Neural Comput & Applic (2014) 24:945–956
DOI 10.1007/s00521-012-1315-5

ORIGINAL ARTICLE

Robust multilayer neural network based on median neuron model

Cagdas Hakan Aladag · Erol Egrioglu ·
Ufuk Yolcu

Received: 15 May 2012 / Accepted: 11 December 2012 / Published online: 3 January 2013
© Springer-Verlag London 2013

Abstract Multilayer perceptron has been widely used in time series forecasting for last two decades. However, it is a well-known fact that the forecasting performance of multilayer perceptron is negatively affected when data have outliers and this is an important problem. In recent years, some alternative neuron models such as generalized-mean neuron, geometric mean neuron, and single multiplicative neuron have been also proposed in the literature. However, it is expected that forecasting performance of artificial neural network approaches based on these neuron models can be also negatively affected by outliers since the aggregation function employed in these models is based on mean value. In this study, a new multilayer feed forward neural network, which is called median neuron model multilayer feed forward (MNM-MFF) model, is proposed in order to deal with this problem caused by outliers and to reach high accuracy level. In the proposed model, unlike other models suggested in the literature, MNM which has median-based aggregation function is employed. MNM is also firstly defined in this study. MNM-MFF is a robust neural network method since aggregation functions in MNM-MFF are based on median, which is not affected much by outliers. In addition, to train MNM-MFF model, particle swarm optimization method was utilized. MNM-MFF was applied to two well-known time series in order to evaluate the performance of the proposed approach. As a result of the implementation, it was observed that the proposed MNM-MFF model has high forecasting accuracy and it is not affected by outlier as much as multilayer perceptron model. Proposed method brings improvement in 7 % for data without outlier, in 90 % for data with outlier, in 95 % for data with bigger outlier.

Keywords Feed forward · Forecasting · Median neuron model · Particle swarm optimization · Robust neural networks · Outlier

1 Introduction

In the literature, various approaches have been used for forecasting. Conventional forecasting methods can be insufficient for real-life time series since these approaches need some assumptions to be satisfied [1]. Not requiring any assumption such as linearity and normal distribution in forecasting time series makes artificial neural networks applicable for many fields [4]. In recent years, an effective way to forecast time series has been to utilize artificial neural networks [5]. This approach has been successfully used for forecasting in various implementations [2].

Various neuron models have been proposed in the literature [6, 14, 16, 17, 26]. The most preferred artificial neural network type is multilayer perceptron (MLP) introduced by Rumelhart et al. [18]. When real-life problems are solved using standard artificial neural networks such as MLP, it requires large number of neurons in the architecture [20]. A neuron having higher-order statistics

C. H. Aladag (✉)
Department of Statistics, Faculty of Science,
Hacettepe University, 06800 Ankara, Turkey
e-mail: aladag@hacettepe.edu.tr

E. Egrioglu
Department of Statistics, Faculty of Arts and Science,
Ondokuz Mayis University, 55139 Samsun, Turkey
e-mail: erole@omu.edu.tr

U. Yolcu
Department of Statistics, Faculty of Art and Science,
Giresun University, 28000 Giresun, Turkey
e-mail: ufuk.yolcu@giresun.edu.tr

can produce superior neural network with comparatively lesser number of neurons [20]. Thus, higher-order neural networks have been suggested by Chaturvedi et al. [8], Giles and Maxwell [9], Homma and Gupta [11], Sinha et al. [21], Taylor and Commbes [22]. Higher-order neurons have demonstrated improved computational power and generalization ability. However, these models are difficult to train because of a combinatorial explosion of higher-order term as the number of inputs to the neuron increases [20]. In addition, it is a well-known fact that the forecasting performance of MLP is negatively affected when data include outliers [10, 25]. In recent years, some artificial neural networks models which are based on some neuron models such as generalized-mean neuron (GMN) [23], geometric mean neuron (G-MN) [20], and single multiplicative neuron (SMN) [24] have been proposed as an alternative for MLP. Like MLP, these suggested models can also be negatively affected by outliers since aggregation functions in these models are based on mean.

A MNM is firstly introduced in this study, and a new feed forward neural networks approach based on MNM is proposed in order to deal with outlier problem. In the proposed MNM, unlike other neuron models, aggregation function based on median, which is not affected much by outliers, is employed instead of using functions based on summation or mean. It is a fact that unlike other measures of location statistics such as mean, median is not affected much by outliers of a data set. Using a mean-based aggregation function in a neuron will prevent that this neuron produces an extreme output value for an outlier input value. Therefore, MNM-MFF model consists of MNM is a robust multilayer neural network approach that is not affected by outliers. In the training process of MNM-MFF model, it is very hard to obtain derivate of cost function with respect to the weights of the model since median-based aggregation functions are used in the model. That means it is not easy to use back propagation learning algorithm to determine the best values of the weights. Therefore, the modified particle swarm optimization method [3] is utilized to train MNM-MFF model. To assess the forecasting performance of the proposed model MNM-MFF, it was applied to two well-known real-time series which are Australian beer consumption and Box-Jenkins gas furnace. In addition, different data scenarios were considered in the implementation to examine the performance of the proposed approach better. Furthermore, other forecasting models available in the literature were also used for the aim of comparison.

The remaining parts of the paper are organized as follows. The modified particle swarm optimization method that is used as a learning algorithm to train MNM-MFF model is briefly summarized in the next section. MNM is introduced in Sect. 3. In Sect. 4, MNM-MFF model is described and how the modified particle swarm optimization method is employed to train this model is addressed. Section 5 presents the implementation and the obtained results. Finally, the results are discussed in the last section.

2 The modified particle swarm optimization (MPSO)

Particle swarm optimization is a population-based heuristic algorithm, and it was firstly proposed by Kennedy and Eberhart [13]. 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. In this study, MPSO was used to train MNM-MFF model. The detailed information about MPSO method can be found in [3]. The MPSO algorithm has time varying inertia weight like in [19]. In a similar way, this algorithm also has time varying acceleration coefficient like in [15].

Algorithm 1 The modified particle swarm optimization

Step 1 Positions of each kth ($k = 1, 2, \ldots, pn$) particles are randomly determined and kept in a vector X_k given as follows:

$$X_k = \{x_{k1}, x_{k2}, \ldots, x_{kd}\}, \quad k = 1, 2, \ldots, pn \tag{1}$$

where x_{ki} ($i = 1,2,\ldots,d$) represents ith position of kth particle. pn and d represent the number of particles in a swarm and positions, respectively.

