Hacettepe Journal of Mathematics and Statistics
Volume 37 (2) (2008), 185 – 200

# A NEW ARCHITECTURE SELECTION STRATEGY IN SOLVING SEASONAL AUTOREGRESSIVE TIME SERIES BY ARTIFICIAL NEURAL NETWORKS

Cagdas Hakan Aladag*, Erol Egrioglu† and Suleyman Gunay*

Received 07 : 08 : 2008 : Accepted 31 : 10 : 2008

**Abstract**

The only suggestions given in the literature for determining the architecture of neural networks are based on observations, and a simulation study to determine the architecture has not yet been reported. Based on the results of the simulation study described in this paper, a new architecture selection strategy is proposed and shown to work well. It is noted that although in some studies the period of a seasonal time series has been taken as the number of inputs of the neural network model, it is found in this study that the period of a seasonal time series is not a parameter in determining the number of inputs.

**Keywords:** Architecture selection, Seasonal autoregressive time series, Neural networks, Forecasting, Simulation.

*2000 AMS Classification:* 62–04, 62 M 10, 82 C 32, 90 C 59.

## 1. Introduction

There are many studies in the literature that use artificial neural networks (ANN) to analyze time series. However, a simulation study does not seem to have been used to determine the architecture of an ANN. Due to the lack of theoretical knowledge about determining the architecture, empirical results are widely employed by researchers. Tang and Fishwick [13] suggested that the number of inputs can be taken as the number of terms of an autoregressive (AR). Lachtermacher and Fuller [8] claimed that when one neuron is used in the output layer, a number of inputs greater than one could affect the results in a negative way. Sharda and Patil [11] and Tang *et al.* [12] took heuristically

*Department of Statistics, Hacettepe University, Ankara 06800, Turkey.
E-mail : (C. H. Aladag) aladag@hacettepe.edu.tr (S. Gunay) sgunay@hacettepe.edu.tr
†Department of Statistics, Ondokuz Mayis University, Samsun 55139, Turkey.
E-mail : erole@omu.edu.tr

the period of the time series as the number inputs in the analysis of seasonal time series. Buhamra *et al.* [2] suggested that the number of inputs should be determined according to the Box-Jenkins approach. The most often applied method in determining the number of neurons in the hidden layer is the trial and error method given by Zhang [15]. In addition, when the number of inputs is $n$, Lippmann [9] and Hecth-Nielsen [5] took $2n + 1$ as the number of neuron in the hidden layer, Wong [14], Tang and Fishwick [13], and Kang [7] took it to be $2n$, $n$ and $n/2$, respectively.

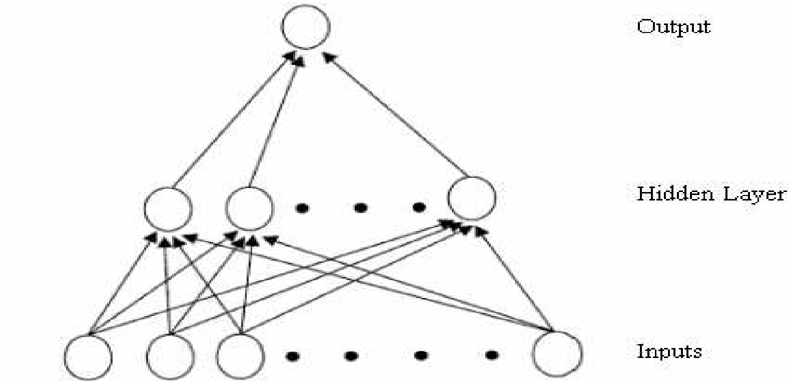
In this study, a simulation study has been performed to determine an architecture for analyzing autoregressive time series that has different periods when an output and a hidden layer are used. A new selection strategy is proposed to decide the number of neurons in the input layer and in the hidden layer. In contrast to the trial and error method by Zhang [15], the proposed method saves time and has an accepted error level without trying all combinations of architecture. In addition, the error that can be caused by the random selection of architectures can be avoided. Section 2 gives some brief information about ANN. Section 3 includes the detailed information about the simulation study and the results. The new selection strategy is introduced in Section 4. Section 5 consists of a discussion of new method and results.

## 2. Artificial Neural Networks

"What is an artificial neural network?" is the first question that should be answered. Picton [10] answered this question by separating it into two parts. The first part is why it is called an artificial neural network. It is called an artificial neural network because it is a network of interconnected elements. These elements were inspired by studies of biological nervous systems. In other words, artificial neural networks are an attempt at creating machines that work in a similar way to the human brain by building these machines using components that behave like biological neurons. The second part is what an artificial neural network does. The function of an artificial neural network is to produce an output pattern when presented with an input pattern. In forecasting, artificial neural networks are mathematical models that imitate biological neural networks. Artificial neural networks consist of some elements. Determining the elements of the artificial neural networks is an issue that affects the forecasting performance of the network, and should be considered carefully. The elements of an artificial neural networks are generally given as the network architecture, the learning algorithm and the activation function [4]. One critical decision is to determine the appropriate architecture, that is, the number of layers, number of nodes in each layers and the number of arcs which interconnects with the nodes [18]. However, in the literature, there are no general rules for determining the best architecture. Therefore, many architecture have to be tried to get the correct results. There are various types of artificial neural network. One of these is known as the *feed forward* neural network. Feed forward neural networks have been used successfully in many studies [4]. In feed forward neural networks, there are no feedback connections. The *broad feed forward neural network architecture* that has single hidden layer and single output is illustrated on the next page.

Determining the *learning algorithm* of an artificial neural network for a specific task is equivalent to finding the values of all weights such that the desired output is generated by the corresponding input. Various training algorithms have been used for the determination of the optimal weights values. The most popularly used training method is the back propagation algorithm. In the back propagation algorithm, learning consists of adjusting all weights according to the measure of error between the desired output and actual output [4].

**A broad feed forward neural network architecture**



Another element of an artificial neural network is the activation function. This determines the relationship between inputs and outputs of the network. In general, the activation function introduces a degree of the non-linearity that is valuable in most artificial neural network applications. The well known activation functions are the logistic, hyperbolic tangent, sine (or cosine) and the linear functions. Among these, the logistic activation function is the most popular [15].

## 3. The simulation study

The computer code, called NN-Back Propagation, given in [1], is employed in the simulation study. By using the expression below,

$$(1) \qquad Z_t = \phi\, Z_{t-s} + a_t$$

ten time series, each with a length of 100 were generated with parameters $s = 4, 6, 12$ and $\phi = 0.5$ for the first order autoregressive model (SAR(1)) where $s$ and $\phi$ represent the period and autoregressive parameter respectively. Thus, the total number of time series was 30. In the literature, a small number of time series are employed for simulation studies in order to complete them in an acceptable time frame when ANN is the method used to analyze the time series. For example, Zhang *et al.* [16] used a total number of 8 series, and a similar study was conducted with 5 time series by Hwrang [6]. In this study, the choice of a total number of 30 times series is aimed at obtaining more reliable results. Each of the time series generated are analyzed with ANN. Then 95 percent of the whole data was taken as training data, and the remaining 5 percentused as test data. The components of the ANN used in the simulation study are expressed below.

