Turkish Journal of Forecasting vol. 8 no. 2 (2024) pp 8-15

Content list available at JournalPark

Turkish Journal of Forecasting

Journal Homepage: tjforecasting.com

Forecasting of Giresun Hazelnut Quantity in Giresun Province Using Pi-Sigma Artificial Neural Networks

Özlem Karahasan[1,*]

[1]*Avrasya University, Faculty of Engineering and Architecture, Department of Computer Engineering, Pelitli Campus, 61010, Trabzon, Turkey*

Abstract

Artificial neural networks are frequently used to solve many problems and give successful results. Artificial neural networks, which we frequently encounter in solving forecasting problems, attract the attention of researchers with the successful results they provide. Pi-sigma artificial neural network, which is a high-order artificial neural network, draws attention with its use of both additive and multiplicative combining functions in its architectural structure. This artificial neural network model offers successful forecasting results thanks to its high-order structures. In this study, the pi-sigma artificial neural network was preferred due to its superior performance properties, and the particle swarm optimization algorithm was used for training the pi-sigma artificial neural network. To evaluate the performance of this preferred artificial neural network, monthly ready-made manufacturer sale shelled hazelnut quantities in Giresun province was used and a comparison was made with many artificial neural network models available in the literature. It has been observed that this tested method has the best performance among other compared methods.

Keywords: Pi-Sigma Artificial Neural Network, Particle Swarm Optimization Algorithm, Giresun Hazelnut Quantity, Forecasting.

1. Introduction

Artificial neural networks, which are a branch of the machine learning approach, can be examined under two headings: deep and shallow. Pi-sigma artificial neural network (PS), one of the shallow artificial neural networks, is also an example of a high-order artificial neural network because it contains additive and multiplicative combinations in its architecture. PS have higher performance power thanks to these combinations in their structure. Since it uses fewer parameters than deep artificial neural networks, it is frequently preferred in time series problems. Since fewer parameters are used in the training process of the network, the learning process is faster. This is more advantageous compared to many methods in the literature.

Literature information about many artificial neural networks and PS is as follows.

Nie and Deng (2008) used a hybrid genetic learning algorithm to train PS and applied it to resolve a function-optimizing problem. Panigrahi et al. (2013) proposed a modified differential evolution (DE) algorithm trained PS for classification problems. Lalis and Maravillas (2014) proposed a scheme with eight steps for dynamic time series forecasting using adaptive MLP with minimal complexity. Nayak et al. (2014) proposed a standard back propagation gradient descent learning trained higher-order Jordan PS for the classification of real-world data. Nayak et al. (2014) proposed a hybrid PSO-GA-based PS with standard back propagation gradient descent learning (PSO-GA-PSNN) for classification problems. Szoplik (2015) presented the results of forecasting the gas demand obtained with the use of a multilayer perceptron model artificial neural network (MLP). Kanungo et al. (2016) proposed PS with an improved PSO for data classification. Akram et al. (2017) presented awareness about PS for time series forecasting, to highlight some benefits and challenges using PS. Bas et al. (2018) proposed a new model for determining

* Corresponding Author.
E-mail addresses: ozlmkrhsn@hotmail.com (Özlem Karahasan)

