American Journal of Intelligent Systems 2014, 4(1): 15-19
DOI: 10.5923/j.ajis.20140401.03

An Investigation of Differencing Effect in Multiplicative Neuron Model Artificial Neural Network for Istanbul Stock Exchange Time Series Forecasting

Ali Zafer Dalar[1], Erol Eğrioğlu[1,*], Ufuk Yolcu[2], Damla İlter[3], Özge Gündoğdu[4]

[1]Department of Statistics, Ondokuz Mayis University, Samsun, 55139, Turkey
[2]Department of Statistics, Ankara University, Ankara, 06100, Turkey
[3]Department of Statistics, Mimar Sinan Fine Arts University, Istanbul, 34349, Turkey
[4]Department of Econometrics, Cumhuriyet University, Sivas, 58140, Turkey

Abstract In recent years, good alternative methods have been proposed to obtain forecasts for a time series. Artificial neural networks have been commonly used for forecasting purpose in the literature. Although, multilayer perceptron artificial neural network is the most used artificial neural network type, multiplicative neuron model artificial neural networks have been used to obtain forecasts for six years. In the literature, many studies used original series without applying any differencing operation. Thereby, non-stationary time series were used in the artificial neural networks. It is very difficult to find appropriate time series model for non-stationary time series in probabilistic time series methods. Similarly, differencing can be useful obtaining forecasts by artificial neural networks. Differencing effect has not been sufficiently discussed for artificial neural networks. It has not been discussed for multiplicative neuron model artificial network in the literature, yet. Aim of this study is discussing of differencing effect for multiplicative neuron model artificial neural networks. Istanbul stock exchange data (IEX) sets were used to explore differencing effect. The data setsare made up of five time series for years between 2009 and 2013. All of time series were daily observed and observations of them are taken for first five months. It is shown that differencing operation is not useful for forecasting IEX as a result of statistical hypothesis tests.

Keywords Artificial Neural Network, Difference Operator, Forecasting, Multiplicative Neuron Model, Particle Swarm Optimization, Artificial Bee Colony Optimization

1. Introduction

Forecasting is the process of making statements about events whose actual outcomes have not yet been observed. Forecasting methods can be classified into two classes as probabilistic and non-probabilistic methods. Artificial neural networks are non-probabilistic methods. Because artificial neural networks do not need strict assumptions such as normality, linearity, they have been commonly used in the literature in recent years.

Zhang et al. (1998) and Hippert et al. (2001) are good surveys of literature about neural network forecasting methods. In the literature, many of papers asserted that neural networks are outperforms traditional forecasting methods like autoregressive moving average (ARMA) models. Multilayer perceptron neural networks have been commonly used for obtaining forecasts. Alpaslan et al.(2012) statistically investigated some important factors to obtain more accurate forecasts by using multilayer perceptron neural networks.Aladag et al. (2012), Cagcag (2013) and Oner et al. (2013) proposed hybrid methods which are based on multilayer perceptron neural networks. In recent years, different kinds of artificial neural network models have been proposed for forecasting. Yadav et al. (2007) introduced multiplicative neuron model ANN (MNM-ANN) which has only one neuron in the hidden layer. Because MNM-ANN has one neuron, determining number of hidden layer neurons is not needed. This is very important, because determining number of hidden layer neurons is important problem for multilayer perceptron. Zhaou andYang (2009) and Samanta (2011) used particle swarm optimization method to train MNM-ANN. Particle swarm optimization has been used different aim in Itamiya et al. (2013). There are two modifications of multiplicative neuron model artificial network in literature. Yolcu et al. (2013) and Aladag et al. (2013) proposed different artificial neural networks which are employed multiplicative neuron model.Egrioglu et al. (2013) used multiplicative neuron model in a fuzzy time series forecasting algorithm.

Stationarity of time series is discussable assumption in the practical application of ANN for forecasting. In the literature,

* Corresponding author:
erole@omu.edu.tr (Erol Eğrioğlu)
Published online at http://journal.sapub.org/ajis

many authors don't take into account stationarity of time series. There are some papers in which stationary assumption are investigated. It is shown that first order differencing is useful for forecasting with ANN in Chow and Leung (1996). Kim et al. (2004) investigated whether it is feasible to relax the stationarity condition to non-stationary time series. Ghazali et al. (2011) put forward that ANN can produce accurate forecasts for non-stationary time series. Kandananond (2013) found that multilayer perceptron ANN is better than support vector machine ANN for stationary time series. Stationary assumption has not been investigated for MNM-ANN in the literature, yet. In this paper, differencing effect is investigated for MNM-ANN by using statistical hypothesis tests.

