



# The Training of Pi-Sigma Artificial Neural Networks with Differential Evolution Algorithm for Forecasting

Oguzhan Yilmaz<sup>1</sup> · Eren Bas<sup>1</sup> · Erol Egrioglu<sup>1</sup>

Accepted: 20 December 2020 / Published online: 3 March 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC part of Springer Nature 2021

## Abstract

Looking at the artificial neural networks' literature, most of the studies started with feedforward artificial neural networks and the training of many feedforward artificial neural networks models are performed with derivative-based algorithms such as levenberg–marquardt and back-propagation learning algorithms in the first studies. In recent years, although many new heuristic algorithms have been proposed for different aims these heuristic algorithms are also frequently used in the training process of many different artificial neural network models. Pi-sigma artificial neural networks have different importance than many artificial neural network models with its higher-order network structure and superior forecasting performance. In this study, the training of Pi-Sigma artificial neural networks is performed by differential evolution algorithm uses DE/rand/1 mutation strategy. The performance of the proposed method is evaluated by two data sets and seen that the proposed method has a very effective performance compared with many artificial neural network models.

**Keywords** Pi-Sigma artificial neural networks · Differential Evolution Algorithm · Higher-order neural networks · Forecasting

## 1 Introduction

Artificial neural networks, one of the most widely used methods of artificial intelligence, is a very simple imitation of the human nervous system. The most basic element of the human nervous system is nerve cells called neurons. McCulloch and

---

✉ Eren Bas  
eren.bas@giresun.edu.tr

Oguzhan Yilmaz  
oguzhanylmz677@gmail.com

Erol Egrioglu  
erol.egrioglu@giresun.edu.tr

<sup>1</sup> Department of Statistics, Faculty of Arts and Science, Giresun University, 28200 Giresun, Turkey

Pitts in (1943) set the artificial mathematical model of a biological neuron for the first time and this model underlies the various artificial neural network model even today. Generally, many artificial neural networks have three layers as input, hidden and output layer and these layers reveal the architecture of an artificial neural network. Besides the running process of each layer differs from an artificial neural network model according to an artificial neural model. Artificial neural networks have been extensively used for pattern recognition, clustering, classification and forecasting task in recent years. Feedforward neural networks (FFNNs) are the popular form of artificial neural network models that can perceive and approximate computational models using their advanced parallel layered structure (Ojha et al. in 2017). Multi-layer perceptron (MLP) is a special form of FFNNs. Unlike the single perceptron model, MLP has the hidden layer(s) structure. The input layer receives the incoming data and sends it to the hidden layer. Then the incoming information is transferred to the next layer. The number of hidden layers varies depending on the problem, at least one and adjusted according to the need or the problem structure. The output of each layer becomes the input of the next layer. In artificial neural networks literature, MLP models have been successfully used and applied in time series forecasting. MLP is employing one or several hidden layers in the artificial neural networks and generally, many FFNNs models including MLP work with additive neuron model (units). Besides, some artificial neural networks use multiplicative neuron model such as multiplicative neuron model artificial neural networks proposed by Yadav et al. in (Yadav et al. 2007). MLP is a type 1st order neural network which can effectively carry out inner products which are then weighted and summed before passing through the non-linear threshold function. The other way to overcome the restriction to linear maps is to introduce higher-order units to model nonlinear dependences (Giles and Maxwell in (Giles and Maxwell 1987), Giles et al. in (Giles et al. 1988)).

Higher-order neural networks (HONNs) are a type of FFNNs which provide non-linear decision boundaries, therefore offering a better classification capability than the linear neuron (Guler and Sahin in 1994). HONNs use higher combinations of inputs. HONNs do not only contain additive or multiplicative units but contain both units at the same time. HONNs also allow the inputs to be used by duplicating them. Unlike many artificial neuron networks, Pi-Sigma artificial neural networks (PS-ANN) one of the HONNs proposed by Shin and Gosh in (1991) have a very different architecture than other artificial neural networks due to using both additive and multiplicative structure. In the literature, there are many studies about PS-ANN for different aims especially forecasting. Ghazali and Jumeily in (2009) used PS-ANN for financial time series prediction. Husaini et al. in (2014) used PS-ANN for temperature forecasting in Batu Pahat. Husaini et al. in (2011) showed the effect of network parameters of PS-ANN for temperature forecasting. Husaini et al. in (2012) used PS-ANN for a one-step-ahead temperature forecasting. Nayak et al. in (2015) proposed a novel chemical reaction optimization based on PS-ANN for nonlinear classification. Mohamed et al. in (Mohamed et al. 2016) used batch gradient method for training of PS-ANN with a penalty. Nayak in (2017) used PS-ANN based on genetic algorithm and particle swarm optimization for exchange rate prediction. Bas et al. in (2016) proposed a high order fuzzy time series method based on PS-ANN and determine the fuzzy relations with PS-ANN. Dash et al. in (2018) used a PS-ANN