ORCID ID:
Özlem Karahasan: 0000-0001-5704-7684

the fuzzy relationships for high-order fuzzy time series forecasting which uses PS. Akdeniz et al. (2018) proposed a new recurrent architecture for PS and used a learning algorithm based on PSO for the training of the proposed neural network. Egrioglu et al. (2019) proposed a forecasting method for a single-variable high-order intuitionistic fuzzy time series forecasting model and did fuzzification of observations by using intuitionistic fuzzy c-means algorithm and defined fuzzy relations by PS. Yan et al. (2019) proposed a hybrid deep learning model combining an ensemble LSTM with the stationary wavelet transform technique aiming at the energy consumption forecasting problem of individual households. Wang et al. (2020) proposed a novel approach based on the LSTM network for forecasting the periodic energy consumption. Bolandnazar et al. (2020) to assess the energy use pattern and select the best method among Cobb-Douglas, multiple linear regression, MLP, radial basis function and support vector machine models to forecast potato output energy in Jiroft city, located in the south of Kerman province, Iran. Nayak (2020) proposed to incarcerate the uncertainties coupled with the crude oil prices, a hybrid forecasting model fireworks algorithm - PS. Swapna Rekha et al. (2020) illustrated an extensive analytical study of a higher-order neural network called PS and its variants in various application domains such as classification, forecasting, function approximation, and pattern recognition. Kocak et al. (2020) proposed a new fuzzy time series algorithm based on an autoregressive integrated moving average-type recurrent PS. Bas et al. (2021) used the sine cosine algorithm for the first time in the training of PS. Yılmaz et al. (2022) performed the training of PS by DE using the DE/rand/1 mutation strategy. Bas et al. (2022) proposed a novel intuitionistic fuzzy time series method to be used in solving forecasting problems. Kumar (2022) proposed a novel higher-order context-layered recurrent PS for the identification of nonlinear dynamical systems. Bas et al. (2022) proposed a new training algorithm based on particle swarm optimization (PSO) for SRNN training. Egrioglu et al. (2023) proposed a nonlinear causality test based on a single multiplicative neuron model artificial neural network which is trained by PSO. Hekimoğlu et al. (2023) considered different machine learning methods for freshwater demand forecasting for Istanbul and compared forecasting accuracies of ARIMA, Holt-Winters, artificial neural networks, recursive neural networks, LSTM, and SRNN models. Amole et al. (2023) used LSTM, SRNN, and GRU to analyse the effect of the COVID-19 pandemic on energy consumption and forecast future energy consumption in various districts in Lagos, Nigeria. Egrioglu and Bas (2023) proposed unlike classical PS, a modified PS by taking the weights and biases as variables between the hidden layer and the output layer of the network. Bas and Eğrioğlu (2023) proposed a new recurrent PS and the architecture of the proposed new recurrent PS used the SES. Shan et al. (2023) devised a deep learning architecture, featuring an Attention-BiLSTM network for short-term water demand forecasting. Cansu et al. (2023) proposed a new PSO-based training algorithm in training LSTM. Egrioglu et al. (2023) proposed a new winsorized dendritic neuron model artificial neural network. Bas et al. (2023) proposed a new robust learning algorithm based on PSO and Huber's loss function for PS. Dash et al. (2023) designed a PS for foretelling the future currency exchange rates in different forecasting horizons. Jhong et al. (2024) proposed a novel long and short-term memory neural network-genetic algorithm (LSTM-GA) model, which integrates LSTM with GA to optimize the LR, NL, and NN for fast food forecasting. Xu et al. (2024) proposed a novel crude oil futures price volatility forecasting framework based on the Bidirectional LSTM -Attention Mechanism Model (Bi-LSTM-Attention) to analyse the impacts of COVID-19 and the Russia-Ukraine conflict on the volatility of crude oil futures price. Zhang et al. (2024) proposed a novel autoregressive integrated moving average (ARIMA)-LSTM hybrid model for WT-ARIMA-LSTM share price index futures forecasting. Kollu et al. (2024) presented a comparative study of some deep learning techniques such as MLP, autoregressive neural network, convolutional neural network (CNN), and LSTM network in forecasting the CPU, and memory usage of many virtual machines. de Moraes Sarmento et al. (2024) proposed a hybrid CNN-LSTM FED, trained using the public Smart* and the building data genome project 2 datasets. The performance of CNN-LSTM FED was evaluated by comparing it against the MLP and CNN-LSTM. Sharma et al. (2024) used artificial neural networks with MLP and extreme learning machines to forecast diesel demand. Karahasan et al. (2024) proposed a hybrid approach SRNN-EXP-S is proposed in which simple recurrent artificial neural network (SRNN) and the simple exponential smoothing (SES) methods, which perform very well in the forecasting problems of time series with seasonal components, are used together. Fan et al. (2024) proposed a new hybrid forecasting model, the EWT-CNN-SRNN-LSTM model to forecast power consumption and compared it with empirical wavelet decomposition, CNN, recurrent neural network, LSTM, and Bayesian optimization algorithm. Kolemen et al. (2024) proposed a hybrid artificial neural network architecture consisting of a combination of a gated recurrent unit artificial neural network and SES methods for the problem of forecasting seasonal time series. Rajasekaran et al. (2024) proposed SRNN-LSTM hybrid models for univariate solar irradiance forecasting, multivariate temperature forecasting and univariate forecasting of wind speed. Bas et al. (2024) put forward the median dendrite artificial neural networks that are not affected by the presence of outliers even when both the input and output contain outliers.

