

Single Multiplicative Neuron Model Artificial Neural Network with Autoregressive Coefficient for Time Series Modelling

Ozge Cagcag Yolcu¹ · Eren Bas² · Erol Egrioglu² ·
Ufuk Yolcu³

Published online: 16 August 2017
© Springer Science+Business Media, LLC 2017

Abstract Single multiplicative neuron model and multilayer perceptron have been commonly used for time series prediction problem. Having a simple structure and features of easily applicable differentiates the single multiplicative neuron model from the multilayer perception. While, multilayer perceptron just as many other artificial neural networks are data-based methods, single multiplicative neuron model has a model structure due to it is composed of a single neuron. Multilayer perceptron can highly compliance with data by changing its architecture, though single multiplicative neuron model, in this respect, is insufficient. In this study, to overcome this problem of single multiplicative neuron model, a new model that its weights and biases are obtained by way of autoregressive equations is proposed. Since the time indexes are considered to determine weights and biases from the autoregressive models, the proposed neural network can be evaluated as a data-based model. To show the performance and capability of the proposed method, various implementations have been executed over some well-known data sets. And the obtained results demonstrate that data-based proposed method has outstanding forecasting performance.

Keywords Single multiplicative neuron model · Autoregressive coefficient · Data-based model · Time series forecasting

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s11063-017-9686-3>) contains supplementary material, which is available to authorized users.

✉ Ozge Cagcag Yolcu
ozgecagcag@yahoo.com

¹ Department of Industrial Engineering, Forecast Research Laboratory, Giresun University, 28200 Giresun, Turkey

² Department of Statistics, Forecast Research Laboratory, Giresun University, 28200 Giresun, Turkey

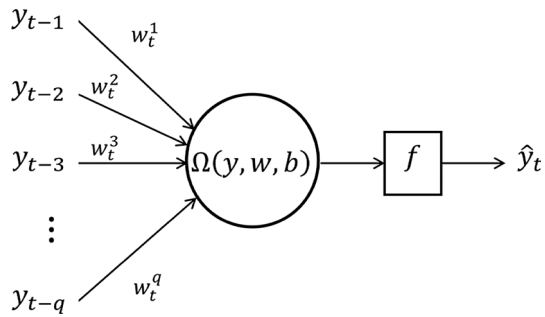
³ Department of Econometrics, Forecast Research Laboratory, Giresun University, 28200 Giresun, Turkey

1 Introduction

As a data mining field, analysis of time series has been one of the main research subjects for decades. For modelling the time series, such as Auto-Regressive (AR) model, Linear Dynamical Systems (LDS) and Hidden Markov Model (HMM) and traditional approaches that include the estimation of parameters from an assumed time series model may fail to analysis of complex real-world time series. To get rid of the restrictions of traditional approaches as model assumption, normal distribution and the number of observation, in recent years, artificial neural networks (ANNs) have been commonly and successfully utilized. While a variety of neuron model were put forward by [6,26,30,32,44], one of the most common method used for time series prediction is multilayer perceptron (MLP) proposed by McCulloch and Pitts [36]. While Zhang et al. [46] and Zhang [45] reviewed the time series literature in which uses MLP, Crone and Kourentzes [10] and Crone et al. [11], in their studies, evaluated the forecasting performance of ANN in time series analysis. In time series literature, while dynamic ANN models [3,20] have been used, some hybrid methods [1,5,12,18,19,35,39,42] have a huge usage area which should not be underestimated. Moreover, some comparative [7,9,13,21,22,31,33] and application [4,14,17] studies were presented. To predict various time series, some new ANN models have been put forward by Voyant et al. [40], Laboissiere et al. [27], Cheng et al. [8], Kim [24] and Wang et al. [41]. Furthermore, Reyes et al. [34] presented a new earthquake prediction system based on the application of ANN to predict earthquakes in Chile. Koprinska et al. [25] introduced a new approach based on sequence similarity with neural networks for forecasting of electricity demand time series. Martínez-Álvarez et al. [29] explored the application of various data mining techniques to time series forecasting.

MLP has more than one neuron in the hidden layer and the output is a non-linear combination of multiplication of weighted sum of the inputs. When it comes to forecast time series by using MLP, the determination of architecture in a proper way seems as an essential problem because, in particular, the number of neurons in hidden layer directly affects the performance of multilayer perceptron neural networks. To determine the number of neurons in hidden layer and inputs of the model, Egrioglu et al. [15] put forward a method. Multiplicative neuron model (MNM) that does not contain this type of problem is introduced by Yadav et al. [43]. MNM named single multiplicative neuron model (S-MNM) has just one neuron. S-MNM uses a multiplicative function in its neuron as an aggregation function on the contrary to MLP that uses additive function. This multiplicative structure strengthens non-linearity characteristic of the model. S-MNM uses less parameter than those employed by MLP since it has only one neuron in the hidden layer [2]. Although S-MNM has some advantages in comparison to MLP, it has some difficulty in certain time series prediction since it is model-based the reason is that it has just a single neuron. To analysis of time series that may contain more complex structures, while MLP has outstanding prediction performance under favour of its high compliance with data by changing its architecture, S-MNM, from this aspect, is insufficient. When we take into consideration both advantages and disadvantages of MLP and S-MNM together, generating a model that has outstanding sides of both MLP and S-MNM conduce toward to obtain better prediction performance. From this point of view, in this study, we proposed a single multiplicative neuron model with autoregressive coefficient (AC-S-MNM). In AC-S-MNM, weights and biases are achieved by way of autoregressive equations. In the obtaining process of weights and biases, since the weights and the biases are determined via autoregressive equations that consider the time index of each observation, the model is a data-based. The parameters of autoregressive equations that generate the weights

Fig. 1 The architecture of AC-S-MNM



and biases of AC-S-MNM are specified by utilizing the modified particle swarm optimization (MPSO). The performance of proposed model is displayed through several implementations and compares with some others models which have been commonly used in the time series prediction literature.