Step 2 Velocities are randomly determined and stored in a vector V_k given below.

$$V_k = \{v_{k1}, v_{k2}, \ldots, v_{kd}\}, \quad k = 1, 2, \ldots, pn \tag{2}$$

Step 3 According to the evaluation function, *Pbest* and *Gbest* particles given in (3) and (4), respectively, are determined.

$$Pbest_k = \left(p_{k,1}, p_{k,2}, \ldots, p_{k,d}\right), \quad k = 1, 2, \ldots, pn \tag{3}$$

$$Gbest = \left(p_{g,1}, p_{g,2}, \ldots, p_{g,d}\right) \tag{4}$$

where $Pbest_k$ is a vector stores the positions corresponding to the kth particle's best individual performance, and *Gbest* and g represent the best particle and index number of best particle respectively, which has the best evaluation function value, found so far.

Step 4 Let c_1 and c_2 represent cognitive and social coefficients, respectively, and w is the inertia parameter. Let (c_{1i}, c_{1f}), (c_{2i}, c_{2f}), and (w_1, w_2) be the intervals which includes possible values for c_1, c_2 and w, respectively. At each iteration, these parameters are calculated by using the formulas given in (5), (6), and (7).

$$c_1 = \left(c_{1f} - c_{1i}\right)\frac{t}{\max t} + c_{1i} \tag{5}$$

$$c_2 = \left(c_{2f} - c_{2i}\right)\frac{t}{\max t} + c_{2i} \tag{6}$$

$$w = (w_2 - w_1)\frac{\max t - t}{\max t} + w_1 \tag{7}$$

where max*t*, *t*, *f*, and *i* represent maximum iteration number, current iteration number, final, and initial, respectively.

Step 5 Values of velocities and positions are updated by using the formulas given in (8) and (9), respectively.

$$v_{i,d}^{t+1} = \Big[w \times v_{i,d}^{t} + c_1 \times \mathrm{rand}_1 \times \left(p_{i,d} - x_{i,d}\right) + c_2 \times \mathrm{rand}_2 \times \left(p_{g,d} - x_{i,d}\right)\Big] \tag{8}$$

$$x_{i,d}^{t+1} = x_{i,d} + v_{i,d}^{t+1} \tag{9}$$

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

Step 6 Steps 3 to 5 are repeated until a predetermined maximum iteration number (max*t*) is reached.

3 Median neuron model

The first artificial neuron model was proposed by McCulloch and Pitts [16]. Then, various neuron models have been proposed in the literature. One of the most preferred types of artificial neural networks is MLP. In each neuron of MLP, the function in (10) has been generally employed as aggregation function.

$$net\left(x_j, w_j\right) = \sum_{j=1}^{N} w_j x_j + w_0 \tag{10}$$

where x_j, w_j, N ($j = 0, 1, 2, \ldots, N$) represent input signals, weights, and number of input signals, respectively. And, w_0 is the weight for bias. It is very clear that the neuron model given in (10) is negatively affected by input signals from outliers since this model is based on summation operation. Besides, mean-based neuron models such as GMN and G-MN have same problem since mean is also negatively affected by outliers. In this study, a new neuron model MNM, in which median is utilized as aggregation function, is proposed to deal with outlier problem. It is a well-known fact that median is not affected much by outliers so MNM is not affected by outliers. MNM is illustrated in Fig. 1.

In Fig. 1, *y* and *f* represent output signal of the neuron and activation function, respectively. The bias value is 1 as seen from the figure. *net* is the activation value obtained from aggregation function. This value is calculated as follows:

$$net = Median(w_1x_1, w_2x_2, \ldots, w_Nx_N, w_0) \tag{11}$$

where x_j, w_j, N ($j = 0,1,2, \ldots ,N$) represent input signals, weights, and number of input signals, respectively. And, w_0 is the weight for bias.

4 Multilayer feed forward network with median neuron model

MNM-MFF model proposed in this study is a multilayer feed forward neural network model. This neural network model is composed of MNM. Architecture structure of the proposed MNM-MFF model that has N and M neurons in input and hidden layers, respectively, is shown in Fig. 2. As seen from this figure, input and output vectors of the model are $X = [x_1, x_2, \ldots, x_N]$ and $[y]$, respectively.

If $w_{h_{ij}}$ is a weight that connects the *i*th hidden neuron with *j*th input, the activation value of the *i*th hidden neuron can be given as

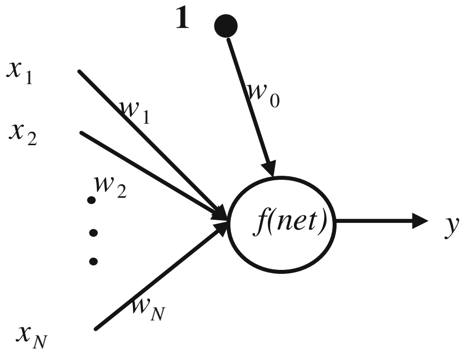

Fig. 1 MNM

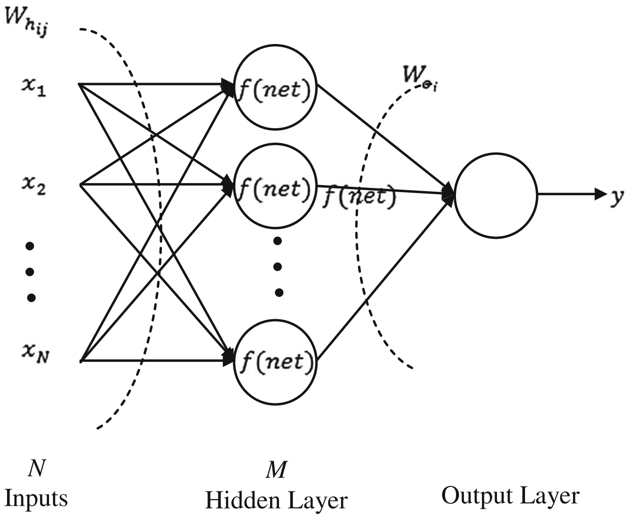

Fig. 2 The architecture of MNM-MFF

$$net_{h_i} = Median(w_{h_{i1}}x_1, w_{h_{i2}}x_2, \ldots, w_{h_{iN}}x_N, w_{h_{i0}}),\quad i = 1, 2, \ldots, M \tag{12}$$

where $w_{h_{i0}}$ is the weight for the bias of the ith hidden neuron. The nonlinear transformation performed by each of M neurons in the network is given as

$$y_{h_i} = f(net_{h_i}), \quad i = 1, 2, \ldots, M \tag{13}$$

where f denotes a sigmoid function. Similarly, the output of neuron in the output layer can be given as (15).

$$net = Median(w_{o_1}y_{h_1}, w_{o_2}y_{h_2}, \ldots, w_{o_M}y_{h_M}, w_{o_0}) \tag{14}$$

$$y = f(net) \tag{15}$$

where w_{o_i} is the weight that connects the ith neuron of hidden layer to neuron of the output layer, and w_{o_0} is the weight for the bias to corresponding output layer neuron.