The motivation of this study is to test the performance power of the PS, using a non-derivative-based training algorithm in the training process, using the data set of monthly ready-made manufacturer sale shelled hazelnut quantities in Giresun province (GHQ). Thanks to this training algorithm, complex derivative structures will not be encountered, and the use of complex mathematical operations will be avoided. This method has been compared with many methods available in the literature.

In this study, the performance of the PS, which is a method available in the literature and trained with the PSO, was tested. The test process was carried out using the data set of GHQ. Thanks to the PSO, which is not a derivative-based algorithm, the use of complex mathematical operations is avoided.

In the second part of the article, PS is explained. In the third part, the PSO is introduced. In the fourth section, the method used is explained. The fifth chapter of the study includes the application part, and finally, the sixth chapter contains the discussion and conclusion part.

2. Pi-Sigma Artificial Neural Network

PS proposed by Shin and Ghosh Shin and Ghosh (1991) is a high-order artificial neural network. PS consists of three layers: an input, a hidden and an output layer. The number of hidden layers in the PS is expressed as the degree of the PS. The hidden layer consists of the linear sum of the inputs. While the weights between the input and hidden layers are different from each other, the weights between the hidden layer and the output layers are constant and take the value of one. In the PS, the product of the linear combinations of the hidden layer outputs creates the output of the network. The architecture of a k-order PS with n inputs is shown in Figure 1.

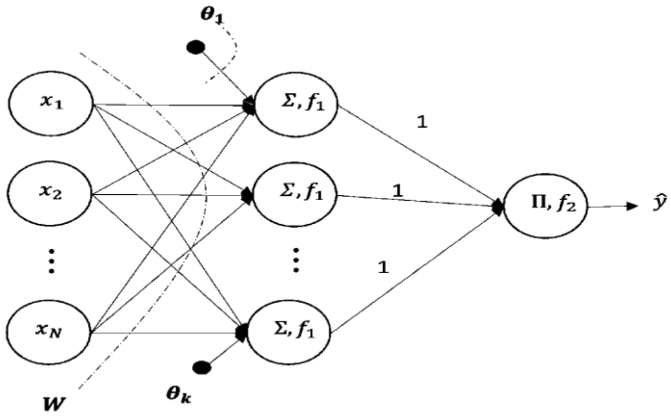

Figure 1. Architecture of PS

With weights w_{ij} $(i = 1,2,\cdots,N\ j = 1,2,\cdots,K)$ and sides θ_j $(j = 1,2,\cdots,K)$ linear combinations of input units are obtained. While w_{ij} $i.$ represents the weight from the input to the j. hidden layer unit, θ_j represents the side value for the jth hidden layer unit. Linear combinations of as many as the number of hidden layer units pass through the linear activation function and create the outputs of the hidden layers. h_j jth represents the output of the hidden layer unit and is calculated with Equation 1.

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

f_1 represents the linear activation function $(f_1(x) = x)$.

The output of the network is calculated with Equation 2.

$$\hat{y} = f_2\left(\prod_{j=1}^{K} h_j\right) = \frac{1}{1+exp(-\prod_{j=1}^{K} h_j)} \tag{2}$$

f_2 represents the logistics activation function $(f_2(x) = \frac{1}{1+exp(-x)})$.

3. Particle Swarm Optimization Algorithm

PSO, which is a population-based heuristic algorithm, was proposed by Kennedy and Eberhart (Kennedy and Eberhart (1995)). It is an optimization algorithm inspired by the social behaviour of fish and bird flocks. PSO is an optimization algorithm that does not need a derivative document. A distinctive feature of the PSO, which offers high solution quality and has very good convergence, is that it simultaneously examines different points in different regions of the solution space to find the global optimum solution. In this way, the local optimum can avoid pitfalls.