In the rest of the paper, Sect. 2 comprises the proposed model in detail and in this section an algorithm is given regarding to operation of the proposed model. Various implementations and their results are presented in Sect. 3. Finally, in the last section; a discussion and an overall assessment are presented over the obtained results.

2 Single Multiplicative Neuron Model with Autoregressive Coefficient

S-MNM and MLP have been widely used to predict time series. While MLP has some advantages such as highly compliance with data, difficulties of architecture specified must be considered the down side of this ANN model. Moreover, whereas the S-MNM does not have such a problem, in prediction problems of some complex time series, it can remain incapable by the reason of containing a strict model-based structure.

AC-S-MNM, proposed in this study, contain in itself the master characteristics of both MLP and S-MNM such as a highly compatibility with data which is a crucial feature in prediction of complex time series and not including architecture selection problem under favour of having just a single neuron. Since the proposed AC-S-MNM model is basically an S-MNM, it has same structure with S-MNM and it can be demonstrated in Fig. 1.

In Fig. 1, function $\Omega(y, w, b)$ is the product of the weighted inputs and f is activation function. Here, $y_{t-1}, y_{t-2}, \dots, y_{t-q}$ and \hat{y}_t are inputs and output of AC-S-MNM, respectively. Moreover, q is called as model order, n is the number of observation in training set of time series. The S-MNM with q inputs given in Fig. 1 has $q \times 2$ weights. Of these, q are the weights corresponding to the inputs ($w_t^i, i = 1, 2, \dots, q; t = 1, 2, \dots, n$) and q to the sides of the weights ($b_t^i, i = 1, 2, \dots, q; t = 1, 2, \dots, n$). The output producing process of AC-S-MNM can be given as an algorithm.

Algorithm 1 The calculation of outputs of AC-S-MNM

Step 1 Autoregressive equations are constituted for weights and biases.

For each of weights and biases, autoregressive models are given in Eqs. (1) and (2), respectively.

$$w_t^i = \varnothing_0^i + \varnothing_1^i w_{t-1}^i + \varepsilon_t^i, i = 1, 2, \dots, q; \quad t = 1, 2, \dots, n \tag{1}$$

$$b_t^i = \theta_0^i + \theta_1^i b_{t-1}^i + \varepsilon_t^i, i = 1, 2, \dots, q; \quad t = 1, 2, \dots, n \tag{2}$$

Totally, there are $2q$ autoregressive models. The initial values of each model are w_0^i and $b_0^i, i = 1, 2, \dots, q$. The parameters of these $2q$ autoregressive models are estimated by using MPSO in an optimization process (see Algorithm 2).

Step 2 Weights and biases are calculated.

For each of learning sample, weights and biases $(w_1^i, w_2^i, \dots, w_n^i; b_1^i, b_2^i, \dots, b_n^i, i = 1, 2, \dots, q)$ are calculated by using autoregressive equations given in Eqs. (1) and (2). For example, Let $q = 2, n = 2$ and the coefficients of autoregressive models are estimated by using MPSO as $\vartheta_0^1 = 0.1, \vartheta_0^2 = 0.2, \vartheta_1^1 = 0.3, \vartheta_1^2 = 0.4; \theta_0^1 = 0.5, \theta_0^2 = 0.6, \theta_1^1 = 0.7, \theta_1^2 = 0.8; w_0^1 = 0.1, w_0^2 = 0.2, b_0^1 = 0.3$ and $b_0^2 = 0.4$ weights and biases are calculated as below:

$$\text{For } t = 1 \text{ and } i = 1; w_1^1 = \vartheta_0^1 + \vartheta_1^1 w_0^1 + w \varepsilon_1^1 = 0.1 + (0.3 \times 0.1) + 0 = 0.13$$

$$b_1^1 = \theta_0^1 + \theta_1^1 b_0^1 + b \varepsilon_1^1 = 0.5 + (0.7 \times 0.3) + 0 = 0.71$$

$$\text{For } t = 1 \text{ and } i = 2; w_1^2 = \vartheta_0^2 + \vartheta_1^2 w_0^2 + w \varepsilon_1^2 = 0.2 + (0.4 \times 0.2) + 0 = 0.28$$

$$b_1^2 = \theta_0^2 + \theta_1^2 b_0^2 + b \varepsilon_1^2 = 0.6 + (0.8 \times 0.4) + 0 = 0.92$$

$$\text{For } t = 2 \text{ and } i = 1; w_2^1 = \vartheta_0^1 + \vartheta_1^1 w_1^1 + w \varepsilon_2^1 = 0.1 + (0.3 \times 0.13) + 0 = 0.139$$

$$b_2^1 = \theta_0^1 + \theta_1^1 b_1^1 + b \varepsilon_2^1 = 0.5 + (0.7 \times 0.71) + 0 = 0.997$$

$$\text{For } t = 2 \text{ and } i = 2; w_2^2 = \vartheta_0^2 + \vartheta_1^2 w_1^2 + w \varepsilon_2^2 = 0.2 + (0.4 \times 0.28) + 0 = 0.312$$

$$b_2^2 = \theta_0^2 + \theta_1^2 b_1^2 + b \varepsilon_2^2 = 0.6 + (0.8 \times 0.92) + 0 = 1.336$$

In autoregressive models, $b \varepsilon_t^i$ and $w \varepsilon_t^i, i = 1, 2, \dots, q; t = 1, 2, \dots, n$ error terms are always taken as zero.

Step 3 Values of *net* are calculated.

$$net_t = \prod_{i=1}^q (w_t^i \times y_{t-i} + b_t^i); \quad t = 1, 2, \dots, n \tag{3}$$

Step 4 The outputs of AC-S-MNM are obtained.