**Architecture structure:** For each case examined, the number of neurons in the input layer varied from 1 through 12, the number of neurons in the hidden layer likewise varies from 1 through 12, and there was one neuron in the output layer so a total of 144 architectures were used for each time series. Hence a total of 4320 different architectures were examined for the 30 time series generated. The feed forward neural network architecture structure, which includes a direct link between the neurons in the input layer and the output neuron, was used.

**The learning algorithm:** The Back Propagation Algorithm, in which the learning parameter is updated at each iteration, was used to find the best values of the weights.

**The activation function:** The logistic function given below was used as an activation function.

$$(2) \qquad f(x) = (1 + \exp(-\gamma\, x))^{-1}$$

In the literature, various performance measures are used to gauge the forecasting accuracy of the neural network model in analyzing time series [3]. In order to measure the performance of the implemented neural networks model, the forecasting root mean square error (FRMSE) was used.

The results obtained from the time series with period 4 are given in Figures 1–3. In Figure 1, the vertical axis represents the mean values of FRMSEs and their standard deviations, while the horizontal axis represents the number of neurons in the hidden layer (NNHL) for the 10 time series. In all of the graphs, while the diamonds linked to each other represent the mean values of the FRMSEs, the linked squares represent the standard deviations. In Figure 1 the mean value of each FRMSE is obtained using the 12 architectures obtained by keeping the hidden numbers fixed while the number of neurons in the input layer vary from 1 through 12. For example, in the graph for series 1, utilizing 12 different architectures constructed by changing the number of inputs from 1 through 12, produced the approximated mean value 15 and the approximated standard deviation value 5 for all the FRMSEs when the number of neurons in the hidden layer was 4. In Figure 2, obtained by keeping the number of neurons in the input layer (NNIL) fixed, the mean and standard deviation values of the FRMSEs were obtained by changing the number of neurons in the hidden layer from 1 through 12.

In Figure 3, the mean values of the FRMSEs and the mean value of the standard deviations of the FRMSEs for the 10 generated time series whose period is 4 are given. The horizontal axis represents the number of neurons in the hidden layer, and the vertical axis the mean values. In figure 4, the mean values for the 10 generated time series are shown. The vertical axis represents the mean values and the horizontal axis the number of neurons in the input layer.

For the time series with period 6, similar graphs are given in Figures 5 through 8. Likewise, Figures 9 through 12 are the graphs for series with period 12.

For the 30 time series studied in the simulation study, the architectures which produce the lowest FRMSE values are shown in Table 1. For example, for the first time series whose period is 4, the best architecture which has the lowest FRMSE value is 6–12, which represents the architecture with 6 input neurons and 12 neurons in the hidden layer. We recall that just one output neuron is used in all of the architectures.

**Table 1. The best ANN architectures and their FRMSE values for the 30 time series generated**

| | | | Period | | | | | Period | |
|---|---|---|---|---|---|---|---|---|---|
| Series | | 4 | 6 | 12 | Series | | 4 | 6 | 12 |
| 1 | FRMSE | 7.9636 | 5.7070 | 14.4032 | 6 | FRMSE | 4.1291 | 7.3017 | 11.9521 |
| | Arch. | 6 - 12 | 12 - 3 | 6 - 1 | | Arch. | 1 - 10 | 7 - 5 | 8 - 9 |
| 2 | FRMSE | 7.1380 | 5.1593 | 9.7384 | 7 | FRMSE | 5.3627 | 4.5014 | 4.7954 |
| | Arch. | 9 - 12 | 1 - 1 | 9 - 11 | | Arch. | 4 - 5 | 6 - 5 | 7 - 1 |
| 3 | FRMSE | 6.0275 | 4.8553 | 9.8878 | 8 | FRMSE | 7.6142 | 4.3405 | 7.0688 |
| | Arch. | 10 - 6 | 1 - 1 | 2 - 1 | | Arch. | 10 - 11 | 6 - 1 | 5 - 11 |
| 4 | FRMSE | 6.0177 | 5.9587 | 6.3203 | 9 | FRMSE | 10.6363 | 7.3677 | 7.7616 |
| | Arch. | 7 - 2 | 10 - 7 | 4 - 1 | | Arch. | 7 - 12 | 9 - 11 | 6 - 2 |
| 5 | FRMSE | 2.8569 | 6.4381 | 7.7698 | 10 | FRMSE | 7.2309 | 6.1912 | 6.3891 |
| | Arch. | 9 - 6 | 7 - 8 | 3 - 12 | | Arch. | 11 - 10 | 9 - 5 | 8 - 8 |

**Figure 1. The graphs of calculated values based on the number of neurons in the hidden layer for time series with period 4**