4. Training of Pi-Sigma Artificial Neural Network with Particle Swarm Optimization Algorithm

The aim of training artificial neural networks is to produce target values appropriate to the inputs. PSO was used for training the PS. Heuristic optimization algorithms, which are one of the success criteria of the PS, are superior to other derivative-based algorithms because they do not require derivative-based calculations and have an increased probability of not getting caught in local optimum traps. The algorithm of the PS based on the PSO is as follows.

Algorithm

Step 1. The dataset used is divided into three parts: training, validation, and testing. Validation and test dataset lengths are determined by the size of the dataset.

$$n_{train} + n_{validation} + n_{test} = n \tag{3}$$

Here n denotes the total number of observations.

Step 2. The universal set $\left(N \in [N_{low}, N_{up}], K \in [K_{low}, K_{up}]\right)$ is defined for the hyperparameter values of the network. Possible values for the hyperparameter values are determined. Here, N is the number of inputs and K is the number of hidden layer units.

Step 3. The parameters to be used in the PSO are determined. These parameters are as stated below.

c_1^i: the initial cognitive component coefficient.
c_1^f: the final cognitive component coefficient.
c_2^i: the initial social component coefficient.
c_2^f: the final social component coefficient.
w^i: the initial inertia weight.
w^f: the final inertia weight.
$vmaps$: the limit value for velocities.
$maksitr$: the maximum number of iterations.
pn: the number of particles.
$limit1$: the limit value for restart strategy.
$limit2$: limit value for early stop rule.

Step 4. $N = N_{low}$, and $K = K_{low}$.
Step 5. For the PSO, the initial population and velocity values are randomly generated. In a PS, positions are expressed as $p_{i,k}, i = 1,2,\cdots, pn, k = 1,2,\cdots,(N \times K + K)$. All velocity values $(v_{i,k}, i = 1,2,\cdots, pn, k = 1,2,\cdots,(N \times K + K)$ are generated from a uniform distribution with parameters 0 and 1. All speed values are produced in the $[-vmaps, vmaps]$ range.

Table 1. Positions of a particle

Positions	1	$\cdots$	$N \times K$	$N \times K + 1$	$\cdots$	$N \times K + K$
Weight and Bias	W_{11}	$\cdots$	W_{NK}	θ_1	$\cdots$	θ_K

Step 6. Fitness function values for each particle are calculated using the mean square error (MSE) value. G_{best} and P_{best} matrices are created for the population. While G_{best} refers to the best particle in the population, the best position in the population refers to the P_{best} matrix.

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2 \tag{4}$$

Step 7. New position and speed values are calculated with Equations (5-7). While itr represents the current number of iterations, r_1 and r_2 are randomly generated in the range [0,1].

$$v_{i,k}^{(itr)} = w^{(itr)} v_{i,k}^{(itr-1)} + c_1^{(itr)} r_1 \left({P_{best}}_{i,k}^{(itr)} - P_{i,k}^{(itr)} \right) + c_2^{(itr)} r_2 \left({G_{best}}_k^{(itr)} - P_{i,k}^{(itr)} \right) \tag{5}$$

$$v_{i,k}^{itr} = min(vmaps, max(ivmaps, v_{i,k}^{(itr)}) \tag{6}$$

$$P_{i,k}^{(itr)} = P_{i,k}^{(itr-1)} + v_{i,k}^{(itr)} \tag{7}$$

Inertia weight and social and cognitive coefficients are calculated with the help of Equations (8-10).

$$w^{(itr)} = (w_1^i - w_2^i)\frac{maxitr - \mathrm{itr}}{maxitr} + w_2^i \tag{8}$$

$$c_1^{(itr)} = (c_1^f - c_1^i)\frac{itr}{maxitr} + c_1^i \quad (9)$$

$$c_2^{(itr)} = (c_2^i - c_2^f)\frac{maxitr-itr}{maxitr} + c_2^f \quad (10)$$

Step 8. The P_{best} and G_{best} matrices are updated.