For each of learning sample, the values of output are calculated by passing through logistic activation function.

$$\hat{y}_t = f(net_t) = \frac{1}{1 + e^{-net_t}}; \quad t = 1, 2, \dots, n \tag{4}$$

While the proposed AC-S-MNM is used to obtain the outputs for each learning sample, it uses weights and biases that are obtained by making use of autoregressive equations. In this stage, it is necessary that the coefficients (parameters) of autoregressive equations are determined to obtain the weights and biases. In our proposed approach, the coefficients of autoregressive equations are obtained by taking advantage of MPSO in an optimization process.

Particle swarm optimization (PSO) is an evolutionary computation technique proposed by Kennedy and Eberhart [23]. PSO can be evaluated as a population based optimization tool. The particle swarm concept originated as a simulation of simplified social system [38]. Distinguishing feature of this heuristic algorithm is that it simultaneously examines different points in different regions of the solution space to obtain the global optimum solution. Local optimum traps can be avoided because of this feature of the method [2]. The MPSO algorithm has time varying inertia weight like in [37]. In a similar way, this algorithm also has time varying acceleration coefficient like in [28].

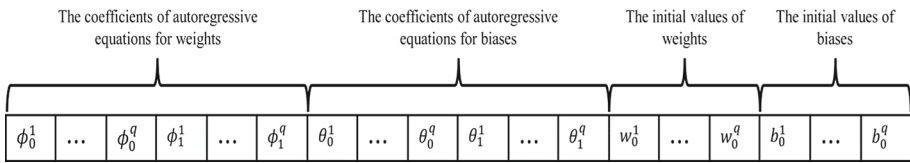


Fig. 2 The structure of a particle in a swarm

In the optimization process, this process can also be called as training of AC-S-MNM, the coefficients aimed to get are $\phi_0^i, \theta_0^i, \phi_1^i, \theta_1^i, w_0^i$ and $b_0^i; i = 1, 2, \dots, q$. Thus, each particle of swarm has $6 \times q$ positions. The structure of a particle in a swarm is illustrated in Fig. 2.

To get the coefficients of autoregressive equations, the optimization process can be given as an algorithm.

Algorithm 2 *The training of AC-S-MNM by taking advantage of MPSO*

Step 1 The parameters of MPSO are determined.

In the first step, the parameters which direct the MPSO algorithm are determined. These parameters are;

- pn : The number of particles in the swarm.
- vm : Velocities size.
- (c_{1i}, c_{1f}) : The intervals which includes possible values of cognitive coefficient c_1 .
- (c_{2i}, c_{2f}) : The intervals which includes possible values of social coefficient c_2 .
- (w_1, w_2) : The intervals which includes possible values of inertia parameter w .
- itr_{max} : The number of maximum iteration.

Step 2 Initial values of positions and velocities are generated.

Initial positions of each k th ($k = 1, 2, \dots, pn$) particle are randomly generated from uniform distribution (0, 1) and kept in a vector of k^X given as follows:

$${}_kX = \{{}_kx_1, {}_kx_2, \dots, {}_kx_{6q}, \}, \quad k = 1, 2, \dots, pn \tag{5}$$

where ${}_kx_l$ ($l = 1, 2, \dots, 6q$) represents l th position of k th particle. Moreover initial values of velocities are randomly generated from uniform distribution $(-vm, vm)$, and kept in a vector of ${}_kV$ given bellow.

$${}_kV = \{{}_kv_1, {}kv_2, \dots, {}kv_{6q}, \}, k = 1, 2, \dots, pn \tag{6}$$

where ${}_kv_l$ ($l = 1, 2, \dots, 6q$) represents velocity for l th position of k th particle.

Step 3 Fitness function values are computed.

Mean square error (MSE) is taken as fitness function in this step. For k_{th} particle, MSE value are obtained in Eq. (7)

$${}_kMSE = \frac{1}{n} \sum_{t=1}^n (y_t - {}_k\hat{y}_t)^2, \quad k = 1, 2, \dots, pn \tag{7}$$

The outputs, ${}_k\hat{y}_t$, are calculated by using algorithm 1 and the positions of corresponding particle k . Weights and biases are designed like in Fig. 1 from positions of corresponding particle k .

Step 4 ${}_kPbest$ and $Gbest$ are determined.

${}_kPbest = ({}_kp_1, {}kp_2, \dots, {}kp_{6q}; k = 1, 2, \dots, pn)$ and $Gbest = (p_1, p_2, \dots, p_{6q})$ are specified via fitness function values calculated in Step 2. ${}_kPbest$ and $Gbest$ represent a vector stores the positions corresponding to the k_{th} particle’s best individual performance,

and the positions of the best particle which has the best fitness function value, obtained so far, respectively.

Step 5 The parameters of MPSO are updated.

The updated values of cognitive coefficient c_1 , social coefficient c_2 and inertia parameter w are calculated using the formulas given in (8), (9), and (10).

$${}^r c_1 = (c_{1f} - c_{1i}) \frac{r}{itr_{max}} + c_{1i} \tag{8}$$

$${}^r c_2 = (c_{2f} - c_{2i}) \frac{r}{itr_{max}} + c_{2i} \tag{9}$$

$${}^r w = (w_2 - w_1) \frac{itr_{max} - r}{itr_{max}} + w_1 \tag{10}$$

where, r is current iteration number. Moreover, indexes of f and i represent final, and initial of possible values for cognitive and social coefficients, respectively.

Step 6 Values of velocities and positions are updated.

