Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 11, Number 1 (2015), pp. 137-148

http://www.ripublication.com

Wavelet Neural Network Model Selection for Nonlinear-Seasonal Time Series Forecasting

Umu Sa'adah[1], Subanar[2], Suryo Guritno[3] and Suhartono[4]

[1]*Department of Mathematics, Faculty of Mathematics and Natural Sciences, University of Brawijaya, Malang, Indonesia*
PhD Student at Department of Mathematics, Faculty of Mathematics and Natural Sciences, Gadjah Mada University, Yogyakarta, Indonesia;
u.saadah@ub.ac.id
[2,3]*Department of Mathematics, Faculty of Mathematics and Natural Sciences, Gadjah Mada University, Yogyakarta, Indonesia*
[4]*Department of Statistics, Faculty of Mathematics and Natural Sciences, Sepuluh November Institute of Technology, Surabaya, Indonesia*

ABSTRACT

A forecasting model based on a hybrid wavelet transform and neural network for nonlinear-seasonal time series is proposed. The decomposition of Maximal Overlap Discrete Wavelet Transform (MODWT) is used as a method of data preprocessing to obtain wavelet and scaling coefficients. The main problem in a hybrid model of Wavelet Neural Network (WNN) is how to determine the optimal combination of the number of neurons in the hidden layer, the number of neurons in the input layer and which lags of the wavelet and scaling coefficients will be used at each scale. In this study, data were generated from a nonlinear-seasonal function. Learning algorithm used in Multi-Layer Perceptron (MLP) is the resilient backpropagation algorithm. Inputs of WNN are seasonal and near-seasonal lags from the wavelet and scaling coefficients, in addition to the same inputs as the Multiscale Autoregressive (MAR) model inputs. The neurons in the input layer are selected based on the Wald test statistic. The number of neurons in the hidden layer is determined by the Akaike's Information Criterion (AIC). All experiments are performed using the R software. The result shows that the optimum WNN model architecture accommodates the seasonal and near-seasonal lags of the wavelet and scaling coefficients as input units.

Keywords: nonlinear, seasonal, near-seasonal, MODWT, wavelet neural network, resilient backpropagation.

1. INTRODUCTION

A wavelet transformation is a decomposition technique for the time series or signal that can produce a good multiresolution in both the time and frequency domain. The multiresolution decomposition separates the trend from the signal or time at different levels of the wavelet and scaling coefficient, so that it processes information effectively at different scales and can be very useful for feature detection from complex time series. Many articles about the wavelet applications have been presented in various kinds of statistical analysis to solve many problems. Some examples of wavelet applications are the use of wavelets and some of their generalizations [1,4], the use of wavelet transform for time series analysis [6,7,8].

In recent years, a forecasting model based on a hybrid wavelet transform and neural network for time series is introduced. Murtagh et al. [5] proposed a hybrid between Discrete Wavelet Transform (DWT) by genetic algorithms to support neural networks for chaotic time series forecasting, Renaud et al. [8] and Shin and Han [10] proposed a hybrid between MODWT by Feed-Forward Neural Network (FFNN) for time series forecasting based on autoregressive model and Suhartono [11] proposed a hybrid between MODWT and FFNN with backpropagation algorithm for seasonal time series forecasting. In this paper, we present a hybrid between MODWT and FFNN with resilient backpropagation algorithm for nonlinear-seasonal time series forecasting. The resilient backpropagation algorithm was introduced by Riedmiller and Braun [9] and was implemented by Günther and Fritsch [3] in R software.

One of the most applications of FFNN is architecture building from an unknown input-output variables relation. In the architecture building, there is often some unavoidable and inherent randomness in relationship between input-output. This lack of an appropriate relation generally arises because inputs do not contain all variables influencing output. To develop an optimum WNN architecture is begun by using nonlinearity test, such as Terasvirta test [12] to know the existence of nonlinear relationship between input-output. It is important to determine the optimal combination of neurons in the input and hidden layer, when there is a nonlinear relationship, but this is a difficult step. Determination of the optimal combination can use the statistic test or the information criterion as proposed in [2,11].