To train MNM-MFF whose architecture structure is shown in Fig. 2, MPSO method which is presented in Sect. 2 is utilized. A particle in MPSO consists of positions which are weights of MNM-MFF. A particle is presented in Fig. 3.

In MPSO algorithm, mean square error (MSE) is employed as evaluation function. The formula in (16) can be used to calculate MSE value.

$$\text{MSE} = \frac{1}{n}\sum_{t=1}^{n}(\text{output}_t - \text{target}_t)^2 \tag{16}$$

where n represents the number of learning samples. The algorithm of MPSO approach used in training process of MNM-MFF can be given as follows:

Algorithm 2 The modified particle swarm optimization to train MNM-MFF

Step 1 The parameters of the method are determined.

In the first step, the parameters which direct the MPSO algorithm are determined. These parameters are pn, vm, c_{1i}, c_{1f}, c_{2i}, c_{2f}, w_1, and w_2 that were given in Sect. 2. Where $-vm$, vm show that minimum and maximum velocities size.

Step 2 Initial values of positions and velocities are determined.

The initial positions and velocities of each particle in a swarm are randomly generated from uniform distribution (0,1) and ($-vm$, vm), respectively.

Step 3 Evaluation function values are computed.

Evaluation function values for each particle are calculated. Evaluation function is MSE whose formula given in (16).

Step 4 $Pbest_k$ ($k = 1, 2, \ldots, pn$) and $Gbest$ are determined due to evaluation function values calculated in the previous step.

$Pbest_k$ is a vector stores the positions corresponding to the kth particle's best individual performance, and $Gbest$ is the best particle, which has the best evaluation function value, found so far.

Step 5 The parameters 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 (5), (6), and (7).

Step 6 New values of positions and velocities are calculated.

New values of positions and velocities for each particle are computed by using the formulas given in (8) and (9). If

Fig. 3 Structure of a particle in MPSO

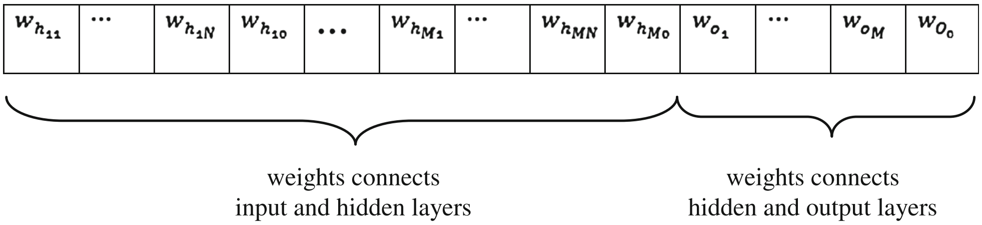

Fig. 4 1956 Q1—1994 Q1 quarterly Australian beer consumption

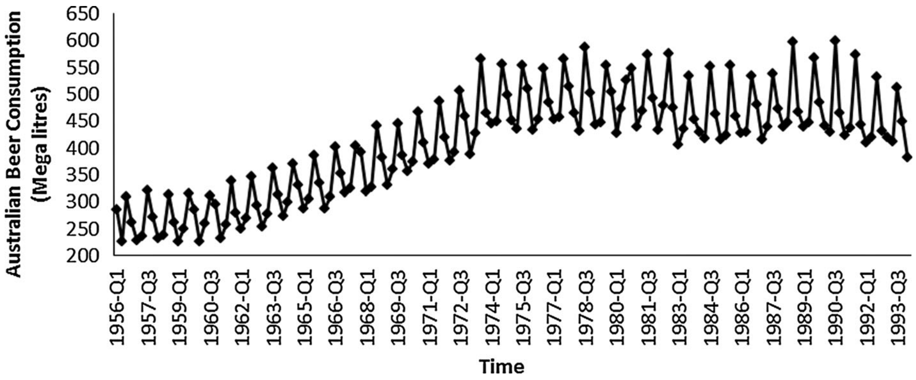

Table 1 The MSE values calculated over the test set for Case 1

Method	Model	MSE
SARIMA	(1,1,0)(0,1,1)	2,212.45
WMES	Trend-seasonal	2,844.04
MLP	4-3-1	582.61
PSO-MLP	4-2-1	2,005.07
MNM-MFF	4-1-1	538.48

Table 2 The training and testing MSE values are obtained from different architectures for MLP

	Number of hidden layer neurons			
	1	2	3	4
Training data				
Number of input layer neuron				
1	3,071.04	2,876.79	2,896.57	2,894.26
2	3,040.25	1,271.57	3,040.24	1,172.51
3	1,849.97	40,972.80	493.18	319.81
4	417.38	367.44	284.36	209.11
Test data				
Number of input layer neuron				
1	4,108.38	4,118.67	4,288.16	4,254.04
2	4,139.15	1,751.77	4,135.25	1,955.43
3	3,955.80	57,618.43	1,483.99	708.83
4	751.71	703.05	582.61	721.95

Table 3 The training and testing MSE values are obtained from different architectures for PSO-MLP

	Number of hidden layer neurons			
	1	2	3	4
Training data				
Number of input layer neuron				
1	3,486.49	3,311.56	5,540.58	3,444.82
2	3,932.92	4,243.54	4,946.45	4,659.43
3	3,002.11	2,959.94	2,790.66	4,704.31
4	2,397.70	2,289.59	3,423.24	5,227.17
Test data				
Number of input layer neuron				
1	4,786.70	4,225.31	3,986.68	3,672.37
2	4,040.10	5,550.34	3,950.50	4,696.87
3	3,858.48	4,435.14	5,434.63	7,117.05
4	3,577.65	2,005.07	4,067.21	3,688.90

maximum iteration number is reached, the algorithm goes to Step 3; otherwise, it goes to Step 7.

Step 7 The optimal solution is determined.