Velocities and positions are re-obtained by using the equations given in (11) and (12), respectively.

$${}^{r+1}{}_k v_l = [{}^r w \times {}^r{}_k v_l + {}^r c_1 \times rand_1 \times ({}_k p_l - {}_k x_l,) + {}^r c_2 \times rand_2 \times (p_l - {}_k x_l,)] \tag{11}$$

$${}^{r+1}{}_k x_l = {}^r{}_k x_l + {}^{r+1}{}_k v_l \tag{12}$$

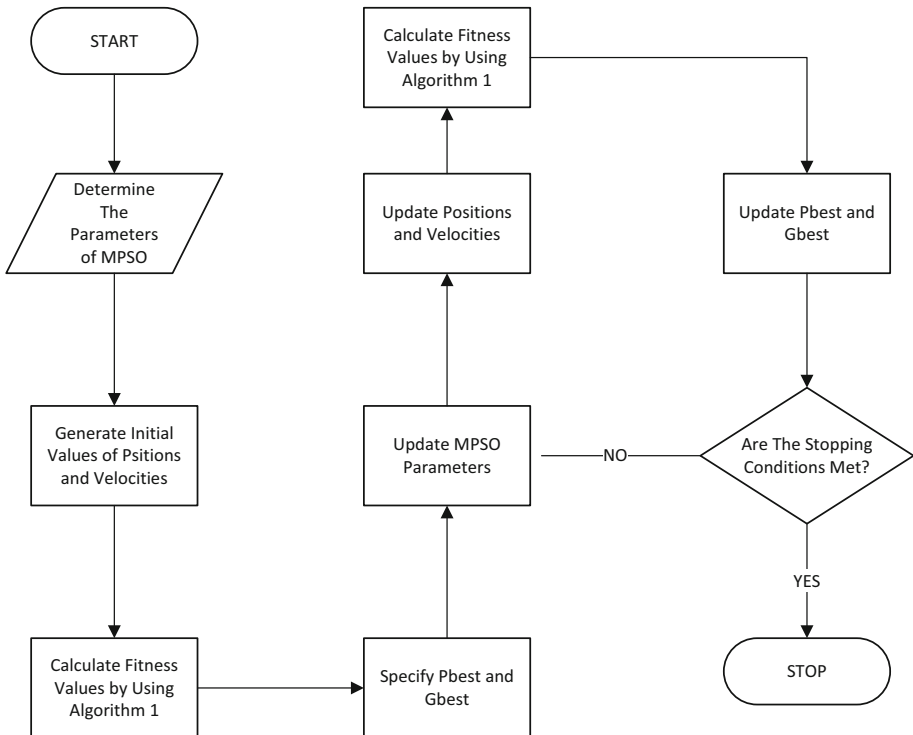


Fig. 3 The flow chart of algorithm 2

Table 1 Time series used in implementations and their some features

Time series no.	Series	Period	Observed period	Number of lag	The number of repetition	Test set size	Total analysis
1–5	DJF	2010–2014	Daily basis	2:5	30	10; 20; 40	450
6–10	BIST	2010–2014	Daily basis	2:5	30	10; 20; 40	450
11–15	NASDAQ	2010–2014	Daily basis	2:5	30	10; 20; 40	450
16–20	TAIEX	2010–2014	Daily basis	2:5	30	10; 20; 40	450
21	AUST	1956 Q1 –1994 Q1	Quarterly	4:8	1	16	1
22	TEC	01/2002 –12/2013	Monthly	2:16	1	12	1

where $rand_1$ and $rand_2$ are random values from the interval $[0, 1]$.

Step 7 Check the stopping criteria.

If the number of repetition maximum iteration number (itr_{max}) is reached then stop the process, or else repeat from Steps 3 to Step 7 until a predetermined maximum iteration number (itr_{max}) is reached.

When itr_{max} is reached the optimum values of weights and biases are specified by Eqs. 1 and 2. And then the training of AC-S-MNM is completed. The flow chart of algorithm 2 can be given as in Fig. 3.

3 The Implementations

In order to investigate the performance of the proposed AC-S-MNM, 22 different time series were analysed. These time series and its some features used in the implementations are given in Table 1. In implementations of AC-S-MNM and the other methods based on MPSO, the parameters were taken as $pn = 30$, $vm = 100$, $(c_{1i}, c_{1f}) = (2, 3)$, $(c_{2i}, c_{2f}) = (2, 3)$, $(w_1, w_2) = (1, 2)$, and $itr_{max} = 300$.

20 of these time series are daily basis stock exchange of Dow Jones Futures (DJF), Istanbul Stock Exchange (BIST), National Association of Securities Dealers Automated Quotations (NASDAQ), and Taiwan Stock Exchange Capitalization Weighted (TAIEX). In the application of these data sets, model order, in other words the number of inputs of the ANN, were taken from two to five and three different test set size were used as 10; 20; 40 and also for each of data set were analysed with 30 different random initialization. Taking into account all of these features, totally, $4 \times 3 \times 30 = 360$ different analysis were performed for each of data set. In this analysis process, in addition to the proposed model, Single Multiplicative Neuron Model ANN trained by MPSO (SMN-ANN-PSO), multilayer perceptron ANN trained by MPSO (MLP-ANN-PSO), and Single Multiplicative Recurrent Neuron Model ANN trained by MPSO (SMN-R-ANN-PSO) were implemented and obtained results were evaluated together in terms of root mean square error (RMSE) criteria for both training and test sets. A summary of the obtained results are given in Table 2.

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2} \quad (13)$$

In Table 2, ranking rates of the proposed model are given with respect to four different values as mean, minimum, maximum and standard deviation of RMSE. In consideration of Table 2, for DJF time series, it is seen that the proposed AC-S-MNM, with regards to mean of RMSE, has best value in all 360 implementations in both training and test sets. Moreover, while the proposed model has best performance in 86.66 percentages of implementations, it has second best performance in 11.67 percentages of implementations for test sets in the analysis of DJF.