Step 9. The restart strategy counter $(rsc = rsc + 1)$is incremented. If $rsc > limit1$, all speed and position values are reproduced, and the counter is reset.

Step 10. The early stopping rule is checked with the help of the equation. While esc is the failure counter, $MSE_{best}^{(t)}$represents the fitness value of MSE in the $t.$ iteration.

$$esc = \begin{cases} esc + 1\,, \dfrac{MSE_{best}^{(t)} - MSE_{best}^{(t-1)}}{MSE_{best}^{(t)}} < 10^{-3} \\ \quad 0,\ \ otherwise \end{cases} \quad (11)$$

$esc > Limit2$, the algorithm is stopped, otherwise go back to Step 7.

Step 11. The RMSE expressed in Equation (12) is calculated for the validation set.

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=n_{train}+1}^{n_{train}+n_{validation}}(y_t - y_t)^2} \quad (12)$$

Step 12. $N = N + 1$ and $K = K + 1$. If $N \leq N_{up}$, $K \leq K_{up}$ go back to Step 5. Otherwise, go to Step 13.

Step 13. The best hyperparameter values are determined according to the validation set performance.

Step 14. By combining training and validation at the best hyperparameter values obtained, the network is trained 30 times on random starts with a larger training set and the test set performance of the network is calculated.

5. Application

In the study, the data set of GHQ between 2010 and 2022 was used. The data set was taken from the https://www.giresuntb.org.tr/GtbVerileri website. The time series was solved using the PS and the performance of the method was compared with MLP (McClelland and Rumelhart (1986)), DNM (Todo et al. (2014)), LSTM (Hochreiter and Schmidhuber (1997)), SRNN methods. In the methods, the number of hidden layers in deep and non-deep neural networks is changed between 1 and s, while in deep neural networks, the number of hidden layers varies between 1 and 2. The data set used is divided into three parts: training, validation, and testing. Validation and test set lengths for all methods were changed to 12. The best hyperparameter values for all methods were trained 30 times using random starting weights. The mean, standard deviation, inter-quarter range, and minimum and maximum statistics of the RMSE values obtained for the test set were calculated. The most possible value of the RMSE value is given by the mean statistic, the variation of repeated solutions is given by the standard deviation statistic, the best case is the minimum statistic, and the worst case is the maximum statistic. The graph of the data set is presented in Figure 2, the analysis results are presented in Table 2.

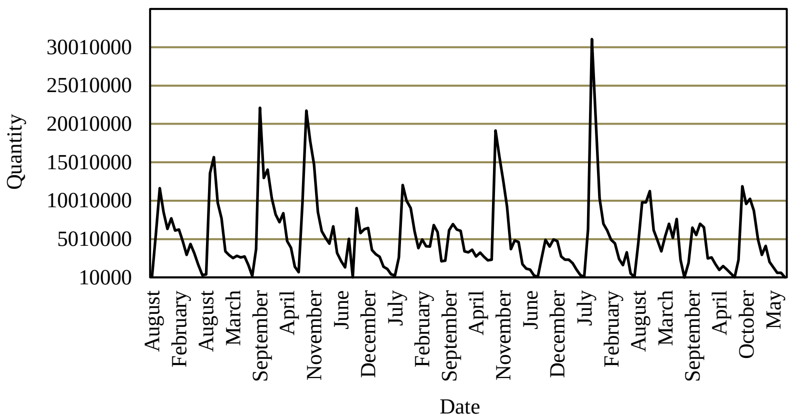

Figure 2. The graph of the time series

As seen in the graph, this series, which includes seasonality, was analysed by taking seasonal differences and comparing them with many existing methods in the literature.