Table 4 The training and testing MSE values are obtained from different architectures for MNM-MFF

	Number of hidden layer neurons			
	1	2	3	4
Training data				
Number of input layer neuron				
1	3,516.60	2,993.79	3,161.90	3,024.09
2	3,287.07	5,488.02	3,140.49	5,488.02
3	1,865.84	5,205.32	1,267.48	4,816.52
4	499.26	3,903.88	515.44	3,887.11
Test data				
Number of input layer neuron				
1	4,796.17	4,066.55	4,251.82	3,979.26
2	4,192.75	4,466.25	4,314.72	4,466.43
3	3,188.76	4,141.72	2,910.69	4,411.00
4	538.48	2,798.88	820.14	2,895.61

Table 5 The training and testing performances of MLP, PSO-MLP, and MNM-MFF for Case 1

	n	MLP	PSO-MLP	MNM-MFF
Training data				
Mean	16	4,073.58	3,772.53	3,253.80
SD	16	9,911.00	1,011.71	1,596.40
Test data				
Mean	16	5,954.83	4,318.32	3,514.95
SD	16	13,866.63	1,104.30	1,272.76

SD standard deviation

The elements of *Gbest* are taken as the optimal weight values of the MNM-MFF.

5 The application

In order to evaluate the forecasting performance of MNM-MFF model, it was applied to two well-known real-time series which are Australian beer consumption and Box-Jenkins gas furnace. When MNM-MFF and other neural networks models were used for forecasting, lagged variables of the time series were taken as inputs of models.

5.1 Australian beer consumption data

The first time series is quarterly Australian beer consumption [12] between 1956 Q1 and 1994 Q1. The graph of this time series is shown in Fig. 4. The last 16 observations of the time series are used for test set. For Australian beer consumption, three data scenarios were

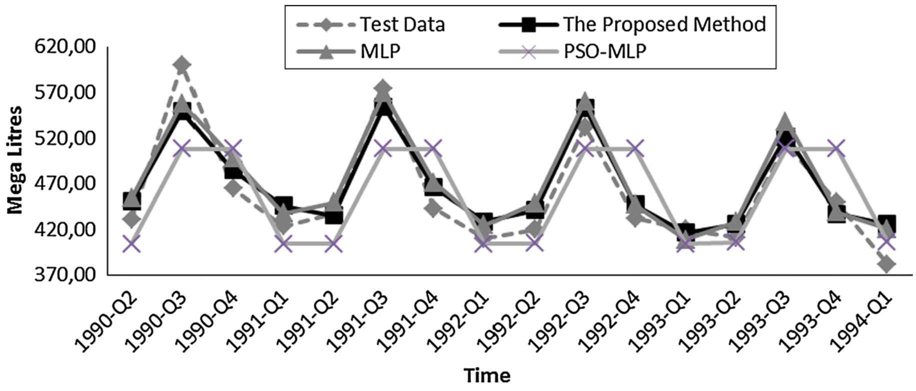

Fig. 5 The prediction results of MLP, PSO-MLP, and proposed method for Case 1

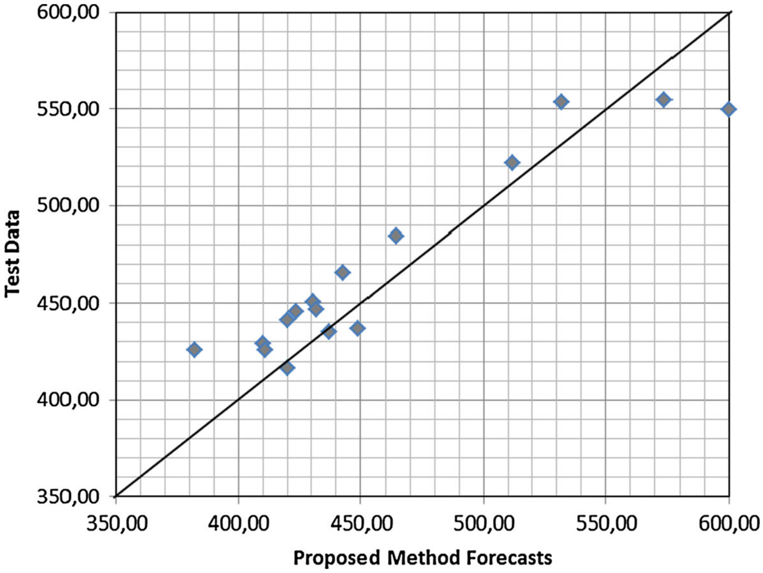

Fig. 6 Scatter plot of forecasts and observations for Case 1

Table 6 The training and testing MSE values are obtained from different architectures for MLP

	Number of hidden layer neurons			
	1	2	3	4
Training data				
Number of input layer neuron				
1	59,198.89	59,194.50	100,754.68	55,833.34
2	59,099.78	100,687.88	53,068.71	100,687.87
3	100,668.45	3,235.96	2,456.08	1,287.74
4	100,647.48	52,447.49	58,098.04	100,671.96
Test data				
Number of input layer neuron				
1	4,437.26	4,419.25	57,618.98	3,682.92
2	4,377.86	57,618.98	3,761.84	57,618.97
3	57,614.59	3,505.10	3,633.02	1,658.80
4	57,590.82	1,068.01	4,085.72	57,618.98

Table 7 The training and testing MSE values are obtained from different architectures for PSO-MLP

	Number of hidden layer neurons			
	1	2	3	4
Training data				
Number of input layer neuron				
1	56,611.81	56,696.63	58,734.83	59,646.08
2	59,467.31	54,933.55	59,522.14	59,712.03
3	60,044.17	59,959.30	59,519.23	60,039.69
4	57,808.43	57,303.23	59,421.21	60,132.28
Test data				
Number of input layer neuron				
1	3,686.47	3,705.32	5,015.05	4,425.78
2	4,243.75	3,911.14	4,639.97	4,470.36
3	4,151.49	4,156.60	4,218.89	4,252.87
4	3,114.27	3,765.91	3,834.07	4,241.61

examined to assess the performance of the proposed approach in case of the data have outliers. In the first case, seasonal autoregressive integrated moving average (SARIMA), Winter's multiplicative exponential smoothing (WMES), MLP, and MNM-MFF methods were applied to original Australian beer consumption data set between 1956 Q1 and 1994 Q1. In the second case, an outlier value was obtained by multiplying the maximum observation value of the original data by 5, and then, this outlier was added to data. In the last scenarios, outlier value was obtained by multiplying the maximum observation value of the original data by 10, and then this outlier was added to data.