based on evolutionary algorithms for gold price prediction. Deepa et al. in (2018) used a PS-ANN based on bioinspired swarm intelligence optimization algorithm for multimodal tumour data analysis. Akram et al. in (2019) proposed an improved PS-ANN with error feedback for physical time series prediction. Nayak in (2020) used a fireworks algorithm for the training of PS-ANN for modelling and forecasting chaotic crude oil price time series. Panda and Majhi in (2020) used improved spotted hyena optimizer with space transformational search for the training of PS-ANN. Kocak et al. in (2020) proposed a new fuzzy time series method based on an ARMA-type recurrent PS-ANN. Pattanayak et al. in (2020) proposed a multi-step-ahead fuzzy time series forecasting by using hybrid chemical reaction optimization with PS-ANN. Nayak and Ansari (2020) used cooperative optimization algorithm in PS-ANN for stock forecasting.

In this paper, the training of PS-ANN is performed by differential evolution algorithm (DEA) uses DE/rand/1 mutation strategy for time series forecasting. The performance of the proposed method (DEA-PS-ANN) is evaluated on two well-known data sets in ANN literature and compared with many studies in ANN literature. The rest of the paper is as follows: Section two is about PS-ANN and DEA. The proposed DEA-PS-ANN is given in section three. The application results are given in section four and finally, section five is about conclusions and discussions.

## 2 Preliminary Section

The main line of the study is about PS-ANN and DEA. So, this section contains basic information about PS-ANN and DEA.

### 2.1 Pi-Sigma Artificial Neural Networks

PS-ANN is a higher-order neural network type that uses higher combinations of its inputs and it was proposed by Shin and Gosh in (1991). PS-ANN consists of an input layer, a hidden layer of a linear sum of units, and finally an output layer of the multiplication of units. The term pi-sigma is that the units are processed throughout a sum and multiplicative process. Although the weights between the input and hidden layers are adjustable, the weights between hidden and output layers are fixed as 1. Increasing the degree of the Pi-Sigma neural network causes the function that defines the relationship between the input and the output layer to depend on more parameters and have a more complex structure. This allows to obtain better forecasting results. However, working with excess parameters causes an overfitting problem and more calculation time for the training algorithm. The architecture of a PS-ANN with an  $N$  input and  $K$ th order is given in Fig. 1. We would like to stress that the template should not be manipulated and that the guidelines regarding font sizes and format should be adhered to. This is to ensure that the end product is as homogeneous as possible.

The linear combinations of input units are obtained with the weights  $w_{ij}(i = 1, 2, \dots, N, j = 1, 2, \dots, K)$ . and biases of  $\theta_j(j = 1, 2, \dots, K)$ .  $w_{ij}$  is the weight

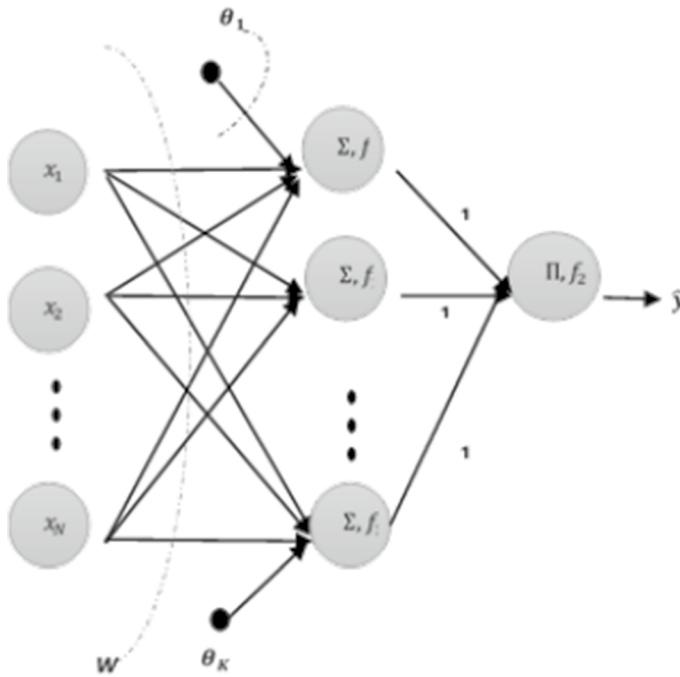


Fig. 1 The architecture of PS-ANN

from the  $i^{\text{th}}$  input to the  $j^{\text{th}}$  hidden layer unit. The linear combination as much as the number of hidden layer units passes through the linear activation function and creates the output of hidden layers.