Table 2. Analysis results and hyperparameter values obtained for the time series

Method	Mean	Median	Standard Deviation	Inter-Quarter Range	Minimum	Maximum	N	m	K
MLP	2,4027	2,3999	0,0190	0,0236	2,3632	2,4487	4	-	1
PS	**2,2292**	**2,1827**	0,1282	0,1994	**2,0465**	2,4884	5	-	2
DNM	2,3874	2,3860	0,0133	0,0112	2,3629	2,4229	4	-	1
LSTM	2,3885	2,3884	0,0101	0,0162	2,3696	**2,4069**	3	1	1
SRNN	2,3836	2,3834	**0,0092**	**0,0085**	2,3631	2,4087	5	1	2

When the statistics obtained according to the RMSE values given in Table 3 are examined, it is observed that PS is the best in the mean, median and minimum statistics in the GHQ dataset, while it ranks last in the standard deviation, inter-quarter range and maximum statistics. As a result of the analysis, the best situation for the average, median and minimum statistics was realized when the number of inputs was 5 and the number of hidden layers was 2.

6. Conclusion and Discussion

Artificial neural networks provide successful results in solving forecasting problems. PS, an artificial neural network, offers successful forecasting results thanks to its high-order structures. In this study, the GHQ data set was analysed with a PS with superior performance features, and the PSO was used for training the network. This preferred neural network has been compared with different artificial neural network models. It has been observed that this tested method has the best performance among other compared methods in mean, median and minimum statistics. The performance of this method used in future studies can be analysed with different time series datasets, or the performance of different artificial neural networks can be tested with this data set used in the study.

Funding

No funding was received for this work.

Credit authorship contribution statement

Özlem Karahasan: Conceptualization, Methodology, Software, Data curation, Writing, Original draft preparation, Visualization, Investigation, Supervision, Validation, Reviewing, Editing.

Declaration of competing interest

The author declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

- The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

[1] J. L. McClelland, D. E. Rumelhart, PDP Research Group, Parallel distributed processing, 2, 1986, 20-21, Cambridge, MA: MIT press.
[2] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation, 9(8), 1997, 1735-1780.
[3] Y. Shin, J. Ghosh, The pi-sigma network: An efficient higherorder neural network for pattern classification and function approximation, In IJCNN91-Seattle international joint conference on neural networks, 1, Jul. 1991, 13-18, IEEE.
[4] J. Kennedy, R. Eberhart, Particle swarm optimization, In Proceedings of IEEE International Conference on Neural Networks, 4, 1995, 1942-1948.
[5] Y. Nie, W. Deng, A hybrid genetic learning algorithm for Pi-sigma neural network and the analysis of its convergence, In 2008 fourth international conference on natural computation, Vol. 3, pp. 19-23, Oct. 2008, IEEE.
[6] S. Panigrahi, A. K. Bhoi, Y. Karali, A modified differential evolution algorithm trained pi-sigma neural network for pattern classification, International Journal of Soft Computing and Engineering, 3(5), (2013), 133-136.
[7] J. T. Lalis, E. Maravillas, Dynamic forecasting of electric load consumption using adaptive multilayer perceptron (AMLP), In 2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), pp. 1-7, Nov. 2014, IEEE.

[8] J. Nayak, B. Naik, H. S. Behera, A hybrid PSO-GA based Pi sigma neural network (PSNN) with standard back propagation gradient descent learning for classification, In 2014 international conference on control, instrumentation, communication and computational technologies (iccicct), pp. 878-885, Jul. 2014, IEEE.

[9] J. Nayak, D. P. Kanungo, B. Naik, H. S. Behera, A higher order evolutionary Jordan Pi-Sigma neural network with gradient descent learning for classification, In 2014 International Conference on High Performance Computing and Applications (ICHPCA), pp. 1-6, Dec. 2014, IEEE.

[10] Y. Todo, H. Tamura, K. Yamashita, Z. Tang, Unsupervised learnable neuron model with nonlinear interaction on dendrites, Neural Networks, 60, 2014, 96-103.

[11] J. Szoplik, Forecasting of natural gas consumption with artificial neural networks, Energy, 85, (2015), 208-220.

[12] D. P. Kanungo, J. Nayak, B. Naik, H. S. Behera, Non-linear classification using higher order pi-sigma neural network and improved particle swarm optimization: an experimental analysis, In Computational Intelligence in Data Mining—Volume 2: Proceedings of the International Conference on CIDM, 5-6 Dec. 2015, pp. 507-518, 2016, Springer India.