In the second section, MNM-ANN and its training algorithm are briefly given. Experimental study is summarized in section three. In the fourth section, conclusions are given and they are discussed.

2. MNM-ANN

Yadav et al. (2007) was firstly proposed MNM-ANN. The architecture of MNM-ANN is given Fig. 1. $x_1, x_2, \ldots, x_m$ are inputs of MNM-ANN.

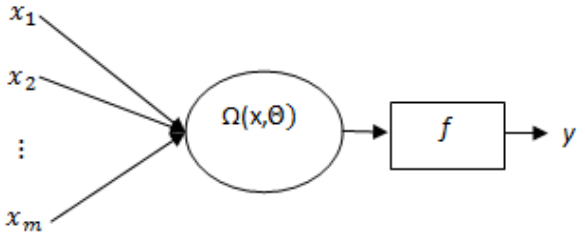

Figure 1. Architecture of MNM-ANN

$\Omega(x, \Theta)$ is aggregation function and it has multiplicative structure. In MNM-ANN architecture, there is only one neuron and its output is calculated as below:

$$\text{net} = \Omega(x, \Theta) = \prod_{i=1}^{m}(x_i w_i + b_i) \quad (1)$$

where $\Theta = (w_1, w_2, \cdots, w_m, b_1, b_2, \cdots, b_m)$ and w_i, b_i , $(i = 1,2, \cdots, m)$ are weight and bias for i^{th} input, respectively. The activation function (f) was selected as logistic activation function in Yadav et al. (2007). The logistic activation function can be given as follow:

$$f(net) = \frac{1}{1+e^{-net}} \quad (2)$$

The output of MNM-ANN can be calculated as $y = f(net)$. Although Yadav et al. (2007) utilized back propagation algorithm for training MNM-ANN, it can be performed by using particle swarm optimization. Zhaou and Yang (2009), Samanta (2011) Yolcu et al. (2013) and Aladag et al. (2013) used particle swarm optimization to train multiplicative neuron based neural network. Alpaslan et al. (2014) used artificial bee colony algorithm to train MNM-ANN.

Particle swarm optimization (PSO) was firstly proposed in Kennedy and Eberhart (1995). The PSO algorithm for training MNM-ANN is given below.

Algorithm 1. PSO algorithm used to train the proposed MNM-ANN model

Step 1. Positionsand velocities of each m^{th} (m = 1,2, …, *pn*) particles are randomly determined and kept in vectors P_m and V_m given as follows:

$$P_m = \{p_{m,1}, p_{m,2}, \ldots, p_{m,d}\}, \; m = 1,2, \ldots, pn \quad (3)$$

$$V_m = \{v_{m,1}, v_{m,2}, \ldots, v_{m,d}\}, \; m = 1,2, \ldots, pn \quad (4)$$

where $x_{m,j}$ (i=1,2,…,d) represents j^{th} position of m^{th} particle. *pn* and *d* represents the number of particles in a swarm and positions, respectively. The initial positions and velocities of each particle in a swarm are randomly generated from uniform distribution (0,1) and (-*vm*,*vm*), respectively. Positions of a particle are consisted from weights and biases. Each particle gives a solution set for the neural network.

Step 2. The parameters of PSO are determined.

In the first step, the parameters which direct the PSO algorithm are determined. These parameters are *pn*, *vm*,c_{1i}, c_{1f}, c_{2i}, c_{2f}, w_1, and w_2. Let c_1 and c_2 represents 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).

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

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

$$w = (w_2 - w_1)\frac{maxt - t}{maxt} + w_1 \quad (7)$$

Step 3. Evaluation function values are computed. Evaluation function values for each particle are calculated. MSE given in below is used as evaluation function.

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(desired_t - output_t)^2 \quad (8)$$

where *n* represents the number of learning sample. The output value of the proposed model is calculated by algorithm 1.

Step 4. $Pbest_m$ (m = 1,2, …, *pn*) and *Gbest* are determined due to evaluation function values calculated in the previous step. $Pbest_m$ is a vector stores the positions corresponding to the m^{th} particle's best individual performance, and *Gbest* is the best particle, which has the best evaluation function value, found so far.

$$Pbest_m = (pb_{m,1}, pb_{m,2}, \ldots, pb_{m,d}), \; m = 1,2, \ldots, pn \quad (9)$$

$$Gbest = (p_{g,1}, p_{g,2}, \ldots, p_{g,d}) \quad (10)$$

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 (11) and (12). If maximum iteration number is reached, the algorithm goes to Step 3; otherwise, it goes to Step 7.

$$v_{m,j}^{t+1} = \left[w \times v_{m,j}^{t} + c_1 \times rand_1 \times \left(pb_{m,j} - p_{m,j}\right) + c_2 \times rand2 \times pg,j - pm,j\right] \quad (11)$$

$$p_{m,j}^{t+1} = p_{m,j}^{t} + v_{m,j}^{t+1} \quad (12)$$

where $= 1,2,\ldots,pn \;, j = 1,2,\ldots,d.$

Step 7. The best solution is determined. The elements of Gbest are taken as the best weight values of the new ANN model.