## 2.2 Differential Evolution Algorithm

DEA is a simple, understandable and effective optimization algorithm compared to other evolutionary algorithms and it was proposed by Storn and Price in (1995). DEA is similar in that it uses the same evolutionary operators as a genetic algorithm, such as mutation, crossover, and selection to direct the population to an optimal solution. Chromosomes represent solutions in DEA process. DEA's performance is based on two main components: selected strategy and control parameters. However, the underlying idea of DEA is the mutation, crossover, and selection operators used to determine global optimum in every generation. Control parameter components consist of population size (PB), scaling factor (F) and crossover operators.

The mutation operator is the main distinguishing component of DEA and is considered as the main strategy of DEA. There are many different mutation strategies, each based on the classical mutation strategy and expressed in short definitions. The mutation strategy, called the classical mutation strategy and given with Eq. (1), is one of the most frequently used mutation strategies in the DEA literature

and is shown by DE/rand/1. In this notation, “DE” indicates the DEA, “rand” is randomness, that is, the chromosomes are selected randomly. Classical DEA contains three random chromosomes ( $c_1, c_2, c_3$ ) selected from the population. ( $c_1, c_2, c_3 \in [1, \dots, PB]; c_1 \neq c_2 \neq c_3$ ). With the mutation operator, these chromosomes come together and recombine to form the total vector or in other words mutation vector ( $c_{mv}$ ). The application of the mutation operator to all candidate chromosomes in the population defines a search rule based on other candidate solutions (chromosomes) as given in Eq. (1).

$$c_{mv} = c_3 + F \times (c_1 - c_2) \quad (1)$$

In the DEA process, a random chromosome called  $c_x$  which is different from  $c_1, c_2, c_3$  and also called as the target chromosome is selected and a new candidate chromosome ( $c_c$ ) is also selected for the population by intersecting this  $c_x$  and  $c_{mv}$  chromosomes and finally crossover operator is performed for the DEA as given in Eq. (2).

$$c_c(i, j) = \begin{cases} c_{mv}(i, j) & \text{rnd}(0, 1) < cr \\ c_x(i, j) & \text{otherwise} \end{cases} \quad (2)$$

When constructing  $c_c$ , random numbers must be generated as much as the number of genes in a chromosome. If each generated random number is smaller than the crossover rate ( $cr$ ), the gene is taken from the  $c_{mv}$  chromosome, if not, from the chromosome  $c_x$ , and the chromosome  $c_c$  is created.

Finally, the fitness values of  $c_c$  and  $c_x$  chromosomes are compared in the DEA process. If the fitness value of  $c_c$  is better than the fitness value of  $c_x$ , the candidate chromosome  $c_c$  will replace the target chromosome  $c_x$  in the next iteration, otherwise the target chromosome  $c_x$  will remain in the population. This process is called selection process in DEA. After the population is updated, mutation and crossover operators and selection process are repeated after reaching a predefined stop criterion, such as a certain number of iterations.

### 3 The Proposed Method

The forecasting of time series is made under some assumptions such as linearity and normal distribution such as autoregressive, moving averages models and etc. Considering that many time series that encounter in daily life are not linear so in the literature, several nonlinear models such as ANN have been proposed for forecasting of time series. PS-ANN, one of the most commonly used artificial neural network models, is a kind of high order ANN type use higher combinations of its inputs. The weight values determined depending on the number of inputs in PS-ANN do not increase exponentially unlike many other high order ANN types. So, since PS-ANN does not have a complex structure it is very useful compared to many other high order ANN types and has quite good analysis performance for time series prediction in the literature.

Genetic algorithm is known as the most frequently preferred and one of the oldest algorithms among artificial intelligence optimization algorithms. In a GA process, each chromosome in the population is not included in a GA process one by one. In contrast to GA process, each chromosome in the population is evaluated one by one during the DEA process, and thus, it is checked whether any chromosome can be transferred to the next generation or not. In short, it can be said that the most important feature of the DEA is that it evaluates each chromosome in the population within the DEA process. Thus, some chromosomes that are important for the population are preserved without any changes and transferred to the next generations.

In the proposed method, the training of the PS-ANN is performed with DEA. In the implementation process of DEA, DE/rand/1 mutation strategy is preferred as a mutation strategy. Moreover, with the use of an artificial intelligence optimization technique, DEA, a search space with multiple solution points is used instead of a random solution point. With the use of DEA in the training process of PS-ANN, using a derivative-based optimization algorithm such as LM has been prevented. Thus, it is avoided the use of complex derivatives. The algorithm of the proposed method (DEA-PS-ANN) is given in the following algorithm.

**Algorithm** The training of DEA-PS-ANN

*Step 1* Determining the parameters of PS-ANN used in the learning process.

These parameters are; the number of inputs ( $m$ ), the degree of PS-ANN ( $d$ ), the number of chromosomes ( $nc$ ), the number of genes ( $g$ ) in a chromosome, and the crossover rate ( $cr$ ).

*Step 2* Creating the initial population.