[13] U. Akram, R. Ghazali, M. F. Mushtaq, A comprehensive survey on Pi-Sigma neural network for time series prediction, Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 9(3-3), (2017), 57-62.

[14] E. Bas, C. Grosan, E. Egrioglu, U. Yolcu, High order fuzzy time series method based on pi-sigma neural network, Engineering Applications of Artificial Intelligence, 72, (2018), 350-356.

[15] E. Akdeniz, E. Egrioglu, E. Bas, U. Yolcu, An ARMA type pi-sigma artificial neural network for nonlinear time series forecasting, Journal of Artificial Intelligence and Soft Computing Research, 8(2), (2018), 121-132.

[16] E. Egrioglu, U. Yolcu, E. Bas, Intuitionistic high-order fuzzy time series forecasting method based on pi-sigma artificial neural networks trained by artificial bee colony, Granular Computing, 4, (2019), 639-654.

[17] K. Yan, W. Li, Z. Ji, M. Qi, Y. Du, A hybrid LSTM neural network for energy consumption forecasting of individual households, Ieee Access, 7, (2019), 157633-157642.

[18] J. Q. Wang, Y. Du, J. Wang, LSTM based long-term energy consumption prediction with periodicity, Energy, 197, (2020), 117197.

[19] E. Bolandnazar, A. Rohani, M. Taki, Energy consumption forecasting in agriculture by artificial intelligence and mathematical models, Energy Sources, Part A: Recovery, Utilization, and Environmental Effects, 42(13), (2020), 1618-1632.

[20] S. C. Nayak, 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, 7(28), (2020), e2-e2.

[21] H. Swapna Rekha, J. Nayak, H. S. Behera, Pi-sigma neural network: Survey of a decade progress, In Computational Intelligence in Pattern Recognition: Proceedings of CIPR 2020, pp. 429-441, 2020, Springer Singapore.

[22] C. Kocak, A. Z. Dalar, O. Cagcag Yolcu, E. Bas, E. Egrioglu, A new fuzzy time series method based on an ARMA-type recurrent Pi-Sigma artificial neural network, Soft Computing, 24, (2020), 8243-8252.

[23] E. Bas, E. Egrioglu, O. Karahasan, A Pi-Sigma artificial neural network based on sine cosine optimization algorithm, Granular Computing, (2021), 1-8.

[24] O. Yılmaz, E. Bas, E. Egrioglu, The training of Pi-Sigma artificial neural networks with differential evolution algorithm for forecasting, Computational Economics, 59(4), (2022), 1699-1711.

[25] E. Bas, E. Egrioglu, E. Kolemen, A novel intuitionistic fuzzy time series method based on bootstrapped combined pi-sigma artificial neural network, Engineering Applications of Artificial Intelligence, 114, (2022), 105030.

[26] R. Kumar, A Lyapunov-stability-based context-layered recurrent pi-sigma neural network for the identification of nonlinear systems, Applied Soft Computing, 122, (2022), 108836.

[27] E. Bas, E. Egrioglu, E. Kolemen, Training simple recurrent deep artificial neural network for forecasting using particle swarm optimization, Granular Computing, 7(2), (2022), 411–420.

[28] E. Egrioglu, E. Bas, T. Cansu, M. A. Kara, A new nonlinear causality test based on single multiplicative neuron model artificial neural network: a case study for Turkey's macroeconomic indicators, Granular Computing, 8(2), (2023), 391-396.

[29] M. Hekimoğlu, A. İ. ÇETİN, B. E. Kaya, Evaluation of Various Machine Learning Methods to Predict Istanbul's Freshwater Consumption, International Journal of Environment and Geoinformatics, 10(2), (2023), 1-11.

[30] A. O. Amole, S. Oladipo, D. Ighravwe, K. A. Makinde, J. Ajibola, Comparative analysis of deep learning techniques based COVID-19 impact assessment on electricity consumption in distribution network, Nigerian Journal of Technological Development, 20(3), (2023), 23-46.

[31] E. Egrioglu, E. Bas, Modified pi sigma artificial neural networks for forecasting, Granular Computing, 8(1), (2023), 131-135.