Artificial bee colony (ABC) algorithm was firstly introduced by Karaboga (2005). Karaboga et al. (2007) utilized ABC algorithm to train feed forward neural network. The ABC algorithm for training MNM-ANN is given below.

Algorithm 2. ABCAlgorithm for training MNM-ANN

Step 1. The number of food sources (SN) andlimitvalue are determined.

Step 2. Initial food source locations are randomly generated from $(z_{min}^{j}, z_{max}^{j})$ interval.

Step 3. Fitness function values are calculated for each food source. Fitness function is MSE value that is calculated by using locations of the source. The locations of source can be used as weights and biases for MNM-ANN.

Step 4. Sending employed bees to the food source locations.

New food source v_i is obtained by using (13). To calculate v_{ij} location, a neighbor source (k^{th}) is randomly selected. The j^{th} location of the new source is obtained from (13). Other locations of the new source are taken by i^{th}source.

$$v_{ij} = z_{ij} + \emptyset_{ij}\left(z_{ij} - z_{kj}\right) \quad (13)$$

The fitness function value is calculated for the new source. If the fitness value of the new source is bigger than the fitness value of i^{th} source, failure counter of this source is increased by one. Otherwise, the new source is taken as i^{th} source and the failure counter is set to zero.

Step 5. Onlooker bee stage is applied. The probability values are calculated by using (14).

$$p_i = \frac{1/f_i}{\sum_{j=1}^{SN} 1/f_j} \quad (14)$$

Onlooker bees are sent to the sources according to their probabilities. If any source have high probability, this source will be visited more by the onlooker bees.

Step 6. The best food source is determined and saved.

Step 7. All food sources are checked and exhausted sources are determined. If failure counter is bigger than the limit value for a source, this source can be considered as exhausted source. For each exhausted source, a scout bee is employed. The locations of new source are randomly generated from $(z_{min}^{j}, z_{max}^{j})$ interval instead of the exhausted source. The failure counter is set to zero for the new source.

Step 8. Stopping conditions are checked. If the stopping conditions are met, skip to step 9. Otherwise, back to step 4.

Step 9. The best food source is taken as the solution.

3. Experimental Study

In the experimental study, IEX data sets were used to test differencing effect. Details of data sets are given below:

Set 1. BIST 100 data for IEX, it is daily observed between 02/01/2009 and 29/05/2009 dates.

Set 2. BIST 100 data for IEX, it is daily observed between 04/01/2010 and 31/05/2010 dates.

Set 3. BIST 100 data for IEX, it is daily observed between 03/01/2011 and 31/05/2011 dates.

Set 4. BIST 100 data for IEX, it is daily observed between 02/01/2012 and 31/05/2012 dates.

Set 5. BIST 100 data for IEX, it is daily observed between 02/01/2013 and 29/05/2013 dates.

Firstly, Augmented Dickey Fuller (ADF) unit root test is applied for five series by using Eviews Package program. According to test results, all series have unit roots and first differences of all series are stationary time series. In the experimental study, all series and their first differences are solved by using MNM-ANN. In the application, three different test sets were used for all series. The data of test sets are obtained %10, %20 and %30 of all observations for all series. PSO and ABC algorithms were used to train MNM-ANN. Input numbers are taken 2,3,4 and 5. In the experimental design, all factors are listed below:

Factor 1 is differencing operation. This factor has two levels which are differenced (1) and original series (2).

Factor 2 is test set length. This factor has three levels. These levels are %10 (1), %20 (2) and %30 (3).

Factor 3 is input numbers of MNM-ANN. Levels of the factor are 2 (1), 3 (2), 4 (3) and 5(4). There are four levels for this factor.

Factor 4 is years. This factor has five levels which are 2009 (1), 2010 (2), 2011 (3), 2012 (4) and 2013 (5).

Factor 5 is training algorithm. Levels are PSO (1) and ABC (2).