Before creating the initial population, let's identify any of the chromosomes that will correspond to each solution in DEA. During the learning process of PS-ANN, the genes of a chromosome of DEA consist of the weight and bias values of PS-ANN. A chromosome consisting of weight and bias values of PS-ANN is given in Fig. 2.

In Fig. 2, the genes from 1 to  $(m \times d)$  are weight values in a chromosome, and genes from  $((m \times d) + 1)$  to  $((m \times d) + d)$  are bias values in a chromosome. In total, there are  $((m \times d) + d)$  genes in a chromosome.

Therefore, in the creating of the initial population,  $nc$  chromosomes consisting of  $((m \times d) + d)$  genes are generated randomly with the help of continuous uniform distribution  $U(0, 1)$ .

*Step 3.* Calculation of the fitness function value for each chromosome.

Gene	1	...	$(m \times d)$	$(m \times d) + 1$	...	$(m \times d) + d$
Weight and Bias	$w_{11}$	...	$w_{md}$	$\theta_1$	...	$\theta_d$

**Fig. 2** A structure of a chromosome in DEA-PS-ANN

First, the output of the  $j$ th hidden layer unit for each hidden layer is calculated by the neural net signals given in Equation (3) and the output of the network is calculated by Equation (4).

$$h_j = f_1 \left( \sum_{i=1}^N w_{ij}x_i + \theta_j \right), \quad j = 1, 2, \dots, K \tag{3}$$

In Equation (3),  $f_1(x) = x$  shows the linear activation function. The output of the network is calculated by Equation (4) with the  $f_2(x) = \frac{1}{1+\exp(-x)}$  logistic activation function.

$$\hat{y} = f_2 \left( \prod_{j=1}^K h_j \right) = \frac{1}{1 + \exp \left( - \prod_{j=1}^K h_j \right)} \tag{4}$$

Finally, for each chromosome, the Root of Mean Square Error (RMSE) value determined as the fitness function given in Equation 5

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (x_t - \hat{x}_t)^2}{n}} \tag{5}$$

In Equation (5)  $n$ ,  $x_t$  and  $\hat{x}_t$  are the number of learning samples the observed values and the forecasting value respectively.

*Step 4. Application of mutation and crossover operators.*

Mutation and crossover operators are applied sequentially to each chromosome in the initial population by using Equations (1) and (2).

*Step 5. Comparison of fitness values.*

The fitness value of the candidate chromosome is compared with the fitness value of the target chromosome. The chromosome with the lowest RMSE value is transferred to the new generation. Mutation and crossover operators are applied to all chromosomes in the population.

*Step 6. Stopping condition.*

If the maximum number of iterations has been reached or the chromosome with the best fitness value is less than a predetermined error value ( $\epsilon$ ), the process is finished if not, go to Step 3.

When the algorithm stops, the optimal weight and bias values are obtained from the chromosome with the lowest RMSE value in the population.

## 4 Application

In the evaluation of the performance of the DEA-PS-ANN, Australian beer consumption(AUST) time series with observation values between the second quarter of 1957 and the first quarter of 1994 given in Fig. 3 was analyzed firstly.

The last 16 observations of the AUST time series were taken as the test set. In the analysis of the AUST time series, the number of inputs of the model was

tried from 4 to 12 with increasing 4, the degree of the model was tried by 2 to 5 with increasing 1, the crossover rate was tried from 0.1 to 0.5 with increasing 0.1 and the number of chromosomes was tried from 30 to 100 with increasing 10. Besides, the number of iterations was taken as 200. As a result of the trials, the best solution was obtained when the number of inputs is 12, the model degree is 4, the number of chromosomes is 40, and the crossover rate is 0.2.

Apart from the proposed DEA-PS-ANN, AUST time series was analyzed with single multiplicative neuron model artificial neural network (SMNM-ANN) based on BP learning algorithm (BP-SMNM-ANN) proposed by Yadav et al. in (2007), multilayer ANN based on PSO (PSO-ML-ANN) proposed by Kelwade and Salankar in (2017), Radial basis ANN (RB-ANN) proposed by Li in (2009), SMNM-ANN based on PSO (PSO-SMNM-ANN) proposed by Zhao and Yang in (2009), Multilayer feedforward ANN (ML-FF-ANN) proposed by Rumelhart et al. in (1986), Elman type recurrent ANN (E-ANN) proposed by Elman in (1990), multiplicative seasonal ANN based on PSO (PSO MS-ANN) proposed by Aladag et al. in (2013), multi-layer forward-feed ANN based on the trimmed mean neuron model (TR-ML-FF-ANN), PS-ANN based on PSO (PSO-PS-ANN), SMNM-ANN based on DEA (DEA-SMNM-ANN) proposed by Bas in (2016), linear and nonlinear ANN (L&NL-ANN) proposed by Yolcu et al. in (2013), recurrent SMNM-ANN (R-SMNM-ANN) proposed by Egrioglu et al. in (2017), SMNM-ANN based on artificial bat algorithm (ABA-SMNM-ANN) proposed by Bas et al. in (2018), SMNM-ANN with autoregressive coefficient (AC-SMNM-ANN) proposed by Cagcag Yolcu et al. in (2018), SMNM-ANN based on Gauss activation function (Gauss-SMNM-ANN) proposed by Gundogdu et al. in (2015), recurrent PS-ANN (R-PS-ANN) proposed by Akdeniz et al. in (2018) methods given in Table 1.