Table 8 The training and testing MSE values are obtained from different architectures for MNM-MFF

	Number of hidden layer neurons			
	1	2	3	4
Training data				
Number of input layer neuron				
1	59,477.55	58,429.68	59,481.45	59,539.03
2	58,917.54	57,346.26	58,598.12	57,612.87
3	58,360.28	58,707.37	58,265.46	58,442.06
4	56,410.00	58,137.66	56,670.90	57,565.44
Test data				
Number of input layer neuron				
1	4,309.46	4,257.44	4,138.77	4,694.39
2	4,399.04	4,030.70	4,495.15	4,148.23
3	4,080.28	3,762.47	3,655.18	3,914.54
4	848.87	2,516.91	1,749.87	2,720.77

Table 9 The training and testing performances of MLP, PSO-MLP, and MNM-MFF for Case 2

	n	MLP	PSO-MLP	MNM-MFF
Training data				
Mean	16	63,002.42	58,721.99	58,247.60
SD	16	36,420.52	1,568.73	935.43
Test data				
Mean	16	23,769.44	4,114.60	3,607.63
SD	16	27,090.36	444.92	1,084.59

SD standard deviation

Case 1 Original Australian beer consumption data are forecasted by using SARIMA, WMES, MLP, and the proposed model MNM-MFF. When both MLP and MNM-MFF were employed, the number of input was changed between 1 and 4 since the period of this time series is 4. And, for both methods, the number of neuron in hidden layer was changed from 1 to 8. Thus, 32 different architectures were totally examined for each neural networks method to determine the best architecture. The obtained best architectures can be seen in Table 1. The best models and corresponding MSE values obtained from all methods are also summarized in Table 1.

When Table 1 is examined, it is seen that artificial neural networks methods MLP, PSO-MLP, and MNM-MFF produced better forecasts than those obtained from other conventional methods in terms of MSE. In addition, it is observed that the proposed MNM-MFF method has the best forecasting accuracy. When the data are forecasted with the proposed method, 4-1-1 was found as the best architecture. That is, the best architecture has 4, 1, and 1 neuron in input, hidden, and output layers, respectively.

To examine the performance of MNM-MFF better, MNM-MFF, PSO-MLP, and MLP were compared in more details. For training and test sets, the results obtained from 16 architectures are given in Tables 2, 3 and 4. Mean and standard deviation of MSE values calculated from 16 architectures are presented in Table 5.

As seen from Table 5, for both training and test sets, mean values of errors produced by MNM-MFF are less than those obtained from MLP and PSO-MLP. Furthermore, MNM-MFF has minimum standard deviation values for both training and test sets. In other words, MNM-MFF and PSO-MLP give more consistent results than those

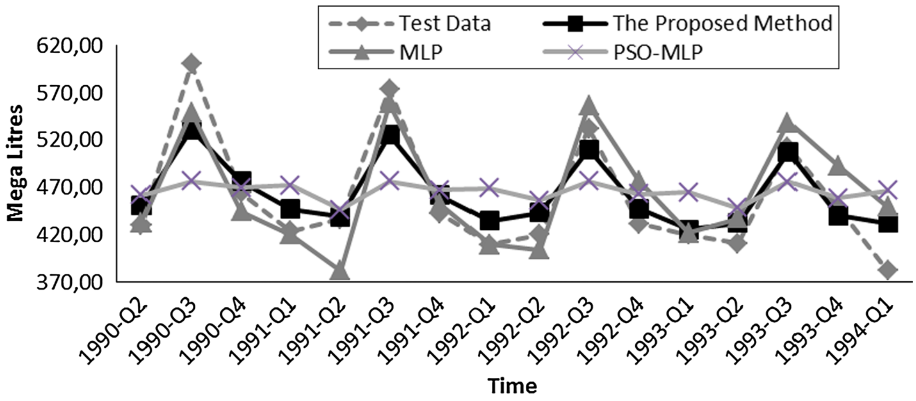

Fig. 7 The prediction results of MLP and proposed method for Case 2

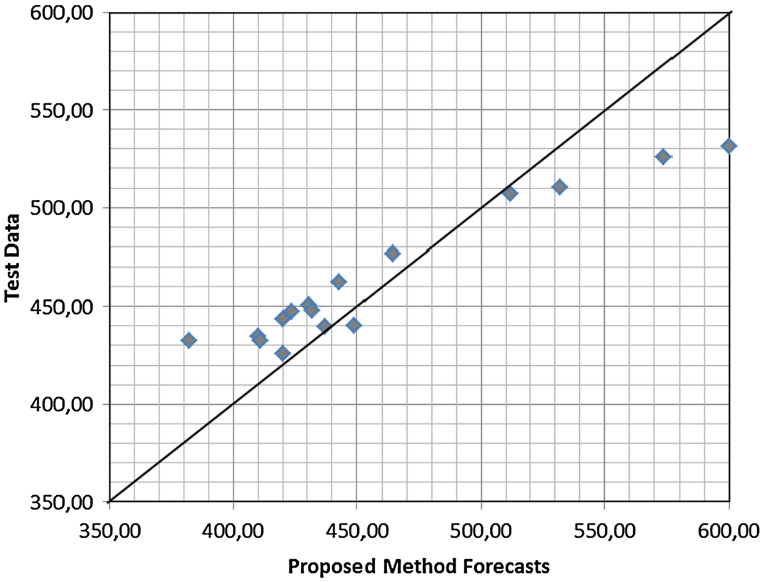

Fig. 8 Scatter plot of forecasts and observations for Case 2

Table 10 The training and testing MSE values are obtained from different architectures for MLP

	Number of hidden layer neurons			
	1	2	3	4
Training data				
Number of input layer neuron				
1	243,398.17	239,972.74	243,247.81	243,091.93
2	243,486.08	240,353.34	294,936.95	294,932.27
3	243,229.44	294,918.12	239,395.87	294,916.81
4	294,910.08	294,919.28	294,921.96	9,125.56
Test data				
Number of input layer neuron				
1	3,664.67	4,067.36	3,672.21	3,729.33
2	3,674.07	5,734.70	57,618.03	57,618.98
3	3,666.29	57,618.68	6,812.04	57,615.90
4	57,606.69	57,616.22	57,618.98	4,888.30

Table 11 The training and testing MSE values are obtained from different architectures for PSO-MLP

	Number of hidden layer neurons			
	1	2	3	4
Training data				
Number of input layer neuron				
1	297,256.62	245,106.56	245,624.49	245,484.50
2	247,286.18	246,935.36	291,246.30	246,935.29
3	251,810.09	249,197.56	248,186.12	255,839.58
4	304,138.28	304,039.26	304,138.28	250,437.88
Test data				
Number of input layer neuron				
1	57,618.98	3,664.29	3,713.44	3,670.78
2	3,785.01	3,667.91	49,556.32	3,667.35
3	6,839.52	4,399.50	3,662.55	11,077.94
4	57,618.98	57,618.98	57,618.98	3,664.67

obtained from MLP. Therefore, it is clear that MNM-MFF produces the most accurate forecasts for Australian beer consumption data. The forecasts of test set for MLP, PSO-MLP, and MNM-MFF are graphically shown in Fig. 5.