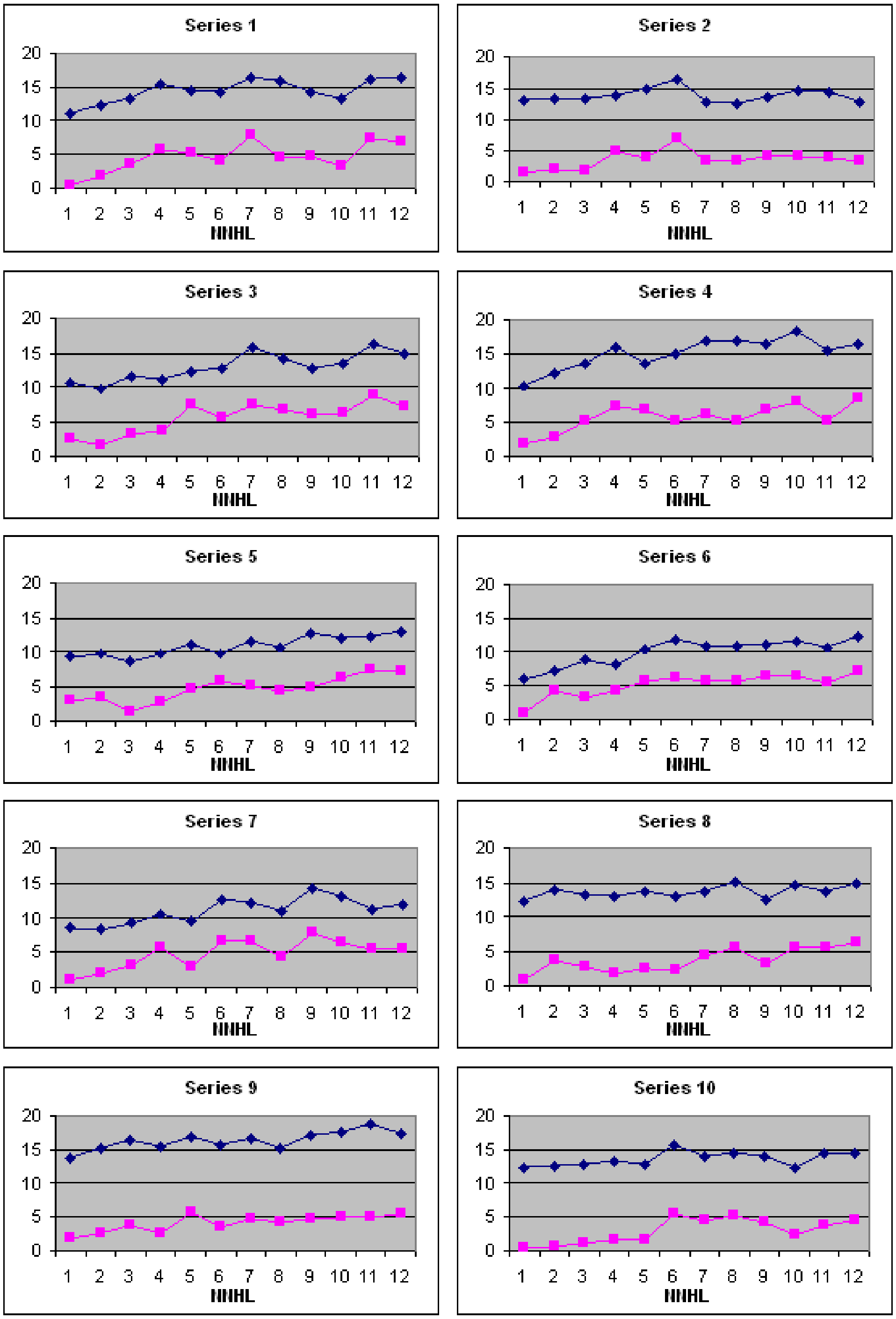


**Figure 2. The graphs of calculated values based on the number of neurons in the input layer for time series with period 4**

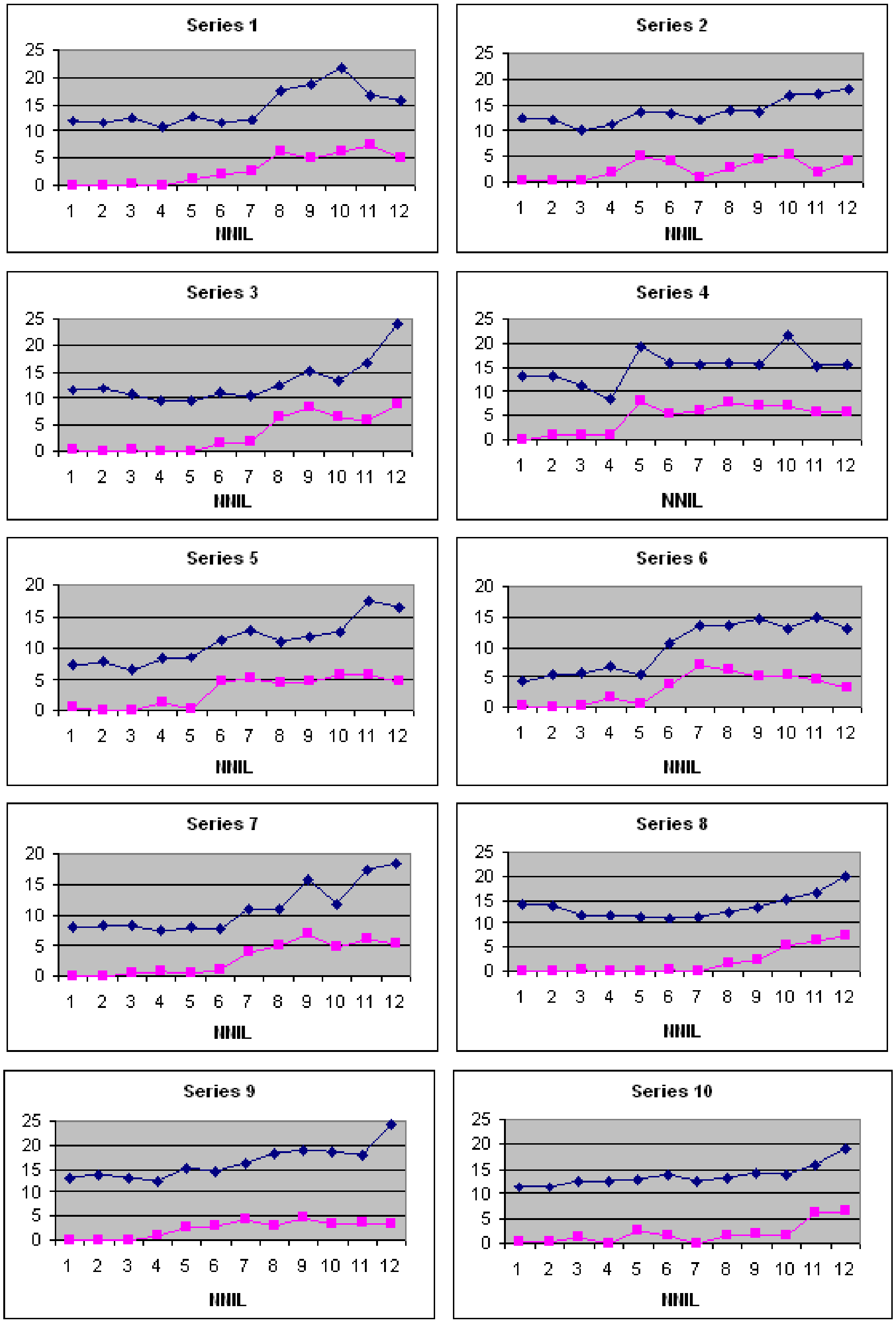


**Figure 3. The graphs of calculated mean values based on the number of neurons in the hidden layer for time series with period 4**

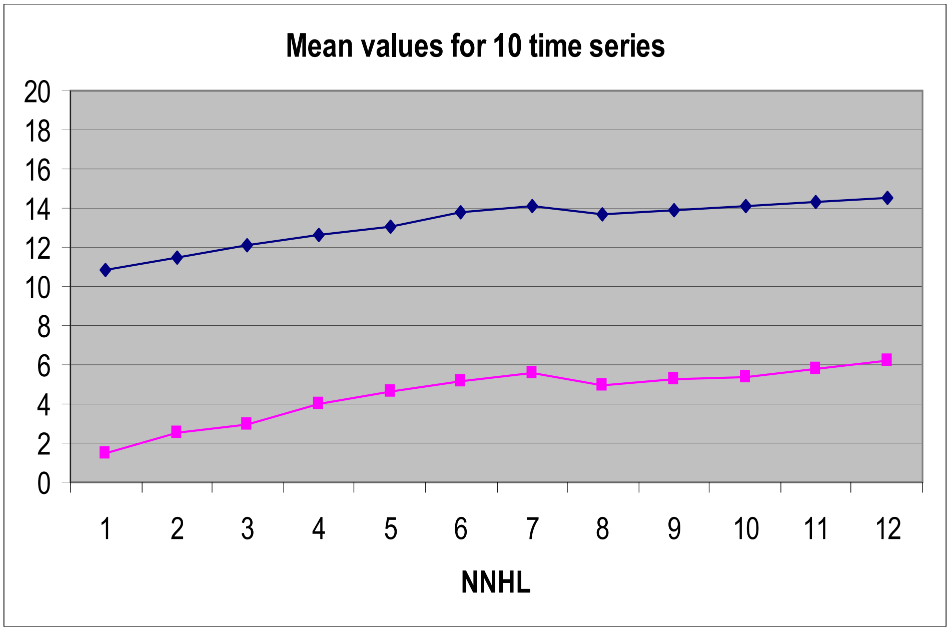


**Figure 4. The graphs of calculated mean values based on the number of neurons in the input layer for time series with period 4**