When Table 1 is examined, the proposed DEA-PS-ANN has the best performance among all methods in terms of both RMSE and MAPE criteria. Besides, the graph of the actual observations of the AUST test set and the forecasts obtained from the proposed DEA-PS-ANN for AUST test data are given in Fig. 4.

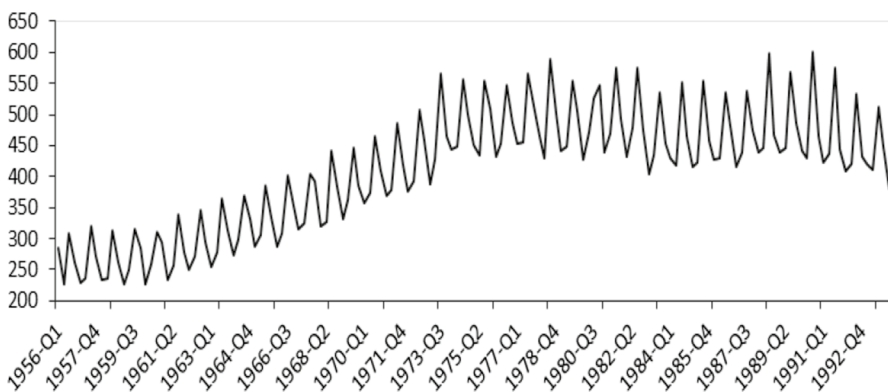
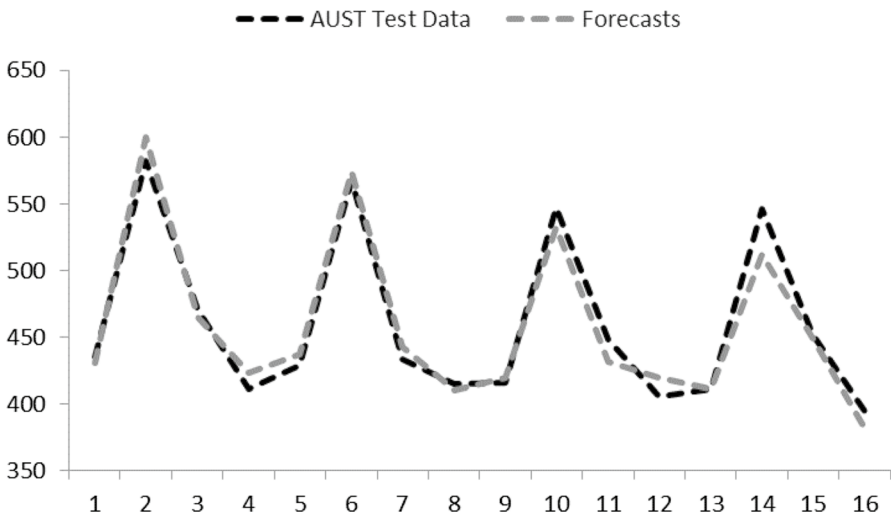


Fig. 3 AUST time series graph



**Table 1** RMSE And MAPE Values Obtained from All Methods for AUST Test Data

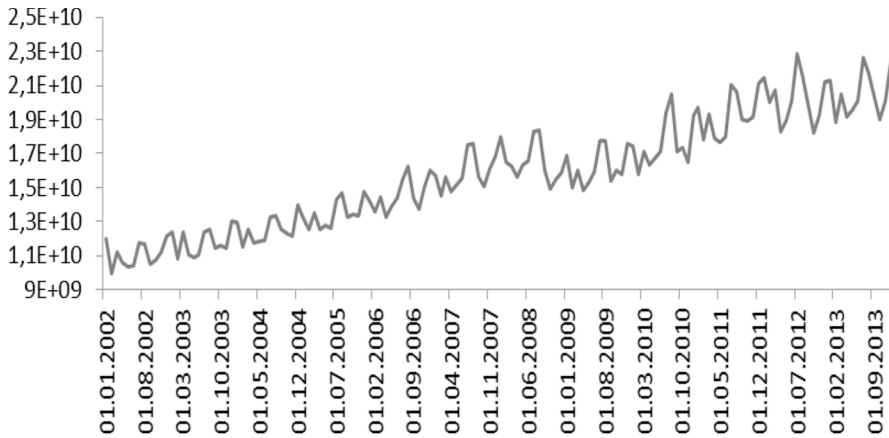
Methods	RMSE	MAPE
BP-SMNM-ANN	74,2551	0,0983
PSO-ML-ANN	44,7780	0,0856
RB-ANN	41,7000	0,0686
PSO-SMNM-ANN	26,7831	0,0489
ML-FF-ANN	24,1052	0,0476
E-YSA	22,6581	0,0436
PSO-MS-ANN	22,1700	0,0394
TR-ML-FF-ANN	21,0623	0,0399
PSO-PS-ANN	20,0886	0,0352
DEA-SMNM-ANN	19,7819	0,0372
L&NL-ANN	18,7888	0,0357
R-SMNM-ANN	17,8573	0,0329
ABA-SMNM-ANN	17,7054	0,0323
AC-SMNM-ANN	17,2930	0,0309
Gauss-SMNM-ANN	15,837	0,0300
R-PS-ANN	15,7100	0,0282
DEA-PS-ANN	13,1928	0,0226