[32] E. Bas, E. Eğrioğlu, A new recurrent pi-sigma artificial neural network inspired by exponential smoothing feedback mechanism, Journal of Forecasting, 42(4), (2023), 802-812.

[33] S. Shan, H. Ni, G. Chen, X. Lin, J. Li, A Machine Learning Framework for Enhancing Short-Term Water Demand Forecasting Using Attention-BiLSTM Networks Integrated with XGBoost Residual Correction, Water, 15(20), (2023), 3605.

[34] T. Cansu, E. Kolemen, Ö. Karahasan, E. Bas, E. Egrioglu, A new training algorithm for long short-term memory artificial neural network based on particle swarm optimization, Granular Computing, (2023), 1-14.

[35] E. Egrioglu, E. Bas, O. Karahasan, Winsorized dendritic neuron model artificial neural network and a robust training algorithm with Tukey's biweight loss function based on particle swarm optimization, Granular Computing, 8(3), (2023), 491-501.

[36] E. Bas, E. Egrioglu, U. Yolcu, M. Y. Chen, A Robust Learning Algorithm Based on Particle Swarm Optimization for Pi-Sigma Artificial Neural Networks, Big Data, 11(2), (2023), 105-116.

[37] R. Dash, R. Rautray, R. Dash, Utility of a Shuffled Differential Evolution algorithm in designing of a Pi-Sigma Neural Network based predictor model, Applied Computing and Informatics, 19(1/2), (2023), 22-40.

[38] Y. D. Jhong, C. S. Chen, B. C. Jhong, C. H. Tsai, S. Y. Yang, Optimization of LSTM parameters for flash flood forecasting using genetic algorithm, Water Resources Management, (2024), 1-24.

[39] Y. Xu, T. Liu, P. Du, Volatility forecasting of crude oil futures based on Bi-LSTM-Attention model: The dynamic role of the COVID-19 pandemic and the Russian-Ukrainian conflict, Resources Policy, 88, (2024), 104319.

[40] J. Zhang, H. Liu, W. Bai, X. Li, A hybrid approach of wavelet transform, ARIMA and LSTM model for the share price index futures forecasting, The North American Journal of Economics and Finance, 69, (2024), 102022.

[41] P. K. Kollu, T. S. Janjanam, K. S. Siram, Comparative analysis of cloud resources forecasting using deep learning techniques based on VM workload traces, Transactions on Emerging Telecommunications Technologies, 35(1), (2024), e4933.

[42] E. M. de Moraes Sarmento, I. F. Ribeiro, P. R. N. Marciano, Y. G. Neris, H. R. de Oliveira Rocha, V. F. S., Mota, R. da Silva Villaça, Forecasting energy power consumption using federated learning in edge computing devices, Internet of Things, 25, (2024), 101050.

[43] A. M. Sharma, S. Baby, V. Raghu, Forecasting High Speed Diesel Demand in India with Econometric and Machine Learning Methods, International Journal of Energy Economics and Policy, 14(1), (2024), 496.

[44] O. Karahasan, E. Bas, E. Egrioglu, New deep recurrent hybrid artificial neural network for forecasting seasonal time series, Granular Computing, 9(1), (2024), 19.

[45] G. F. Fan, Y. Y. Han, J. W. Li, L. L. Peng, Y. H. Yeh, W. C. Hong, A hybrid model for deep learning short-term power load forecasting based on feature extraction statistics techniques, Expert Systems with Applications, 238, (2024), 122012.

[46] E. Kolemen, E. Egrioglu, E. Bas, M. Turkmen, A new deep recurrent hybrid artificial neural network of gated recurrent units and simple seasonal exponential smoothing, Granular Computing, 9(1), (2024), 7.

[47] U. Rajasekaran, G. K. Sriram, A. Malini, V. Sharma, Hybrid Explainable SRNN-LSTM Architecture for Irradiance, Temperature and Wind Speed Forecasting, (2023).

[48] E. Bas, E. Egrioglu, T. Cansu, Robust training of median dendritic artificial neural networks for time series forecasting, Expert Systems with Applications, 238, (2024), 122080.