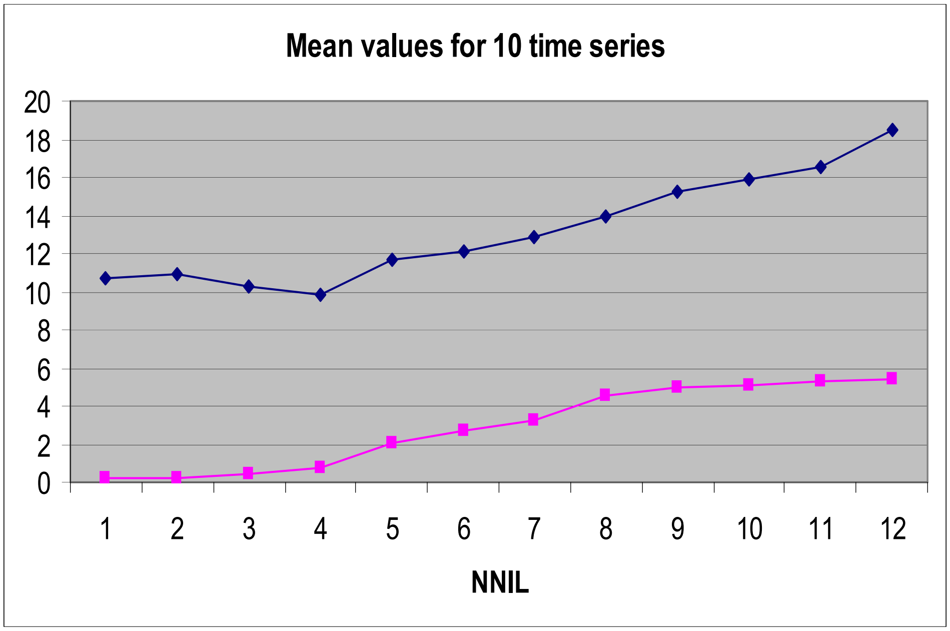


**Figure 5. The graphs of calculated values based on the number of neurons in the hidden layer for time series with period 6**

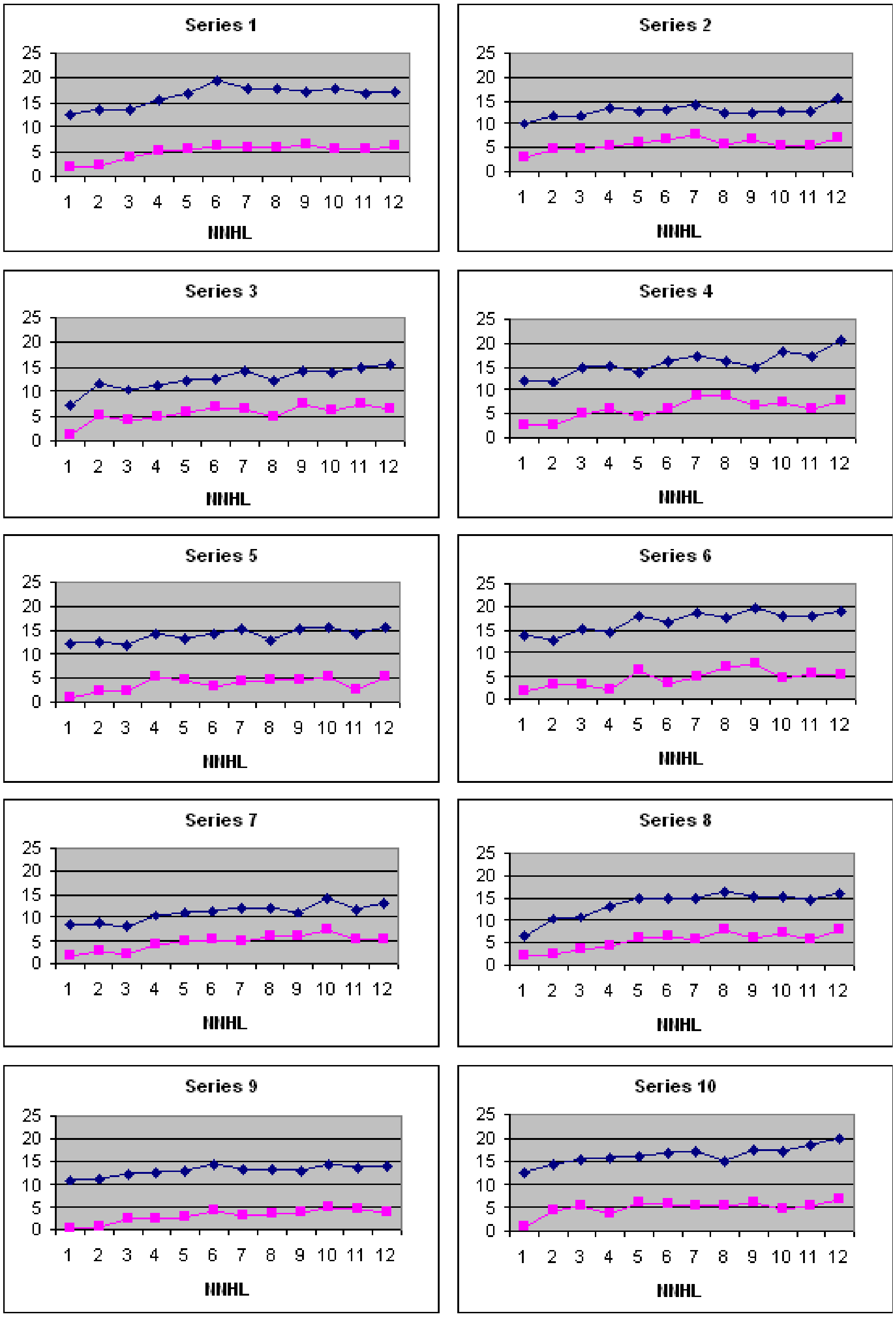


**Figure 6. The graphs of calculated values based on the number of neurons in the input layer for time series with period 6**