In addition to RMSE values and forecasting graph, scatter plot of forecasts obtained from the proposed method and observations is depicted in Fig. 6.

Case 2 In Case 1, it is seen that MNM-MFF has a superior forecasting performance. In the second case, performances of MLP,PSO-MLP, and MNM-MFF methods were compared to each other when the data include an outlier. An outlier value was obtained by multiplying the maximum observation value of the original data by 5. Then, 15th observation of the original data, which is the maximum observation value, was changed by this outlier. The obtained data set was forecasted using MLP, PSO-MLP, and MNM-MFF by using same settings in Case 1. All obtained results are shown in Tables 6, 7, 8, and 9.

According to Table 9, the mean error and standard deviation values obtained from MNM-MFF are less than those produced by MLP and PSO-MLP for both training and test sets. In Case 2, MNM-MFF produces more accurate and consistent forecasting results. In addition, if Tables 5 and 9 are compared, it is seen that MLP and PSO-MLP produce worse forecasting results when the data have an outlier. On the other hand, MNM-MFF gives almost same forecasting results even if the data include an outlier especially for test set. In other words, the proposed approach is not affected by outlier as much as MLP and PSO-MLP methods. The forecasts of test set for MLP,

Table 12 The training and testing MSE values are obtained from different architectures for MNM-MFF

	Number of hidden layer neurons			
	1	2	3	4
Training data				
Number of input layer neuron				
1	245,285.46	245,481.61	245,300.10	245,159.54
2	246,364.01	246,762.62	246,633.76	246,266.25
3	248,404.00	248,784.07	248,114.02	248,885.91
4	250,069.35	249,583.93	249,085.65	249,578.25
Test data				
Number of input layer neuron				
1	3,648.08	3,684.06	3,649.03	3,812.73
2	3,365.87	3,365.01	3,680.61	3,290.51
3	3,576.56	3,537.85	3,523.68	3,666.78
4	3,483.31	3,103.18	1,866.73	2,839.56

Table 13 The training and testing performances of MLP, MLP-PSO, and MNM-MFF for Case 3

	n	MLP	PSO-MLP	MNM-MFF
Training data				
Mean	16	250,609.70	264,603.90	247,484.90
SD	16	69,604.50	25,080.33	1,750.04
Test data				
Mean	16	27,701.40	20,740.32	3,380.05
SD	16	27,260.48	24,699.07	473.18

SD standard deviation

Table 14 Average values obtained from MLP, MLP-PSO, and MNM-MFF for all cases

	MLP	MLP-PSO	MNM-MFF
Case 1	5,954.83	4,318.32	3,514.95
Case 2	23,769.44	4,114.60	3,607.63
Case 3	27,701.40	20,740.32	3,380.05

PSO-MLP, and MNM-MFF are graphically shown in Fig. 7.

Also, scatter plot of forecasts obtained from the proposed method, and observations can be shown in Fig. 8.

Case 3 In the last case, an outlier value was obtained by multiplying the maximum observation value, which is 15th observation of the original data, by 10. Then, the observation which has the maximum value was changed by this outlier. MLP, PSO-MLP, and MNM-MFF methods were applied to the obtained data set by using same settings in Case 1. All obtained results are presented in Tables 10, 11, 12, and 13.

When Table 13 is examined, it is observed that MNM-MFF has the minimum mean error and standard deviation for both training and test sets. Thus, in Case 3, MNM-MFF again gives the most accurate and consistent forecasting results. In light of the results obtained in all cases, it can be clearly said that the proposed MNM-MFF model is not affected by outlier as much as MLP model. For both training and test sets, the results produced by MLP and PSO-MLP methods are inaccurate and inconsistent in Cases 2 and 3 where the data have an outlier. However, MNM-MFF is not affected by these extreme values as much as MLP and PSO-MLP. Besides, MNM-MFF produces very similar forecasting results especially for test set whether or not the data have an outlier.

In order to compare the forecasting performances of the methods MLP, PSO-MLP, and MNM-MFF better, averages for MSE values for test sets are also summarized in Table 14 for all cases. According to Table 14, it is clearly seen that MNM-MFF produces more accurate out of sample forecasts than those produced by MLP and PSO-MLP in all cases. In Cases 2 and 3 where the data have an outlier, the proposed MNM-MFF model gives accurate forecasts like in Case 1. On the other hand, MLP and PSO-MLP models cannot produce good results in Cases 2 and 3. This is an important finding indicates that MNM-MFF model is not affected by outlier as much as MLP and PSO-MLP models for Australian beer consumption data. The forecasts of test set for MLP, PSO-MLP, and MNM-MFF are graphically shown in Fig. 9.