When it comes to consideration the implementations of four different data sets (DJF, BIST, NASDAQ and TAIEX) as a whole, the proposed model has best values in terms of mean of RMSE in 99.17 percentages of implementations; in terms of minimum values of RMSE in 91.67 percentages of implementation; in respect to maximum values of error criteria in 100 percentages of implementation and in recognition of standard deviation of RMSE that is also an evidence of consistency of the models 73.33 percentages of implementation for the test sets (detailed results can be seen from supplementary tables).

Moreover, for a statistical evaluation as another emphasis of the outstanding performance of the proposed AC-S-MNM, Kruskal Wallis-H (KW-H) test performed and the results obtained from four different ANN models including the proposed AC-S-MNM were compared for DJF, BIST, NASDAQ and TAIEX data sets. In the comparison, significance level alpha was taken into account with two different levels as 0.05 and 0.10 and the p values obtained from KW-H test were compared to alpha. According to KW-H test's results, for whole alternating analysis including training and test data sets, there are significant differences among the performance of ANN models for both 0.05 and 0.10 significance levels apart from just one case for analysis of NASDAQ 2011/test set 10 when the number of input is 2 (see the Supplementary Tables). Even for these cases, there is a significant difference among ANN models in the level of significant 0.10 ($P = 0.099$). Considering the results both obtained from KW-H tests and given in Table 2 together, it is statistically said that the proposed AC-S-MNM has the best forecasting performance for 98.33 percentages and 99.17 percentage of whole analysis in the training and the test sets, respectively.

Secondly, Australian beer consumption data (AUST), a well-known real-time series data, between 1957 Q2 and 1994 Q1 was analysed. In the analysis of AUST given in Fig. 4, as in other previous studies, the number of inputs of the ANN was changed from 4 to 16 and the last 16 observations were taken as test set.

The best performance of the proposed model in terms of RMSE and mean absolute percentage error (MAPE), for AUST, is represented with the best results that obtained from 11 different models in Table 3. While the results of SARIMA (seasonal autoregressive integrated moving average), WMES (Winter's multiplicative exponential smoothing), MLP-ANN, RBF-ANN (radial bases function ANN) L&NL-ANN (linear and non-linear ANN), E-ANN (Elman ANN), MS-ANN (multiplicative seasonal ANN), and R-MNM-ANN (recurrent multiplicative neuron ANN) were taken from [16], the results of SMN-ANN-PSO, MLP-ANN-PSO, and SMN-R-ANN-PSO were obtained in this study by running MATLAB software program.

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \quad (14)$$

When all results in Table 3 are considered together, it is clearly seen that the proposed model has superior forecasting performance with 17.2390 RMSE value and 3.09% MAPE value. The graph of best forecasts obtained in case of model order 16 and observations of test data is given in Fig. 5. Figure 5 shows that the forecasts of AC-S-MNM are in good agreement with observations just as the scatter plot given in Fig. 6.

Table 2 Ranking rates of each model

	Proposed method				Data (%)
	First (%)	Second (%)	Third (%)	Fourth (%)	
Training					
Mean	100.00	0.00	0.00	0.00	DJF
Min	61.66	31.67	6.67	0.00	
Max	100.00	0.00	0.00	0.00	
SD	93.33	6.67	0.00	0.00	
Test					
Mean	100.00	0.00	0.00	0.00	
Min	86.66	11.67	1.67	0.00	
Max	100.00	0.00	0.00	0.00	
SD	75.00	21.67	3.33	0.00	
Training					
Mean	100.00	0.00	0.00	0.00	BIST
Min	81.66	16.67	1.67	0.00	
Max	100.00	0.00	0.00	0.00	
SD	88.33	10.00	1.67	0.00	
Test					
Mean	96.67	3.33	0.00	0.00	
Min	90.00	10.00	0.00	0.00	
Max	100.00	0.00	0.00	0.00	
SD	83.33	16.67	0.00	0.00	
Training					
Mean	93.33	1.67	0.00	0.00	NASDAQ
Min	53.33	31.67	13.33	1.67	
Max	98.33	1.67	0.00	0.00	
SD	88.33	10.00	1.67	0.00	
Test					
Mean	100.00	0.00	0.00	0.00	
Min	95.00	5.00	0.00	0.00	
Max	100.00	0.00	0.00	0.00	
SD	68.33	26.67	5.00	0.00	
Training					
Mean	100.00	0.00	0.00	0.00	TAIEX
Min	90.00	8.33	1.67	0.00	
Max	100.00	0.00	0.00	0.00	
SD	86.67	11.67	1.67	0.00	
Test					
Mean	100.00	0.00	0.00	0.00	
Min	95.00	5.00	0.00	0.00	
Max	100.00	0.00	0.00	0.00	
SD	66.67	23.33	5.00	5.00	

Table 2 continued

	Proposed method				Data (%)
	First (%)	Second (%)	Third (%)	Fourth (%)	
Training					
Mean	98.33	0.42	0.00	0.00	General
Min	71.66	22.09	5.84	0.42	
Max	99.58	0.42	0.00	0.00	
SD	89.17	9.59	1.25	0.00	
Test					
Mean	99.17	0.83	0.00	0.00	
Min	91.67	7.92	0.42	0.00	
Max	100.00	0.00	0.00	0.00	
SD	73.33	22.09	3.33	1.25	