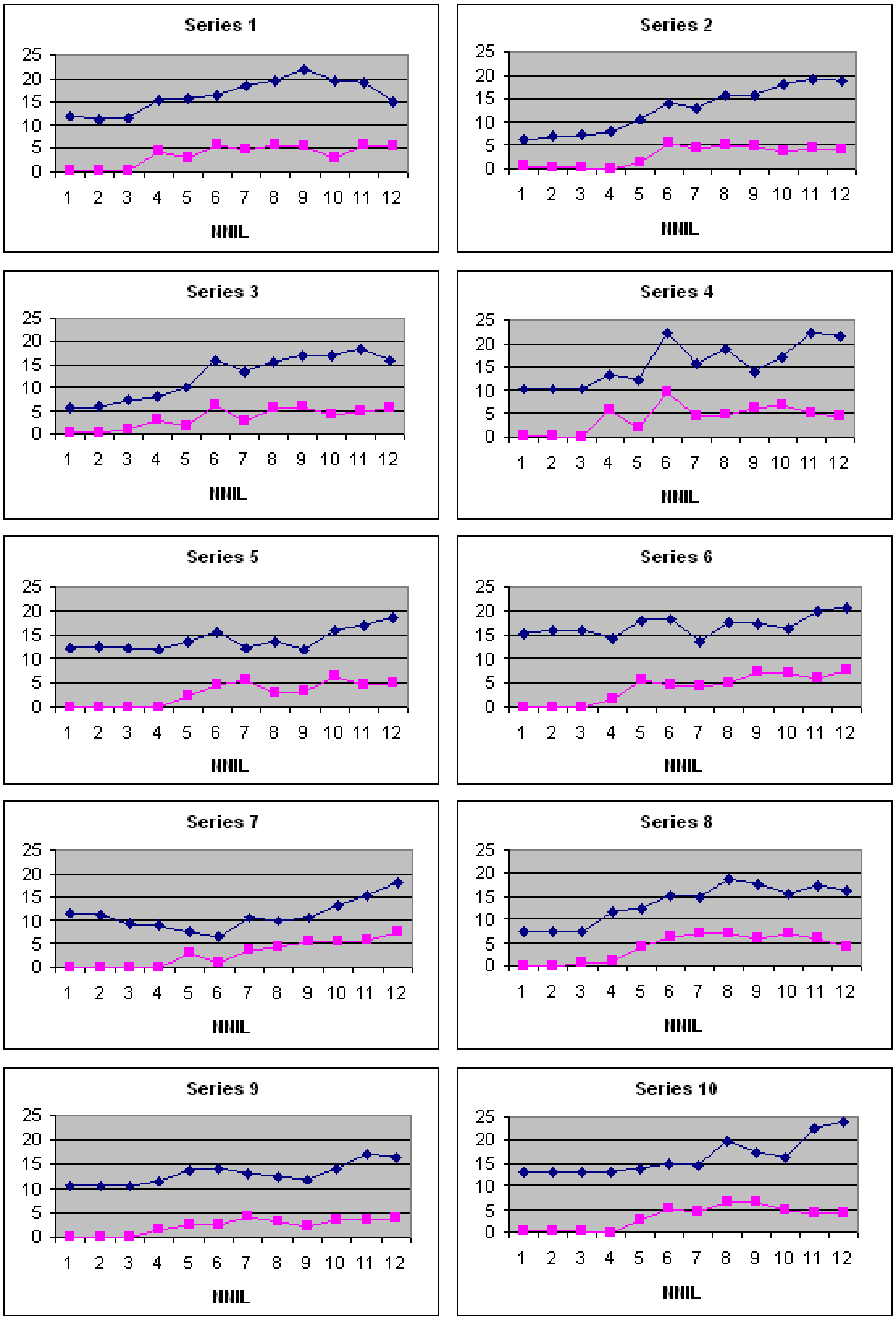


**Figure 7. The graphs of calculated mean values based on the number of neurons in the hidden layer for time series with period 6**

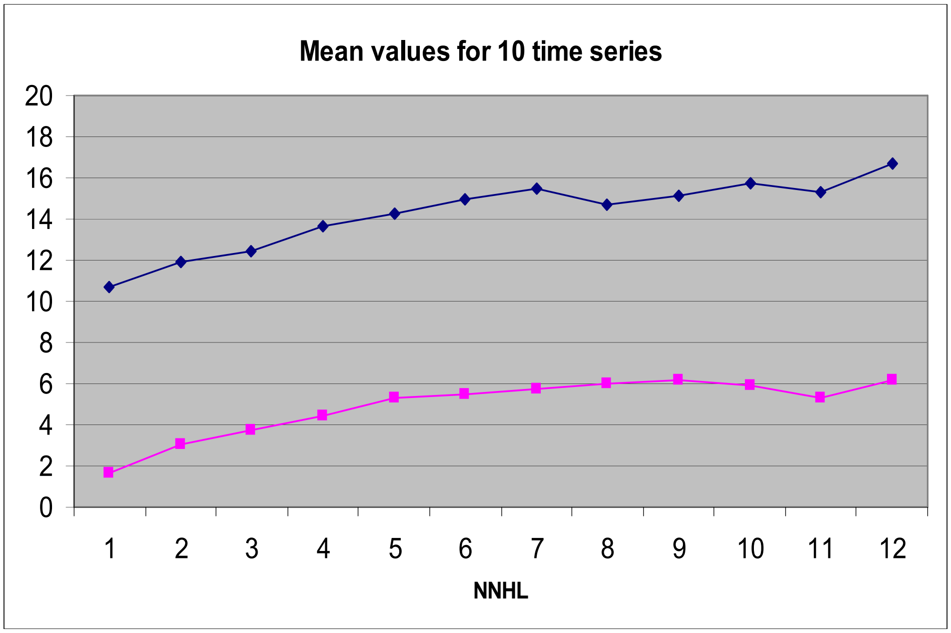


**Figure 8. The graphs of calculated mean values based on the number of neurons in the input layer for time series with period 6**

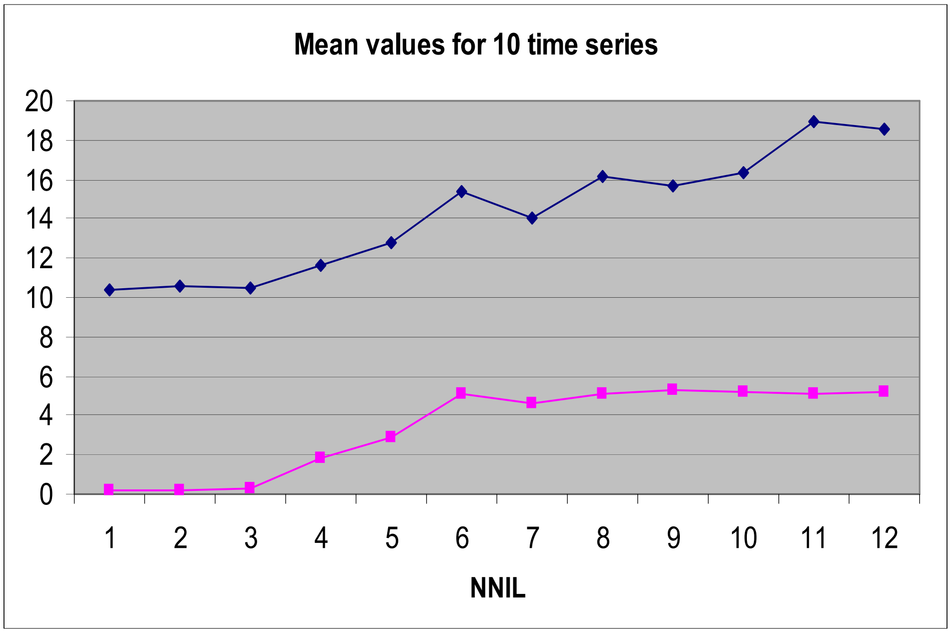


**Figure 9. The graphs of calculated values based on the number of neurons in the hidden layer for time series with period 12**