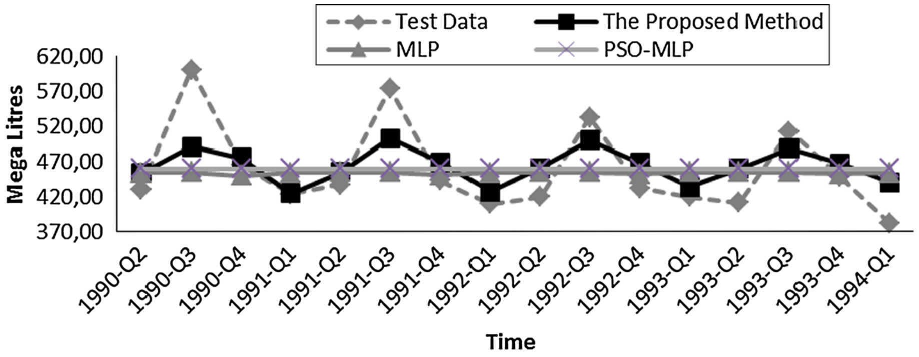

Fig. 9 The prediction results of MLP and proposed method for Case 3

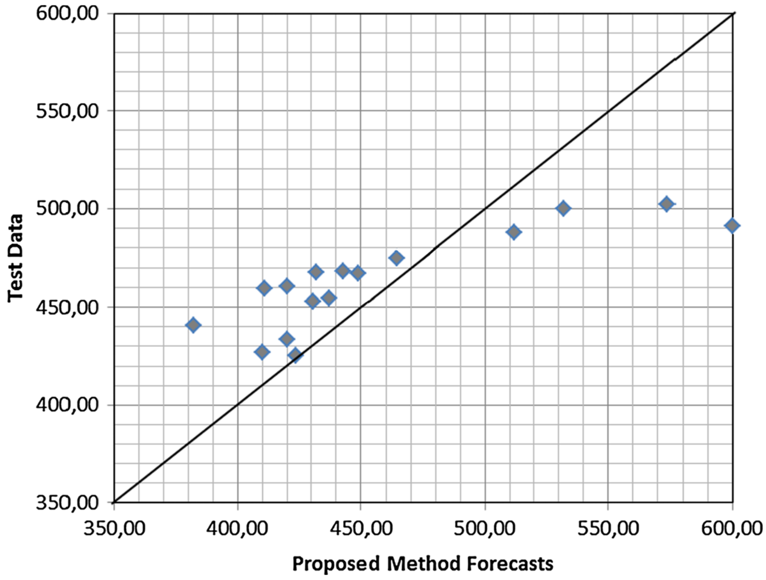

Fig. 10 Scatter plot of forecasts and observations for Case 3

Table 15 MSE values obtained from all methods for training and test sets

	BP-SMN	PSO-SMN	CRPSO-SMN	GA-SMN	MLP	PSO-MLP	MNM-MFF
Training	0.0016	0.0016	0.0016	0.0016	0.0082	0.0397	0.0005
Test	0.0018	0.0019	0.0018	0.0018	0.0226	0.0432	0.0015

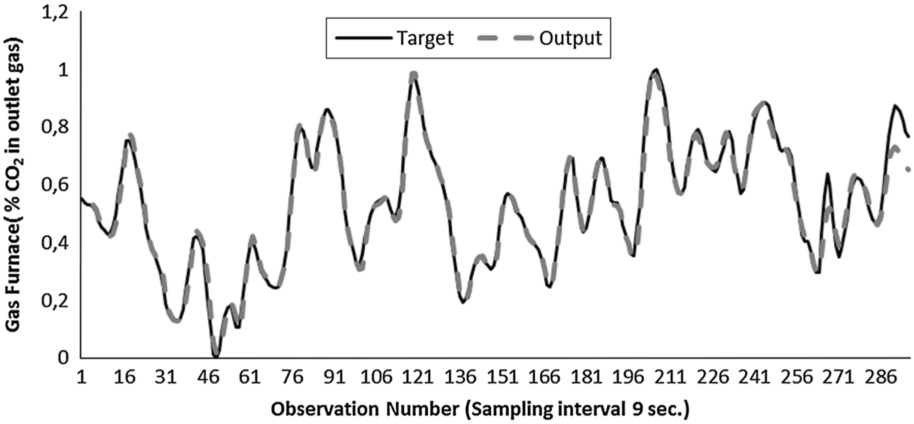

Fig. 11 The prediction results of MNM-MFF for the gas furnace data

Table 16 MSE values obtained from MLP, PSO-MLP, and MNM-MFF

	MLP	PSO-MLP	MNM-MFF
Train	0.0732	0.0818	0.0068
Test	0.0682	0.0130	0.0003

In addition, scatter plot of forecasts obtained from the proposed method, and observations are given in Fig. 10.

5.2 The gas furnace data

The second time series used in the implementation is Box-Jenkins gas furnace data set [7]. In gas furnace data, gas flow rate $x(t)$ and CO_2 concentration $y(t)$ are input and output, respectively. Because of the characteristic of this well-known data set, when artificial neural networks have been applied to the data, $x(t-4)$ and $y(t-1)$ have been taken as inputs, while $y(t)$ has been taken as output in all studies in the literature. The first 146 and the last 150 observations were used for training and test, respectively, like in other studies [24, 27]. To show the forecasting performance of the proposed MNM-MFF model, the data were forecasted using MNM-MFF and other artificial neural networks models available in the literature. When artificial neural networks have been utilized, the architecture 2-2-1 has been used in all these studies [24, 27]. That is, the architecture has 2, 2, and 1 neuron in input, hidden,

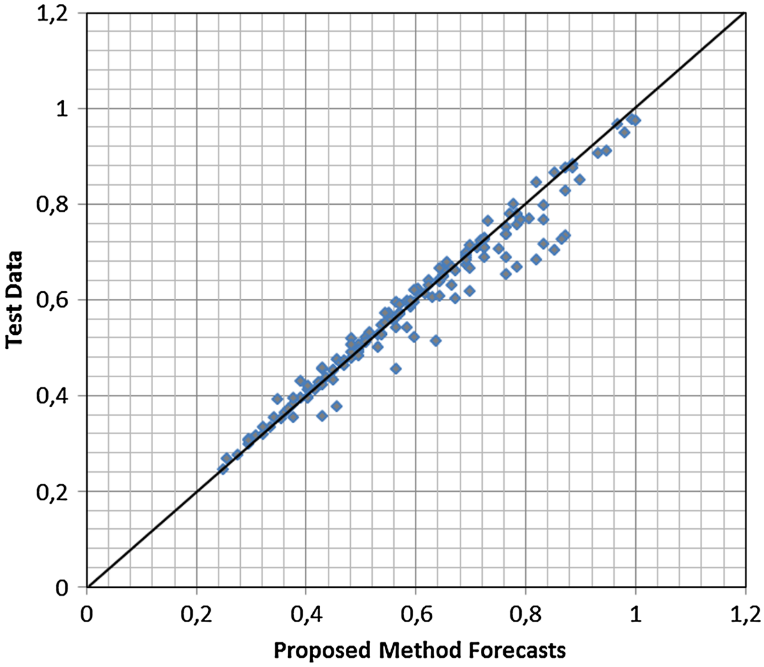

Fig. 12 Scatter plot of forecasts and observations for the gas furnace data

and output layers, respectively. Thus, the same architecture 2-2-1 was used for the proposed MNM-MFF model in the implementation. MSE values calculated over both the training and the test sets are summarized in Table 15. The results obtained from back propagation single multiplicative neuron model (BP-SMN) and MLP were taken from [24]. The other results produced by particle swarm optimization single multiplicative neuron model (PSO-SMN), cooperative random learning particle swarm optimization single multiplicative neuron model (CRPSO-SMN), and genetic algorithm single multiplicative neuron model (GA-SMN) were taken from [27].

When Table 15 is examined, it is clearly seen that the best forecasts for the test set were obtained when the proposed MNM-MFF model was employed. Also, the proposed model produced the best result for training set in terms of MSE. To examine the results visually, the graph of the observations (targets) and predictions (outputs) produced by MNM-MFF is given in Fig. 11. According to Fig. 11, it is obvious that the proposed MNM-MFF model gives very accurate results for gas furnace data. The agreement between the observations and the predictions of the proposed model is quite satisfactory.