In particular on the use of wavelet decomposition for time series forecasting, the first point is to know how many and which lags of the wavelet and scaling coefficients will be used at each scale as FFNN input. In this paper, we propose an extended WNN model for nonlinear-seasonal time series forcasting. The extended WNN model accommodates some appropriate lags additions (i.e. seasonal and near-seasonal lags) of scaling and wavelet coefficients besides lag inputs selection that proposed by Renaud et al. [8]. This seasonal and near-seasonal lags addition is motivated by the fact that there are several Cross Correlation Functions (CCFs) between a stationary seasonal time series data X and each of wavelet coefficients or scaling coefficients that form appropriate significant and seasonal pattern. It means that the seasonal and near-seasonal lags from the wavelet or scaling coefficients have influence on X.

The j^{th} level MODWT can be implemented for any sample size, whereas the j^{th} level DWT can only be implemented for the sample size $N = 2^j$, where j is positive integer. Hence, in this paper we use MODWT as decomposition method with level $j = 1, 2, 3, 4, 5, 6$ and wavelet filter "Haar".

2. MATERIALS AND METHODS

2.1. Data

In this study, 240 data are generated from a nonlinear-seasonal function having the following equation**:**

$$X_t = 3X_{t-12}\exp\left(-0.1X_{t-12}^2\right) + e_t, \quad e_t \sim N(0,1) \tag{1}$$

The data are divided to be two parts. The first 180 data are the training data and the remaining 60 data are the testing data.

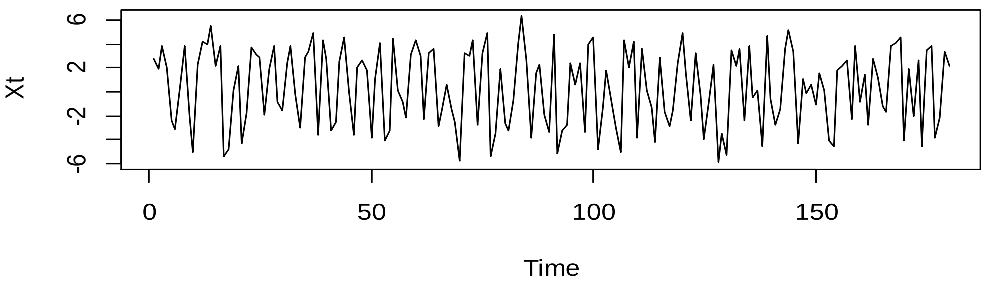

Figure 1. Plot of the training data from simulation data.

2.2. Maximal Overlap Discrete Wavelet Transform (MODWT)

One of wavelet transformation methods is MODWT. It is a modification from DWT. The efficient calculation of MODWT is also done by pyramid algorithm as in DWT. In efficient pyramid algorithm, computation on the j^{th} level MODWT wavelet and scaling coefficients and is based on the scaling coefficients of level *j*-1. The key of this algorithm is to note the relationship among the filters used to compute the wavelet and scaling coefficients of level *j*-1 and *j* [4].

According to Percival and Walden [7], a real-value wavelet filter $\{h_l : l = 0,1,\ldots,L-1\}$ having width L must satisfy the following three conditions: L must be an even integer, $h_0 \neq 0$ and $h_{L-1} \neq 0$. Defined $h_l = 0$ for $l < 0$ and $l \geq L$ so that $\{h_l\}$ is actually an infinite sequence with at most L nonzero values. A MODWT wavelet filter $\{\tilde{h}_l\}$ is defined via $\tilde{h}_l \equiv h_l / \sqrt{2}$ and must satisfy the following properties:

$$\sum_{l=0}^{L-1} \tilde{h}_l = 0, \quad \sum_{l=0}^{L-1} \tilde{h}_l^2 = \frac{1}{2} \text{ and } \sum_{l=-\infty}^{\infty} \tilde{h}_l \tilde{h}_{l+2m} = 0, \tag{2}$$

for all nonzero integers *m*. Similarly, a MODWT scaling filter $\{\tilde{g}_l\}$ is defined via $\tilde{g}_l \equiv g_l / \sqrt{2}$ and must satisfy the following properties:

$$\sum_{l=0}^{L-1} \tilde{g}_l = 1, \quad \sum_{l=0}^{L-1} \tilde{g}_l^2 = \frac{1}{2} \text{ and } \sum_{l=-\infty}^{\infty} \tilde{g}_l \tilde{g}_{l+2m} = 0, \tag{3}$$

for all nonzero integers *m*. Coefficients $\tilde{h}_l$ and $\tilde{g}_l$ have the following relationship:

$$\tilde{g}_l = (-1)^{l+1} \tilde{h}_{L-1-l} \text{ and } \tilde{h}_l = (-1)^l \tilde{g}_{L-1-l} . \tag{4}$$