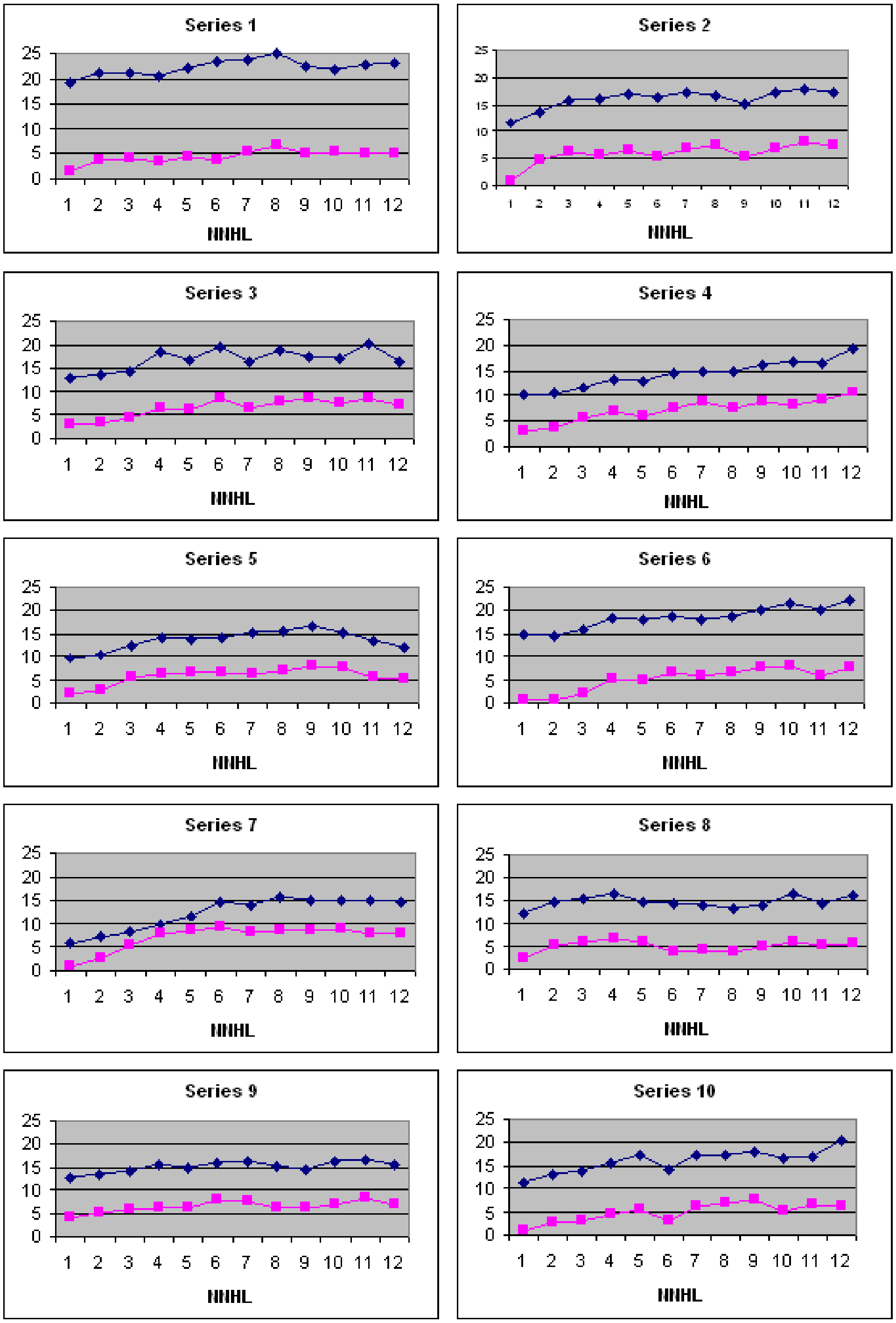


**Figure 10. The graphs of calculated values based on the number of neurons in the input layer for time series with period 12**