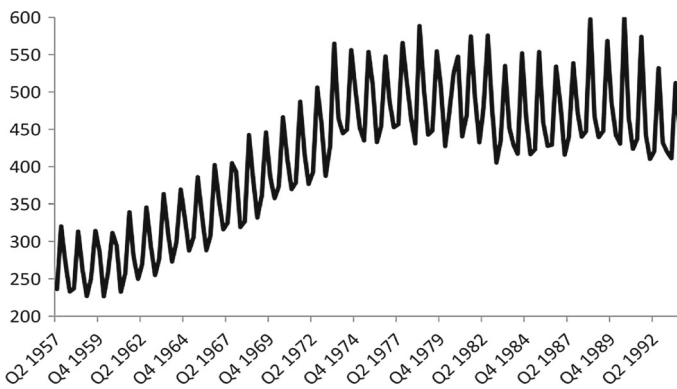


Fig. 4 The graph of AUST

Table 3 The performance criteria of models for AUST

Method	RMSE	MAPE (%)
SARIMA	47.0367	9.49
WMES	53.3295	10.72
MLP-ANN	24.1052	4.76
RBF-ANN	41.7000	6.86
L&NL-ANN	18.7888	3.57
E-ANN	22.6581	4.36
MS-ANN	22.1700	3.93
R-MNM-ANN	17.8573	3.29
SMN-ANN-PSO	18.3896	3.29
MLP-ANN-PSO	24.8945	4.47
SMN-R-ANN-PSO	22.2994	4.11
The proposed AC-S-MNM	<i>17.2930</i>	<i>3.09</i>

Italic values refer to the best performance values

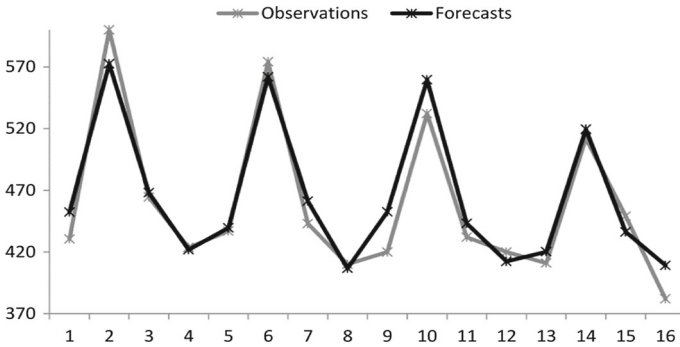


Fig. 5 The graph of forecasts obtained from AC-S-MNM and observations for test set of AUST

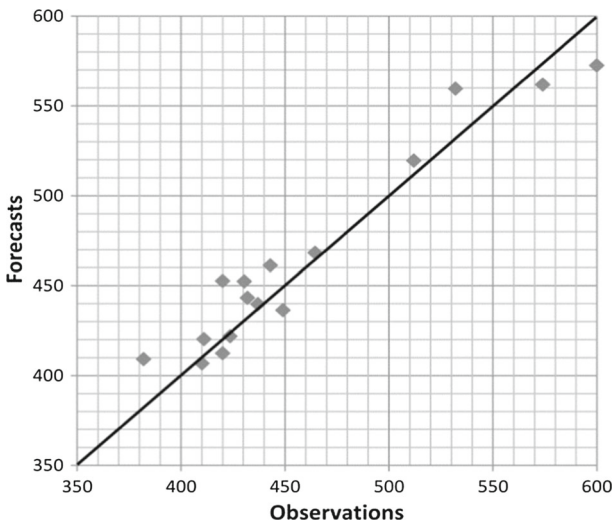


Fig. 6 The scatter plot of forecasts obtained from AC-S-MNM and observations for test set of AUST

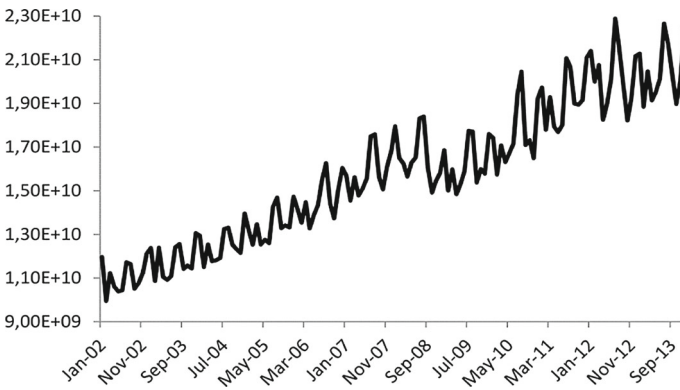


Fig. 7 The graph of ELC

Table 4 The performance criteria of models for ELC

Method	RMSE	MAPE (%)
SARIMA	9.1732E+08	3.88
MLP-ANN	1.0659E+09	3.98
L&NLANN	8.2098E+08	2.54
MS-ANN	8.1326E+08	3.01
SMN-ANN-PSO	6.7279E+08	2.89
MLP-ANN-PSO	9.1519E+08	3.39
SMN-R-ANN-PSO	6.5648E+08	2.72
The proposed AC-S-MNM	<i>6.2437E+08</i>	2.39

Italic values refer to the best performance values

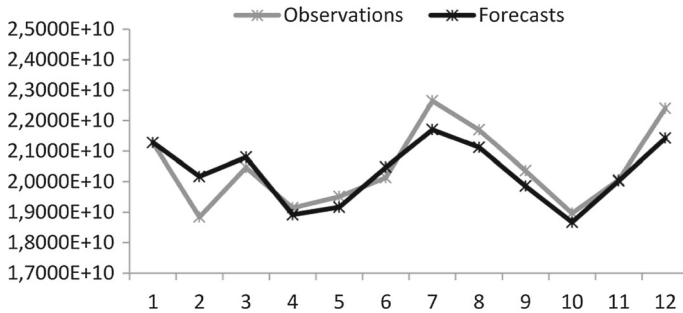


Fig. 8 The graph of forecasts obtained from AC-S-MNM and observations for test set of ELC