Observations of dependent variable are mean absolute percentage error (MAPE) values which are obtained for test data sets in each run. MAPE can be calculated by using (15).

$$\boldsymbol{MAPE} = \sum_{t=1}^{n} \left| \frac{\boldsymbol{desired_t - output_t}}{\boldsymbol{desired_t}} \right| \quad (15)$$

In SPSS package program, univariate general linear model section was used to obtain statistical test results. The test results are given in Table 1.

According to Table 1, Differences between levels of factor 5 and factor 3 are not statistically significant. Differences among the levels of Factor 1,2 and 4 are statistically significant.

Figure 2 is given to understand differences between the levels of Factor 1,2 and 4. According to Figure 2, the bigger MAPE values for test sets were obtained from differenced series. For all years, the lagged variables of original series can be used as inputs. When the test set length is taken bigger, the mean MAPE values are increased. The obtained test results for 2009 year are worse than results of other years.

Table 1. Univariate general linear model test results

Dependent Variable: MAPE

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	,469[a]	9	,052	85,695	,000
Intercept	,072	1	,072	118,284	,000
Factor 2	,064	2	,032	53,021	,000
Factor 5	,000	1	,000	,415	,520
Factor 1	,095	1	,095	155,570	,000
Factor 4	,309	4	,077	127,307	,000
Factor 3	8,267E-008	1	8,267E-008	,000	,991
Error	,140	230	,001		
Total	1,383	240			
Corrected Total	,608	239			

a. R Squared = ,770 (Adjusted R Squared = ,761)

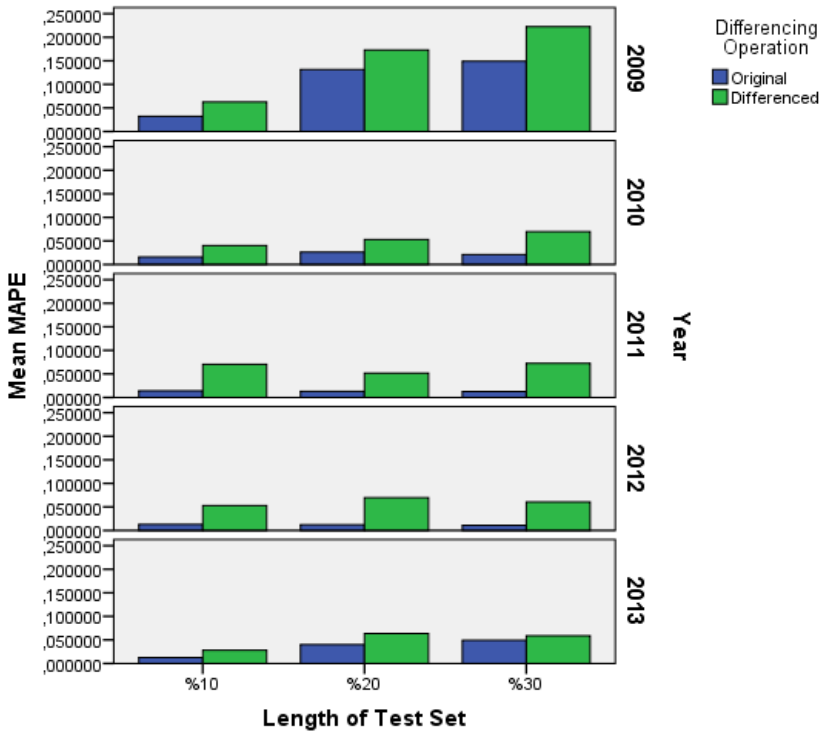

Figure 2. Mean of MAPE values for the levels of Factor 1,2 and 4

4. Conclusions

In this study, differencing effect for MLP-ANN are investigated. An experimental study was designed to test differencing effect. The obtained results are showed that there is no need to difference in MLP-ANN. MLP-ANN can give better results for original and non-stationary time series. These conclusions are obtained for IEX data. It is clear that the different results can be obtained for different data sets. According to obtained results, it can be asserted that IEX time series can be solved with MLP-ANN by using their original data. Thus, stationary is not needed and it is not strict assumption of MLP-ANN for IEX time series. In the future studies, the similar experimental study can be applied for exchange data sets of other countries. The results may generalize for stock exchange data sets.

REFERENCES

[1] G. Zhang, B.E. Patuwo, and Y.M. Hu, "Forecasting with artificial neural networks: The state of the art," International Journal of Forecasting, vol. 14, pp. 35-62, 1998.