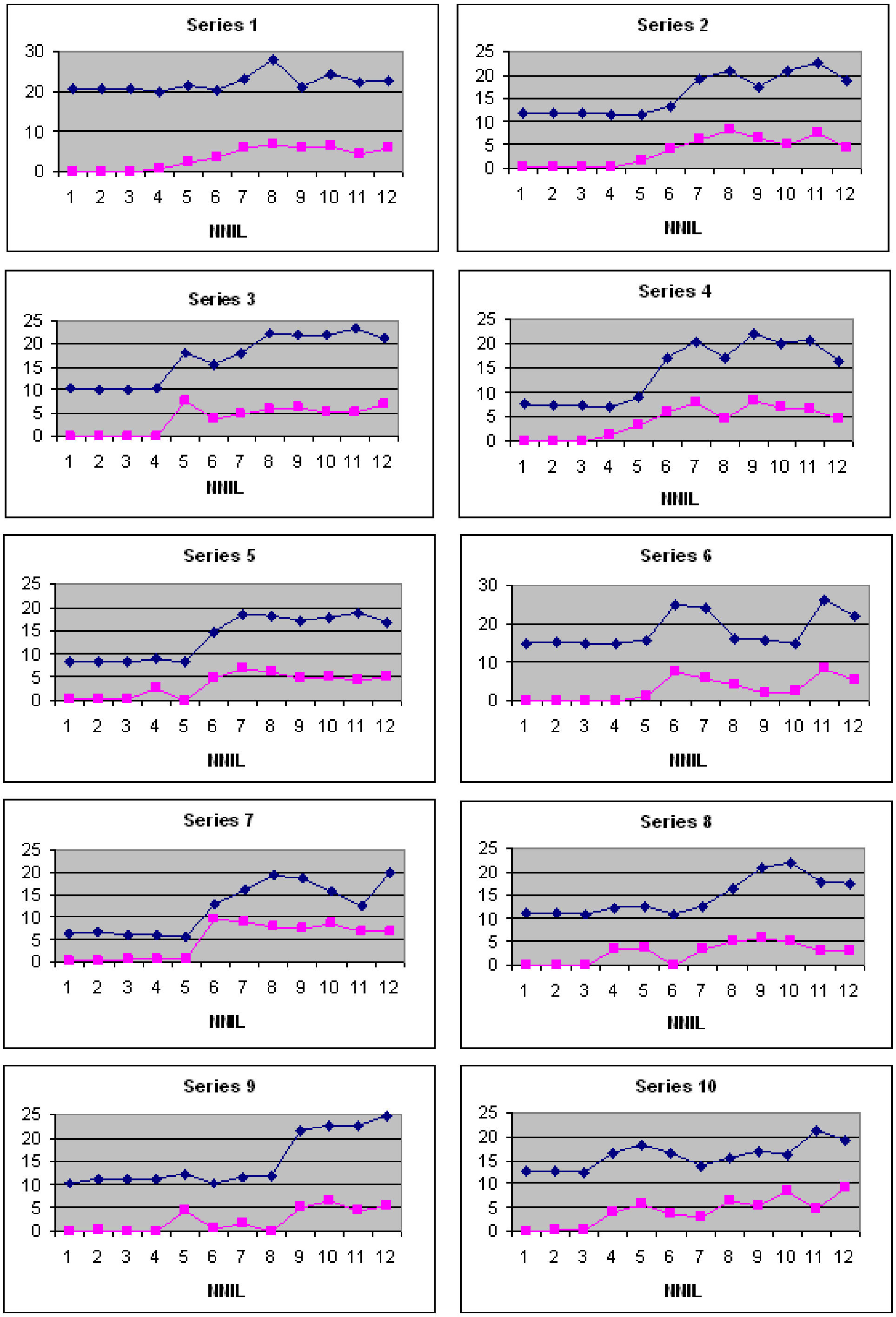


**Figure 11. The graphs of calculated mean values based on the number of neurons in the hidden layer for time series with period 12**

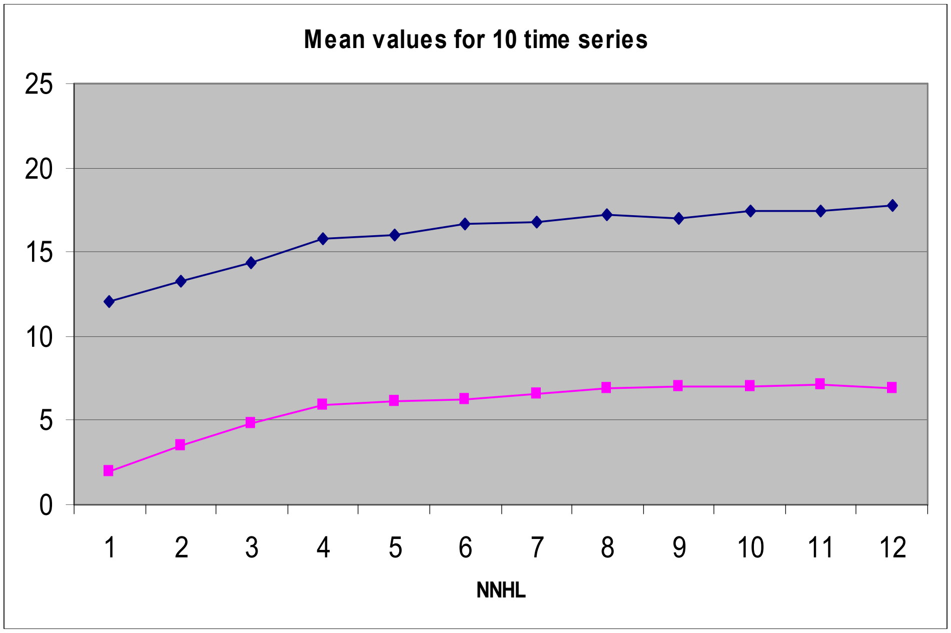
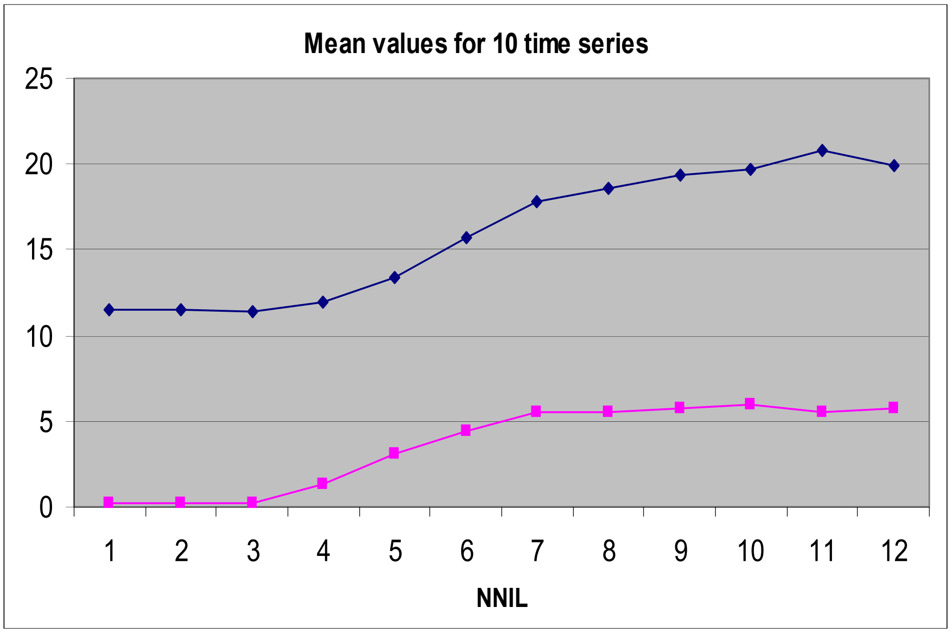


**Figure 12. The graphs of calculated mean values based on the number of neurons in the input layer for time series with period 12**



## 4. The proposed architecture selection strategy

In Figure 1, it is seen that as a general pattern the curves that are related to the mean value are parallel to the horizontal axes. This indicates that changing the number of neurons in the hidden layer affects FRMSE values only slightly. Once the curves that are related to the standard deviations are examined, the parallel structure is again observed. However, in the case of having a larger number of neurons in the hidden layer, it is observed that the values of the standard deviation are larger. Based on the derived information, lower FRMSE values can be obtained by increasing the number of neurons in the hidden layer. Besides, when one looks at Figure 3, which shows the mean of all series with period 4, the same observations can be made in general.

According to the graphs in Figure 2, it is seen that the FRMSE values tend to increase as the number of neurons in the input layer increases. Thus, better forecasts can be obtained based on a smaller numbers of neurons in the input layer, so fewer neurons in the input layer should be preferred. For the number of neurons in the input layer between 1 through 4, values of the standard deviations are very close to zero. Standard deviation is a measure of variation and if it is equal to zero, no variation is seen among the values. Therefore, when the number of input neurons is equal and fewer than 4, no matter how many neurons in the hidden layer are chosen, the FRMSE value is not affected by this choice. In the case of using more than 4 input neurons, FRMSE values has a tendency to increase as the number of neurons in the input layer rises. Also, the same holds for the values of the standard deviation. However, it is possible to obtain the lowest FRMSE value when using more input neurons since the values of the standard deviation has a tendency to increase. The FRMSE values where the number of input neurons varies between 6 and 10 are smaller than those when the number of input neurons is 11 or 12, and larger than when the number of input neurons varies between 1 and 5. The values of the standard deviation of the FRMSEs where the number of input neurons varies between 6 and 10 are close to those where they are 11 or 12, and larger than the case for 1 and 5. Therefore, the best architecture, which has the lowest FRMSE value, can be most probably found among the architectures in which the number of input neurons takes the values between 6 and 10 and the number of neurons in the hidden layer takes all possible values. When Figure 4, which shows the mean values of all data is examined carefully, the aforementioned results can be easily generalized. The explanation given above has led to a new architecture selection strategy for seasonal first order autoregressive time series whose period is 4. This proposed strategy is given below.