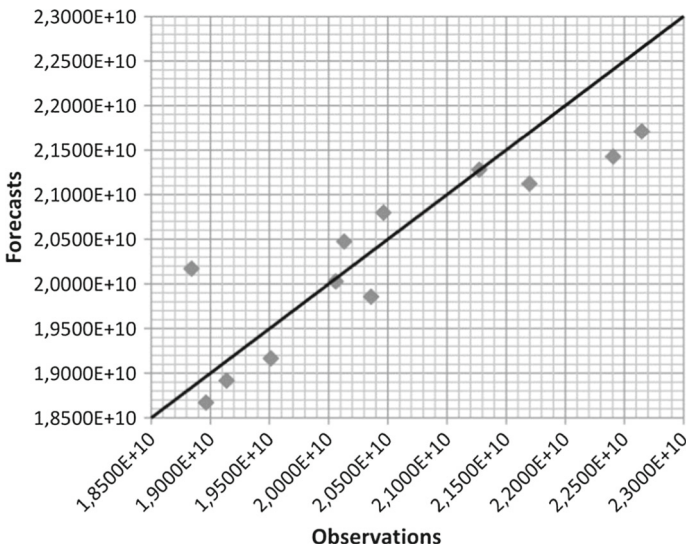


Fig. 9 The scatter plot of forecasts obtained from AC-S-MNM and observations for test set of ELC

Finally, Turkey Electricity Consumption data observed monthly between first month of 2002 and last month of 2013 (TEC) was used to evaluate the forecasting performance of AC-S-MNM. In the implementation of TEC which is given in Fig. 7, the model order was changed from 2 to 16, and the last 12 observations were taken as test set.

The best performance of the proposed model, for ELC, is represented with the best results that obtained from some other models in Table 4. Considering the results in Table 4, it is clearly seen that AC-S-SMN has superior forecasting performance with $6.2437\text{E}+08$ RMSE value and 2.39% MAPE value. The graph of best forecasts obtained in case of model order 15 and observations of test data are given in Fig. 8. From Fig. 8, we clearly see that the forecasts of AC-S-MNM are in compliance with observations. Moreover the scatter plot given in Fig. 9 is a supportive evidence for the cohesion of the forecasts with observations.

4 Conclusion and Discussion

Nowadays, although researchers have commonly take advantage of MLP and S-MNM in time series prediction problems and they have various high quality features, each of them also contains some issues itself such as architecture determination problem and having a strict model-based structure. However, a more practical and successful ANN model can be used for time series prediction by integrating the superior aspect of S-MNM and MLP by eliminating the weaknesses of them. In this study, from this point of view, we proposed an ANN model—AC-S-MNM—that does not have the architecture selection problem just as S-MNM and have a feature of data-based just as MLP. In AC-S-MNM, weights and biases are produced through autoregressive equations. In the producing process of weights and biases, since the time index of each observation are considered, the model is evaluated as a data-based. The coefficients of autoregressive equations and the initial values of the weights and biases were determined by using PSO in an optimization process. The proposed model was applied to various real-world time series data sets and the obtained results were compared to the results of some well-known methods. And the results also show that AC-S-MNM exhibits significantly better performance as compared to the existing models. In future studies, some other equation structure can be utilized to determine the weights and biases.

References

1. Adhikari R, Agrawal RK (2014) A combination of artificial neural network and random walk models for financial time series forecasting. *Neural Comput Appl* 24:1441–1449
2. Aladag CH, Egrioglu E, Yolcu U (2014) Robust multilayer neural network based on median neuron model. *Neural Comput Appl* 24:945–956
3. Almonacid F, Pérez-Higueras PJ, Fernández E, Hontoria L (2014) A methodology based on dynamic artificial neural network for short-term forecasting of the power output of a PV generator. *Energy Convers Manage* 85:389–398
4. Ata R (2015) Artificial neural networks applications in wind energy systems: a review. *Renew Sustain Energy Rev* 49:534–562
5. Babu N, Reddy BE (2014) A moving-average filter based hybrid ARIMA-ANN model for forecasting time series data. *Appl Soft Comput* 23:27–38
6. Basu M, Ho TK (1999) Learning behavior of single neuron classifiers on linearly separable or non-separable inputs. In: *IEEE LICNN'99*
7. Chen XY, Chau KW, Busari AO (2015) A comparative study of population-based optimization algorithms for downstream river flow forecasting by a hybrid neural network model. *Eng Appl Artif Intell* 46:258–268
8. Cheng C-T, Niu W-J, Feng Z-K, Shen J-J, Chau K-W (2015) Daily reservoir runoff forecasting method using artificial neural network based on quantum-behaved particle swarm optimization. *Water* 7:4232–4246
9. Claveria O, Monte E, Torra S (2015) Tourism demand forecasting with neural network models: different ways of treating information. *Int J Tourism Res* 17:492–500