**Fig. 4** AUST test set together with the forecasts obtained from the proposed DEA-PS-ANN

In Fig. 4, it is seen that the real observations of AUST test set are quite compatible with the forecasts obtained from the method proposed DEA-PS-ANN for AUST test data.

Secondly and finally Turkey electricity consumption (TEC) time series observed monthly between January 2002 and December 2013 given in Fig. 5 was analyzed.



**Fig. 5** TEC time series graph

The last 12 observations of the TEC time series were taken as the test set. In the analysis of the TEC time series, the number of inputs of the model was tried from 5 to 15 with increasing 1, the degree of the model was tried from 2 to 5 with increasing 1, the crossover rate was tried from 0.1 to 0.5 with increasing 0.1 and the number of chromosomes was tried from 30 to 100 with increasing 10. Besides, the number of iterations is taken as 200. As a result of the trials, the best solution was obtained when the number of inputs is 13, the model degree is 2, the number of chromosomes is 40, and the crossover rate is 0.2.

Apart from the proposed DEA-PS-ANN, TEC time series was analyzed with BP-SMNM-ANN, ML-FF-ANN, ABA-SMNM-ANN, PSO-ML-ANN, L&NL-ANN, ANN with the deterministic trend and seasonal components (DT&S-ANN) proposed by Egrioglu et al. in (2015), PSO-PS-ANN, PSO-SMNM-ANN and BP-PS-ANN.

**Table 2** RMSE And MAPE Values Obtained from All Methods for TEC Test Data

Methods	RMSE	MAPE (%)
BP-SMNM-ANN	4,243,944,603	19,54
ML-FF-ANN	1065900000	3,98
ABA-SMNM-ANN	924,149,635	3,73
PSO-ML-ANN	915,190,000	3,39
L&NL-ANN	820,980,000	2,54
DT&S-ANN	810,300,852	2,47
PSO-PS-ANN	697,763,268	2,81
PSO-SMNM-ANN	672,790,000	2,89
BP-PS-ANN	651,383,248	2,22
DEA-PS-ANN	585,195,649	2,31

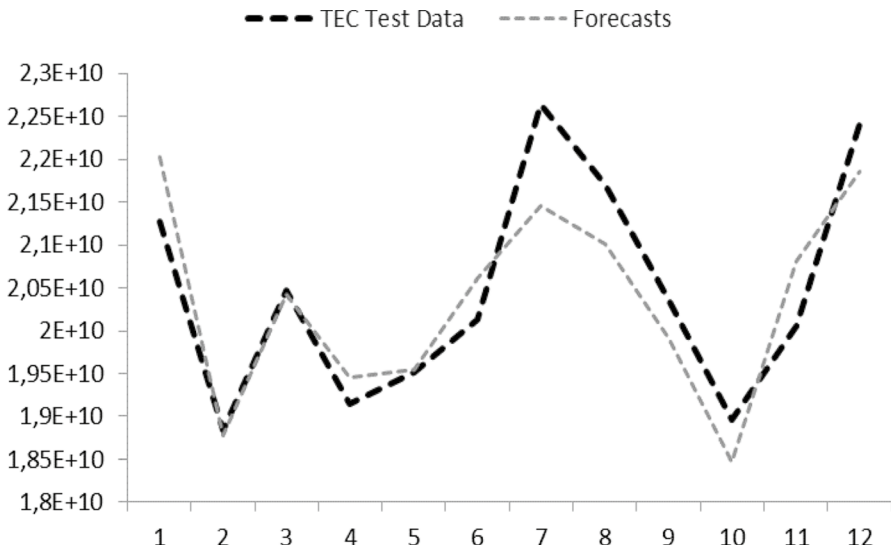


Fig. 6 TEC test set together with the forecasts obtained from the proposed DEA-PS-ANN

When Table 2 is examined, just as in the TEC time series, the proposed DEA-PS-ANN method has the best performance among all methods in term of RMSE criterion and the second-best method in terms of MAPE criterion. Besides, the graph of the actual observations of the TEC test set and the forecasts obtained from the proposed DEA-PS-ANN method for the TEC test set data are given in Fig. 6.