Step 1: Set the number of input neurons to 1, then assign any number between 1 and 12 for the number of neurons in the hidden layer. Obtain the FRMSE value for this architecture. Then repeat the process for 2, 3 and 4 input neurons. Select the best architecture, that is the one with the lowest FRMSE value among these 4 architectures. If the FRMSE value is low enough, this architecture is accepted as the best architecture. If not, proceed to the next step.

Step 2: Construct 60 different architectures where the number of input neurons varies between 6 and 10, and the number of neurons in the hidden layer between 1 and 12. Select the best architecture based on the lowest FRMSE value among these architectures. Compare the FRMSE values of the architectures found in Step 1 and Step 2 and select as the best architecture the one with the lower FRMSE.

Instead of using 144 architectures, the proposed selection strategy has enabled us to employ fewer architectures, from as little as 4 to a maximum of 64.

Similar results can be derived from the periods 6 and 12 by using Figures 5 through 12. The proposed strategy used for period 4 can be easily employed for the periods 6 and 12. Also, the number of periods is seen to be unimportant in analyzing time series using ANN. It can be seen that the proposed architecture selection strategy can be used to obtain lower FRMSE values in seasonal first order autoregressive time series without taking the period into account.

The success of the proposed selection strategy in this study can be deduced from Table 1. Based on the information derived from the 10 time series generated for each of the 3 different periods, the best architectures are determined for 9 out of 10 of the time series by using the proposed strategy. In the cases where the best architecture is not determined, the second best architecture is determined by the proposed selection strategy. The related results are given in Table 2.

**Table 2. The first and second best architectures based on the value of FRMSE**

| | | Period | | |
|---|---|---|---|---|
| | | 4 | 6 | 12 |
| | | 10th Series | 1st Series | 8th Series |
| The best | FRMSE | 7.230864 | 5.706968 | 7.068751 |
| | Arch. | 11 - 10 | 12 - 3 | 5 - 11 |
| The second best | FRMSE | 11.13902 | 9.947635 | 10.35416 |
| | Arch. | 1 - 12 | 6 -1 | 8 -10 |

## 5. Results and Discussion

In the literature, there is no an architecture selection strategy for analyzing time series using ANN. Selections of ANN architecture are made based on observation or chosen arbitrarily in general. In this study, to avoid the error which may be caused by the selection of architecture arbitrarily or heuristically, a new architecture selection strategy is proposed. Also, when the proposed strategy is used, instead of examining a large number of architectures, better forecasts can be obtained by examining less architecture. When the propose selection strategy was employed for the 30 time series generated for this study, it selected the best architectures for 27 out of 30 series, and the second best for the rest. According to the results obtained, the proposed selection strategy enables one to determine the architectures which produce the lowest FRMSE values.

In the literature, the number of input neurons is often taken heuristically as the period of the studied time series. In the result of this simulation study, it is seen that taking the number of input neurons equal to the period is a false assumption. This is a vital finding that will have important consequences for future studies.

We would like to point out that the simulation study here is conducted for stationary SAR(1) time series. If non-stationary time series are available, then the proposed architecture selection strategy can be employed for the time series whose structure is SAR(1) after taking differences. It is possible to analyze non-stationary time series using ANN without taking differences. However, a new simulation study on which to base a strategy would probably be necessary. Zhang and Qi [17] suggested as a result of a simulation study that non-stationary time series including trend should be converted to stationary time series before applying ANN. Implementations of this suggestion could well be supported by our proposed strategy.

## References

[1] Aladag, C. H. and Egrioglu, E. *A forecasting study of time series based on the artificial neural Networks*, Proceedings of 14th Statistical Research Conference 2005, 397-406, 2005.

[2] Buhamra, S., Smaoui, N. and Gabr, M. *The Box-Jenkins analysis and neural networks: Prediction and time series modelling*, Applied Mathematical Modelling **27**, 805–815, 2003.

[3] Egrioglu, E., Aladag, C. H. and Gunay, S. *A new model selection strategy in artificial neural networks*, Applied Mathematics and Computation **195**, 591–597, 2008.

[4] Gunay, S., Egrioglu, E. and Aladag, C. H. *Introduction to single variable time series analysis* (Hacettepe University Press, 2007).

[5] Hecth-Nielsen, R. *Neurocomputing* (Addison-Wesley, Menlo Park, CA., 1990).

[6] Hwrang, H. B. *Insights into neural network forecasting of the time series corresponding to ARMA(p,q) structures*, The International Journal of Management Science **29**, 273–289, 2001.

[7] Kang, S. *An Investigation of the use of feedforward neural networks for forecasting* (PhD. Thesis, Kent State University, 1991).

[8] Lachtermacher, G. and Fuller, J. D. *Backpropagation in time-series forecasting*, Journal of Forecastinng **14**, 381–393, 1995.

[9] Lippmann, R. P. *An introduction to computing with neural nets*, IEEE ASSP Magazine, April, 4–22, 1987.

[10] Picton, P. D. *Introduction to Neural Networks* (Macmillan Press Ltd., 1994).

[11] Sharda, R. and Patil, R .B. *Connectionist approach to time series prediction: An emprical test*, Journal of Intelligent Manufacturing **3**, 317–323, 1991.

[12] Tang, Z., Almeıda, C. and Fishwick P. A. *Time series forecasting using neural networks vs Box-Jenkins methodology*, Simulation **57**, 303–310, 1991.

[13] Tang, Z. and Fishwick, P. A. *Feedforward neural nets as models for time series forecasting*, Operations Research Society of America **5**, 374–385, 1993.

[14] Wong, F. S. *Time series forecasting using backpropagation neural networks*, Neurocomputing **2**, 147–159, 1991.

[15] Zhang, G. P., Patuwo, B. E. and Hu, Y. M. *Forecasting with artificial neural networks: The state of the art*, International Journal of Forecasting **14**, 35–62, 1998.

[16] Zhang, G.P., Patuwo, B. E. and Hu, Y. M. *A simulation study of artificial neural networks for nonlinear time-series forecasting*, Computers and Operations Research **28**, 381–396, 2001.

[17] Zhang, G. P. and Qi, M. *Neural network forecasting for seasonal and trend time series*, European Journal of Operational Research **160**, 501–514, 2005.

[18] Zurada, J. M. *Introduction of artificial neural systems* (St. Paul: West Publishing, 1992).