10. Crone S, Kourentzes N (2010) Naive support vector regression and multilayer perceptron benchmarks for the 2010 neural network grand competition (NNGC) on time series prediction. In: Proceedings of the international joint conference on neural networks, IJCNN'10, Barcelona, IEEE, New York
11. Crone S, Nikolopoulos K, Hibon M (2011) New evidence on the accuracy of computational intelligence for monthly time series prediction—results of the NN3 forecasting competition. *Int J Forecast* 27(3):635–660
12. De Mattos-Neto PSG, Madeiro F, Ferreira TAE, Cavalcanti GDC (2014) Hybrid intelligent system for air quality forecasting using phase adjustment. *Eng Appl Artif Intell* 32:185–191
13. De-Giorgi MG, Campilongo S, Ficarella A, Congedo PM (2014) Comparison Between Wind Power Prediction Models Based on Wavelet Decomposition with Least-Squares Support Vector Machine (LS-SVM) and Artificial Neural Network (ANN). *Energies* 7:5251–5272
14. Deo RC, Şahin M (2015) Application of the Artificial Neural Network model for prediction of monthly standardized precipitation and evapotranspiration index using hydrometeorological parameters and climate indices in eastern Australia. *Atmos Res* 161–162:65–81
15. Egrioglu E, Aladag CH, Gunay S (2008) A new model selection strategy in artificial neural network. *Appl Soft Comput* 195:591–597
16. Egrioglu E, Yolcu U, Aladag CH, Bas E (2015) Recurrent multiplicative neuron model artificial neural network for non-linear time series forecasting. *Neural Process Lett* 41:249–258
17. Falamarzi Y, Palizdan N, Huang YF, Lee TS (2014) Estimating evapotranspiration from temperature and wind speed data using artificial and wavelet neural networks (WNNs). *Agric Water Manag* 140:26–36
18. Fard AK, Akbari-Zadeh M-R (2014) A hybrid method based on wavelet, ANN and ARIMA model for short-term load forecasting. *J Exp Theor Artif Intell* 26(2):167–182
19. Firmino PRA, De Mattos-Neto PSG, Ferreira TAE (2015) Error modeling approach to improve time series forecasters. *Neurocomputing* 153:242–254
20. Godarzi AA, Amiri RM, Talaei A, Jamasb T (2014) Predicting oil price movements: a dynamic artificial neural network approach. *Energy Policy* 68:371–382
21. Gong Y, Zhang Y, Lan S, Wang H (2016) A comparative study of artificial neural networks, support vector machines and adaptive neuro fuzzy inference system for forecasting groundwater levels near Lake Okeechobee, Florida. *Water Resour Manage* 30:375–391
22. He Z, Zhang Y, Guo Q, Zhao X (2014) Comparative study of artificial neural networks and wavelet artificial neural networks for groundwater depth data forecasting with various curve fractal dimensions. *Water Resour Manage* 28:5297–5317
23. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks. IEEE Press, Piscataway pp 1942–1948
24. Kim M-K (2015) A new approach to short-term price forecast strategy with an artificial neural network approach: application to the Nord Pool. *J Electr Eng Technol* 10(4):1480–1491
25. Koprinska I, Rana M, Troncoso A, Martínez-Álvarez F (2014) Combining pattern sequence similarity with neural networks for forecasting electricity demand time series, 2013. In: The international joint conference on neural networks (IJCNN), pp 1–8
26. Labib R (1999) New single neuron structure for solving nonlinear problems. In: IEEE IJCNN'99, pp 617–620
27. Laboissiere LA, Fernandes RAS, Lage GG (2015) Maximum and minimum stock price forecasting of Brazilian power distribution companies based on artificial neural networks. *Appl Soft Comput* 35:66–74
28. Ma Y, Jiang C, Hou Z, Wang C (2006) The formulation of the optimal strategies for the electricity producers based on the particle swarm optimization algorithm. *IEEE Trans Power Syst* 21(4):1663–1671
29. Martínez-Álvarez F, Troncoso A, Asencio-Cortés G, Riquelme JC (2015) A survey on data mining techniques applied to electricity-related time series forecasting. *Energies* 8:13162–13193
30. McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys* 5:115–133
31. Piotrowski AP, Napiorkowski MJ, Napiorkowski JJ, Osuch M (2015) Comparing various artificial neural network types for water temperature prediction in rivers. *J Hydrol* 529:302–315
32. Plate TA (2000) Randomly connected sigma-pi neurons can form associate or networks. *NETCNS Netw Comput Neural Syst* 11:321–322
33. Rahman NHA, Lee MH, Suhartono Latif MT (2015) Artificial neural networks and fuzzy time series forecasting: an application to air quality. *Qual Quant* 49:2633–2647
34. Reyes J, Morales-Esteban A, Martínez-Álvarez F (2013) Neural networks to predict earthquakes in Chile. *Appl Soft Comput* 13:1314–1328
35. Ruiz-Aguilar JJ, Turias JJ, Jiménez-Come MJ (2014) Hybrid approaches based on SARIMA and artificial neural networks for inspection time series forecasting. *Transp Res Part E* 67:1–13
36. Rumelhart E, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation, chapter 8. The M.I.T. Press, Cambridge, pp 318–362

37. Shi Y, Eberhart RC (1999) Empirical study of particle swarm optimization. *Proc IEEE Int Congr Evol Comput* 3:101–106
38. Shuhua W, Xueliang Z, Hainan L, Shuyang L, Jiaying W (2005) A modified particle swarm optimization algorithm. In: *International conference of neural networks and brain, 2005. ICNN&B '05*, pp 318–321
39. Smith C, Jin Y (2014) Evolutionary multi-objective generation of recurrent neural network ensembles for time series prediction. *Neurocomputing* 143:302–311
40. Voyant C, Randimbivololona P, Nivet ML, Paolic C, Musellic M (2014) Twenty four hours ahead global irradiation forecasting using multi-layer perceptron. *Meteorol Appl* 21:644–655
41. Wang W-C, Chau K-W, Qiu L, Chen Y-B (2015) Improving forecasting accuracy of medium and long-term runoff using artificial neural network based on EEMD decomposition. *Environ Res* 139:46–54
42. Wei S, Yang H, Song J, Abbaspour K, Xu Z (2013) A wavelet-neural network hybrid modelling approach for estimating and predicting river monthly flows. *Hydrol Sci J* 58:374–389
43. Yadav RN, Kalra PK, John J (2007) Time series prediction with single multiplicative neuron model. *Appl Soft Comput* 7:1157–1163
44. Zhang CN, Zhao M, Wang M (2000) Logic operations based on single neuron rational model. *IEEE Trans Neural Netw* 11:739–747
45. Zhang G (2003) Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* 50:159–175
46. Zhang G, Patuwo BE, Hu YM (1998) Forecasting with artificial neural networks: the state of the art. *Int J Forecast* 14:35–62