy clustering," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 3, pp. 195-204 (1993).

[4] R. E. Quandt, "A new approach to estimating switching regressions," *Journal of the American Statistical Association*, vol. 67, no. 338, pp. 306-310 (1972).

[5] R. E. Quandt and J. B. Ramsey, "Estimating mixtures of normal distributions and switching regressions," *Journal of the American Statistical Association*, vol. 73, no. 364, pp. 730-738 (1978).

[6] L.-F. Lee and R. H. Porter, "Switching Regression Models with Imperfect Sample Separation Information—With an Application on Cartel Stability," *Econometrica: Journal of the Econometric Society*, pp. 391-418 (1984).

[7] T. E. Dalkılıç, K. Ş. Kula, and A. Apaydın, "Parameter estimation by ANFIS where dependent variable has outlier," *Hacettepe Journal of Mathematics and Statistics*, vol. 43, no. 2, pp. 309-322 (2014).

[8] T. E. Dalkılıç and A. Apaydın, "A fuzzy adaptive network approach to parameter estimation in cases where independent variables come from an exponential distribution," *Journal of Computational and Applied Mathematics*, vol. 233, no. 1, pp. 36-45 (2009).

[9] L. A. Zadeh, "Fuzzy sets," Information and Control, vol. 8, no. 3, pp. 338-353, 1965.

[10] J.-S. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665-685 (1993).

[11] D. Karaboga and E. Kaya, "Adaptive network based fuzzy inference system (ANFIS) training approaches: a comprehensive survey," Artificial Intelligence Review, vol. 52, no. 4, pp. 2263-2293 (2019).

[12] I. K. Omurlu, F. Cantas, M. Ture, and H. Ozturk, "An empirical study on performances of multilayer perceptron, logistic regression, ANFIS, KNN and bagging CART," *Journal of Statistics Management Systems*, vol. 23, no. 4, pp. 827-841 (2020).

[13] C.-B. Cheng and E. S. Lee, "Applying fuzzy adaptive network to fuzzy regression analysis," *Computers Mathematics with Applications*, vol. 38, no. 2, pp. 123-140 (1999).

[14] S. Danesh, R. Farnoosh, and T. Razzaghnia, "Fuzzy nonparametric regression based on an adaptive neuro-fuzzy inference system," *Neurocomputing*, vol. 173, pp. 1450-1460 (2016).

[15] S. Danesh, "Fuzzy parameters estimation via hybrid methods," *Hacettepe Journal of Mathematics Statistics*, vol. 10 (2018).

[16] T. Razzaghnia, "Regression parameters prediction in data set with outliers using neural network," *Hacettepe Journal of Mathematics Statistics*, vol. 48, no. 4, pp. 1170-1184 (2019).

[17] D. Karaboga and E. Kaya, "Estimation of number of foreign visitors with ANFIS by using ABC algorithm," *Soft Computing*, vol. 1-13 (2019).

[18] D. Karaboga and E. Kaya, "Training ANFIS by using an adaptive and hybrid artificial bee colony algorithm (aABC) for the identification of nonlinear static systems," *Arabian Journal for Science Engineering*, vol. 44, no. 4, pp. 3531-3547 (2019).

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

[20] W.-T. Pan, "Forecasting classification of operating performance of enterprises by ZSCORE combining ANFIS and genetic algorithm," *Neural Computing Applications*, vol. 18, no. 8, p. 1005 (2009).

[21] H. Yang, M. Hasanipanah, M. Tahir, and D. T. Bui, "Intelligent prediction of blasting-induced ground vibration using ANFIS optimized by GA and PSO," *Natural Resources Research*, vol. 1-12 (2019).

[22] A. Khosravi, S. Nahavandi, and D. Creighton, "Prediction interval construction and optimization for adaptive neurofuzzy inference systems," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 5, pp. 983-988 (2011).

[23] S. F. Sheniha, S. S. Priyadharsini, and S. E. Rajan, "Removal of artifact from EEG signal using differential evolution algorithm," in 2013 International Conference on Communication and Signal Processing, *IEEE* (2013).

[24] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics,* no. 1, pp. 116-132 (1985).

[25] J. Holland, Adaptation in natural and artificial systems: an introductory analysis with application to biology, *Control Artificial Intelligence Review* (1975).

[26] D. E. Goldberg and J. H. Holland, Genetic algorithms and machine learning, Kluwer Academic Publisher (1988).

[27] E.-G. Talbi, Metaheuristics: from design to implementation, vol. 74, John Wiley & Sons (2009).

[28] X.-S. Yang, Nature-inspired metaheuristic algorithms, Luniver Press (2010).

[29] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proceedings of ICNN'95—International Conference on Neural Networks, *IEEE* (1995).

[30] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, *IEEE* (1995).

Received April, 2023