## 5 Conclusions and Discussions

Different from many artificial neural networks in the literature, PS-ANN which is a kind of HONNs has bot additive and multiplicative units in its structure. PS-ANN attracts the attention as an effective model in the literature as an artificial neural network model with high forecasting performance by using these two units together.

In this study, the training of the PS-ANN was performed by DEA. To test the forecasting performance of the proposed DEA-PS-ANN method, analysis of real-life time series was performed using two different data sets. As a result of the analyses, it was concluded that the performance of the proposed method is better than many artificial neural network models available in the literature. In future studies, different mutation operators can be used during the training process of PS-ANN with DEA, or DEA can be used in the training of different neural networks.

## References

- Akdeniz, E., Egrioglu, E., Bas, E., & Yolcu, U. (2018). An ARMA type pi-sigma artificial neural network for nonlinear time series forecasting. *Journal of Artificial Intelligence and Soft Computing Research*, 8(2), 121–132.
- Akram, U., Ghazali, R., Ismail, L. H., Zulqarnain, M., Husaini, N. A., & Mushtaq, M. F. (2019). An improved pi-sigma neural network with error feedback for physical time series prediction. *International Journal of Advanced Trends in Computer Science and Engineering*, 8, 276–284.
- Aladag, C. H., Yolcu, U., & Egrioglu, E. (2013). A new multiplicative seasonal neural network model based on particle swarm optimization. *Neural Processing Letters*, 37(3), 251–262.
- Bas, E. (2016). The training of multiplicative neuron model based artificial neural networks with differential evolution algorithm for forecasting. *Journal of Artificial Intelligence and Soft Computing Research*, 6(1), 5–11.
- Bas, E., Yolcu, U., Egrioglu, E., Cagcag Yolcu, O., & Dalar, A. Z. (2016). Single multiplicative neuron model artificial neuron network trained by bat algorithm for time series forecasting. *American Journal of Intelligent Systems*, 6(3), 74–77.
- Bas, E., Grosan, C., Egrioglu, E., & Yolcu, U. (2018). High order fuzzy time series method based on pi-sigma neural network. *Engineering Applications of Artificial Intelligence*, 72, 350–356.
- Cagcag Yolcu, O., Bas, E., Egrioglu, E., & Yolcu, U. (2018). Single Multiplicative Neuron Model Artificial Neural Network with Autoregressive Coefficient for Time Series Modelling. *Neural Processing Letters*, 47, 1133–1147.
- Dash, R., Routray, A., Rautray, R., & Dash, R. (2018). Gold price prediction using an evolutionary pi-sigma neural network. *International Journal of Engineering & Technology*, 7, 742–746.
- Deepa, M., Rajalakshmi, M., & Nedunchezian, R. (2018). Higher order neural networks based on bio-inspired swarm intelligence optimization algorithm for multimodal tumor data analysis. *Biomedical Research Special Issue*, 29, 113–117.
- Egrioglu, E., Aladag, C.H., Yolcu, U., Bas, E., Dalar, A.Z.: A new neural network model with deterministic trend and seasonality components for time series forecasting. In: *Advances in time series forecasting*, Vol 2, Chapter 4, Bentham, 76–92. (2017)
- Egrioglu, E., Aladag, C. H., Yolcu, U., & Bas, E. (2015). Recurrent multiplicative neuron model artificial neural network for non-linear time series forecasting. *Neural Processing Letters*, 41(2), 249–258.
- Elman, J. L. (1990). Finding Structure in Time. *Cognitive Science*, 14(2), 179–211.
- Ghazali, R., & Jumeily, D. A. (2009). Application of Pi-Sigma neural networks and ridge polynomial neural networks to financial time series prediction. *Artificial Higher Order Neural Networks for Economics and Business*, Chapter, 12, 271–293.
- Giles, C. L., & Maxwell, T. (1987). Learning, invariance, and generalization in high-order neural networks. *Applied Optics*, 26, 4972–4978.
- Giles, C. L., Griffin, R., & Maxwell, T. (1988). *Application of Pi-Sigma neural networks and ridge polynomial neural networks to financial time series prediction* (pp. 301–309). New York: American Institute of Physics.
- Guler, M., Sahin, E.: A binary-input supervised neural unit that forms input dependent higher-order synaptic correlations. *Proceedings of world congress on neural networks III*. pp 730–735. (1994)
- Gundogdu, O., Egrioglu, E., Aladag, C. H., & Yolcu, U. (2015). Multiplicative neuron model artificial neural network based on gauss activation function. *Neural Computing and Applications*, 27(4), 927–935.
- Husaini, N. A., Ghazali, R., Nawi, M. N., & Ismail, L. H. (2011). Pi-Sigma neural network for temperature forecasting in batu pahat. In J. M. Zain, W. M. Wan Mohd, & E. El-Qawasmeh (Eds.), *Software Engineering and Computer Systems. Communications in Computer and Information Science*. Berlin Heidelberg: Springer.
- Husaini, N. A., Ghazali, R., Nawi, M. N., & Ismail, L. H. (2012). The Effect of Network Parameters on Pi-Sigma Neural Network for Temperature Forecasting. *International Journal of Modern Physics: Conference Series*, 9, 440–447.
- Husaini, N. A., Ghazali, R., Nawi, M. N., Ismail, L. H., Deris, M. M., & Herawan, T. (2014). Pi-Sigma neural network for a one-step-ahead temperature forecasting. *International Journal of Computational Intelligence and Applications*, 13(4), 1450023.