In the first case, the gas furnace data were forecasted and it was seen that MNM-MFF has a superior forecasting performance. In the second case, performances of MLP, PSO-MLP, and MNM-MFF methods were compared to each other when the data include an outlier. An outlier value was obtained by multiplying the maximum observation value of the original data by 10. Then, 15th observation of the original data, which is the maximum observation value, was changed by this outlier. The obtained data set was forecasted using MLP and MNM-MFF by utilizing the same architecture in the first case. All obtained results are shown in Table 16. According to Table 16, the proposed MNM-MFF has the best accuracy for both training and test sets. This is also indicates that MNM-MFF model was not affected much by the outlier. However, the outlier leaded MLP model to misleading results.

In addition to RMSE values and forecasting graph, scatter plot of forecasts obtained from the proposed method and observations is depicted in Fig. 12.

6 Conclusions

In this study, a new neuron model which is called MNM is proposed. The proposed neuron model can produce an output which is not affected extreme values since it employs a median-based aggregation function. In addition, a new multilayer feed forward neural network (MNM-MFF) model that consists of MNM is firstly proposed in this study in order to reach high accuracy level and to cope with outlier problem. The proposed MNM-MFF model is a robust neural network model since its ability to deal with the outlier problem. Also, the modified particle swarm optimization method is used to train the proposed MNM-MFF model. In order to evaluate the performance of the proposed approach, it was applied to two real-time series. These time series were also forecasted using some other methods available in the literature for comparison. As a result of the comparison, it was clearly seen that the proposed approach produces very accurate forecasts for both Australian beer consumption and the gas furnace data sets. In addition to this, different data scenarios were considered in the implementation to examine the performance of the proposed approach when the data have outlier. Then, the forecasts obtained from the proposed approach were compared to those produced by MLP model, which has been the most preferred type of artificial neural networks in many implementations. It was shown that the proposed model is not affected much by outliers. To sum up, the proposed MNM-MFF model composed of MNM provides two important advantages. First, the proposed approach can produce very accurate forecasts. Secondly, it can be used to forecasts time series which include outliers. It should be noted that these results are obtained for the parameter sets given above, and two time series examined in the study. For instance, if the length of test set is shifted, the results can change or similarly if these parameter sets are used for other time series, the obtained results can change. Therefore, the obtained results are valid for only these parameter sets and these time series. In order to reach general results, a comprehensive simulation study has to be made. However, it is very hard to perform such a simulation study since there are many types of time series and many parameter combinations.

References

1. Aladag CH, Egrioglu E (2012) Advanced time series forecasting methods. In: Aladag CH, Egrioglu E (eds) Advances in time series forecasting. Bentham Science Publishers, Netherlands. eISBN: 978-1-60805-373-5. http://www.benthamscience.com/ebooks/9781608053735/preface.htm
2. Aladag CH, Kayabasi A, Gokceoglu C (2012) Estimation of pressuremeter modulus and limit pressure of clayey soils by various artificial neural network models. Neural Comput Appl (in press). doi:10.1007/s00521-012-0900-y
3. Aladag CH, Egrioglu E, Yolcu U, Dalar AZ (2012) A new time invariant fuzzy time series forecasting method based on particle swarm optimization. Appl Soft Comput 12:3291–3299
4. Aladag CH, Egrioglu E, Yolcu U (2010) Forecast combination by using artificial neural networks. Neural Process Lett 32(3): 269–276
5. Aladag CH, Egrioglu E, Kadilar C (2009) Forecasting nonlinear time series with a hybrid methodology. Appl Math Lett 22: 1467–1470
6. Basu M, Ho TK (1999) Learning behavior of single neuron classifiers on linearly separable or nonseparable inputs. In: IEEE LICNN'99
7. Box GEP, Jenkins GM, Reinse GC (1994) Time series analysis: forecasting and control. Prentice-Hall, Englewood Cliffs
8. Chaturvedi DK, Mohan M, Singh RK, Kalra PK (2004) Improved generalized neuron model for short-term load forecasting. Soft Comput 8(5):370–379
9. Giles CL, Maxwell T (1987) Learning invariance and generalization in higher order networks. Appl Opt 26:4972–4978
10. Hill T, Marquez L, O'Connor M, Remus W (1994) Artificial neural networks for forecasting and decision making. Int J Forecast 10:5–15
11. Homma N, Gupta MM (2002) Generalized second-order neural unit. Bull Coll Med Sci Tohoku Univ 11(1):1–6
12. Janacek G (2001) Practical time series. Oxford University Press Inc., New York, p 156
13. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, Piscataway, NJ, USA. IEEE Press, pp 1942–1948
14. Labib R (1999) New single neuron structure for solving nonlinear problems. In: IEEE IJCNN'99, pp 617–620
15. 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
16. McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 5:115–133
17. Plate TA (2000) Randomly connected sigma–pi neurons can form associator networks. NETCNS Netw Comput Neural Syst 11:321–322
18. Rumelhart E, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation, chapter 8. The M.I.T. Press, Cambridge, pp 318–362
19. Shi Y, Eberhart RC (1999) Empirical study of particle swarm optimization. Proc IEEE Int Congr Evol Comput 3:101–106
20. Shiblee M, Chandra B, Kalra PK (2010) Learning of geometric mean neuron model using resilient propagation algorithm. Expert Syst Appl 37:7449–7455
21. Sinha M, Kumar K, Kalra PK (2000) Some new neural network architectures with improved learning schemes. Soft Comput 4(4):214–223
22. Taylor JG, Commbes S (1993) Learning higher order correlations. Neural Netw 6(3):423–428
23. Yadav RN, Kumar N, Kalra PK, John J (2006) Learning with generalized-mean neuron model. Neurocomputing 69(16–18): 2026–2032
24. Yadav RN, Kalra PK, John J (2007) Time series prediction with single multiplicative neuron model. Appl Soft Comput 7:1157–1163
25. Zhang G, Patuwo BY, Hu M (1998) Forecasting with artificial neural networks: the state of the art. Int J Forecast 14(1):35–62
26. Zhang CN, Zhao M, Wang M (2000) Logic operations based on single neuron rational model. IEEE Trans Neural Netw 11:739–747
27. Zhao L, Yang Y (2009) PSO-based single multiplicative neuron model for time series prediction. Expert Syst Appl 36:2805–2812