[2] H.S. Hippert, C.E. Pedreira, and R.C. Souza, "Neural Networks for Short Term Load Forecasting: A Review and Evaluation," IEEE Transaction on Power Systems, vol. 16(1), pp. 44-55, 2001.

[3] F. Alpaslan, E. Egrioglu, C.H. Aladag andE. Tiring, An Statistical Investigation On Feed Forward Neural Networks

For Forecasting Time Series, American Journal of Intelligence Systems, vol. 2(3), pp. 21-25, 2012.

[4] C.H. Aladag, E. Egrioglu and C. Kadilar, "Improvement in forecasting accuracy using the hybrid model of ARFIMA and Feed Forward neural network," American Journal of Intelligence Systems, vol. 2(2), pp. 12-17, 2012.

[5] O. Cagcag, U. Yolcu, E. Egrioglu, and C.H. Aladag, "A Novel Seasonal Fuzzy Time Series Method to the Forecasting of Air Pollution Data in Ankara," American Journal of Intelligent Systems, vol. 3(1), pp. 13-19, 2013.

[6] Y. Oner, T. Tunc, E. Egrioglu, and Y. Atasoy, "Comparisons of Logistic Regression and Artificial Neural Networks in Lung Cancer Data," American Journal of Intelligent Systems vol. 3(2), pp. 71-74, 2013.

[7] R.N. Yadav, P.K. Kalra, and J. John, "Time series prediction with single multiplicative neuron model," Applied Soft Computing, vol. 7, pp. 1157-1163, 2007.

[8] L.Zhao, Y.Yang, "PSO-based single multiplicative neuron model for time series prediction," Expert Systems with Applications, vol. 36, pp. 2805-2812, 2009.

[9] B. Samanta, "Prediction of chaotic time series using computational intelligence," Expert Systems with Applications, vol. 38(9), pp. 11406-11411, 2011.

[10] K. Itamiya, M. Sawada, D. Kikuta, and K. Tod, "A Hybrid Measurement System of Three Dimensional Coordinates by Combination of a Multi-link Manipulator and Particle Swarm Optimization Techniques," American Journal of Intelligent Systems, vol. 3(2), pp. 51-56, 2013.

[11] U.Yolcu, C.H.Aladag, and E.,"A New Linear & Nonlinear Artificial Neural Network Model for Time Series Forecasting," Decision Support System, vol. 54, pp. 1340-1347, 2013.

[12] C.H.Aladag, U.Yolcu, and E. Egrioglu,"A new multiplicative seasonal neural network model based on particle swarm optimization," Neural Processing Letters, vol. 37(3), pp. 251-262, 2013.

[13] E. Egrioglu, C.H. Aladag, U. Yolcu, B.S. Corba, and O. Cagcag, "Fuzzy Time Series Method Based On Multiplicative Neuron Model and Membership Values," American Journal of Intelligent Systems," vol. 3(1), pp. 33-39, 2013.

[14] T.W.S. Chow and C.T. Leung, "Neural Network Based Short Term Load Forecasting Using Weather Compensation," IEEE Transaction on Power Systems, vol. 11(4), pp. 1736-1742, 1996.

[15] T.Y. Kim, K.J. Oh, C. Kim, and J. D. Do, "Artificial Neural Networks for Non-Stationary Time Series," Neurocomputing, vol. 61, pp. 439-447, 2004.

[16] R. Ghazali, A.J. Hussain, and P. Liatsis, "Dynamic Ridge Polynomial Neural Network: Forecasting the univariate non-stationary and stationary trading signals," Expert Systemswith Applications, vol. 38, pp. 3765-3776, 2011.

[17] K. Kandananond, "Applying 2^k Factorial Design to assess the performance of ANN and SVM for Forecasting Stationary and Non-stationary Time Series, Procedia Computer Science, vol. 22, pp.60-69, 2013.

[18] Alpaslan F., Egrioglu E., Aladag C.H., Ilter D., and Dalar A.Z., "Comparison of Single Multiplicative Neuron Artificial Neural Network Models Using ABC and BP Training Algorithms, Anadolu University Journal of Science and Technology, Accepted Paper, 2014.

[19] J. Kennedy, and R. Eberhart, "Particle swarm optimization, In Proceedings of IEEE International Conference on Neural Networks,", pp. 1942–1948, Piscataway, NJ, USA, IEEE Press, 1995.

[20] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[21] D. Karaboga, B. Akay, C. Ozturk, "Artificial Bee Colony (ABC) Optimization Algorithm for Training Feed-Forward Neural Networks," LNCS: Modeling Decisions for Artificial Intelligence, vol. 4617, pp. 318-329, 2007.