- Kelwade, J. P., & Salankar, S. S. (2017). Training of multilayer perceptrons with improved particle swarm optimization for the heart diseases prediction. *International Journal of Swarm Intelligence and Evolutionary Computation*, 6(2), 1–8.
- Kocak, C., Dalar, A. Z., Cagcag Yolcu, O., Bas, E., & Egrioglu, E. (2020). A new fuzzy time series method based on an ARMA-Type recurrent Pi-Sigma artificial neural network. *Soft Computing*, 24, 8243–8252.
- Li, X. B. (2009). RBF neural network optimized by particle swarm optimization for forecasting urban traffic flow. In *Proceedings of the third international symposium on intelligent technology application*, (pp. 124–127).
- McCulloch, W. S., & Pitts, W. A. (1943). A logical calculus of the ideas immanent in nervous activity. *Buttetin of Mathematics and Biophysics*, 5, 115–133.
- Mohamed, K. S., Liu, Y., Wu, W., & Alemu, H. Z. (2016). Batch gradient method for training of Pi-Sigma neural network with penalty. *International Journal of Artificial Intelligence & Applications (IJAA)*, 7(1), 11–20.
- Nayak, S. C. (2017). Development and performance evaluation of adaptive hybrid higher order neural networks for exchange rate prediction. *International Journal of Intelligent Systems and Applications*, 9(8), 71–85.
- Nayak, S. C. (2020). A fireworks algorithm-based Pi-Sigma neural network (FWA-PSNN) for modelling and forecasting chaotic crude oil price time series. *EAI Endorsed Transactions on Energy Web*. <https://doi.org/10.4108/eai.13-7-2018.162803>.
- Nayak, S. C., & Ansari, M. D. (2020). Cooperative optimization algorithm based higher order neural networks for stock forecasting. *Recent Advances in Computer Science and Communications*. <https://doi.org/10.2174/2213275912666190902121153>.
- Nayak, J., Naik, B., & Behera, H. S. (2015). A Novel Chemical Reaction Optimization Based Higher Order Neural Network (CRO-HONN) For nonlinear classification. *Ain Shams Engineering Journal*, 6, 1069–1091.
- Ojha, V. K., Abraham, A., & Snášel, V. (2017). Metaheuristic design of feedforward neural networks: a review of two decades of research. *Engineering Applications of Artificial Intelligence*, 60, 97–116.
- Panda, N., & Majhi, S. K. (2020). Improved spotted hyena optimizer with space transformational search for training Pi-Sigma higher order neural network. *Computational Intelligence*, 36(1), 320–350.
- Pattanayak, R.M., Behera, H.S., Panigrahi, S.: A multi-step-ahead fuzzy time series forecasting by using hybrid chemical reaction optimization with Pi-Sigma higher-order Neural Network. *Computational Intelligence in Pattern Recognition*, 1029–1041. (2020)
- Rumelhart, E., Hinton, G. E., & Williams, R. J. (1986). *Learning Internal Representations by Error Propagation*, Chapter 8 (pp. 318–362). Cambridge: The M.I.T. Press.
- Shin, Y., Gosh, J.: The sigma network: an efficient higher order neural network for pattern classification and function approximation. In *Proceedings of the International Joint Conference on Neural Networks*. (1991)
- Storn, R., & Price, K. (1995). *Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces*, technical report TR-95-012. Berkeley: International Computer Science Institute.
- Yadav, R. N., Kalra, P. K., & John, J. (2007). Time series prediction with single multiplicative neuron model. *Applied Soft Computing*, 7, 1157–1163.
- Yolcu, U., Egrioglu, E., Aladag, C.H. A New Linear & Nonlinear artificial neural network model for time series forecasting. *Decision Support Systems*, 1340–134. (2013)
- Zhao, L., & Yang, Y. (2009). PSO-based single multiplicative neuron model for time series prediction. *Expert Systems with Applications*, 36, 2805–2812.