For any sample size *N*, the wavelet and scaling coefficients from MODWT level *j* to be *N*-dimensional vectors w_j and v_j have elements respectively,

$$w_{j,t} \equiv \sum_{l=0}^{L_j-1} \tilde{h}_{j,l} X_{t-l \bmod N} \text{ and } v_{j,t} \equiv \sum_{l=0}^{L_j-1} \tilde{g}_{j,l} X_{t-l \bmod N} , \tag{5}$$

$t = 0,1,...,N-1$, $L_j - 1 \equiv (2^j - 1)(L-1)$, where $\tilde{h}_{j,l} \equiv h_{j,l} / 2^{j/2}$ and $\tilde{g}_{j,l} \equiv g_{j,l} / 2^{j/2}$. The filters $\{\tilde{h}_{j,l}\}$ and $\{\tilde{g}_{j,l}\}$ are called the j^{th} level MODWT wavelet and scaling filter.

Assume a stationary time series data $X = (X_1, X_2, ..., X_t)$, for any dimension *N*. The decomposition based on MODWT using Haar wavelet filter and level *J* for stationary time series data *X* yields *J* levels of the Haar wavelet and scaling coefficients, that is w_j and v_j, $j = 1,...,J$. The MODWT is computed via the pyramid algorithm, using pseudo code written by Percival and Walden [7]. When the boundary method used in the decomposition is "periodic" the resulting Haar wavelet and scaling coefficients are computed without making changes to the length of original series. The Haar wavelet transform from the stationary time series data *X* can be represented by $W = w_1, w_2, ..., w_J, v_J$, where $X = \sum_{j=1}^{J} w_j + v_J$.

2.3. Multiscale Autoregressive (MAR)

Autoregressive (AR) model is one of the simplest linear model for prediction. MAR model is one of modifications from AR model that proposed by Renaud et al. [8]. The decomposition based on MODWT yields wavelet and scaling coefficients. Some certain lags, i.e. wavelet coefficients lags $w_{j,t-2^j(k-1)}$ and scaling coefficients lags $v_{J,t-2^J(k-1)}$ for order $k = 1,...,A_j$ and level $j = 1,...,J$ are used as inputs in MAR model. The one-step forward forecast of a MAR process by order A_j, called MAR(A_j) is written as

$$\hat{X}_{t+1} = \sum_{j=1}^{J}\sum_{k=1}^{A_j} \hat{a}_{j,k} w_{j,t-2^j(k-1)} + \sum_{k=1}^{A_j} \hat{a}_{J+1,k} v_{J,t-2^J(k-1)} , \tag{6}$$

where $\hat{a}_{j,k}$ and $\hat{a}_{J+1,k}$ are values of MAR coefficients. To estimate parameters in MAR model, it is used Normal Equation which follows Least Squares Principle.

2.4. Wavelet Neural Network

The MAR forecasting model is actually linear. Furthermore, the MAR model by Renaud et al. [8] was extended to another model, linear or nonlinear model. The architecture of the MLP neural network or known as FFNN consists of one hidden layer with *Q* neurons and inputs consists of the wavelet and scaling coefficients lags in the same manner as inputs in MAR model. The one-step forward forecast of the model is written as

$$\hat{X}_{t+1} = f_2\left[\sum_{q=1}^{Q} \widehat{b}_q f_1\left(\sum_{j=1}^{J}\sum_{k=1}^{A_j} \hat{a}_{j,k,q} w_{j,t-2^j(k-1)} + \sum_{k=1}^{A_j} \hat{a}_{J+1,k,q} v_{J,t-2^J(k-1)}\right)\right] \tag{7}$$

where Q is the number of neurons in the hidden layer, $\hat{a}_{j,k,q}$ and $\hat{a}_{J+1,k,q}$ are connection weight from neurons in the input layer to neurons in the hidden layer, $\widehat{b}_q$ is connection weight from neurons in the hidden layer to output or target, f_1 and f_2 are the activation functions in the hidden layer and the output layer respectively. Further, model in Equation (7) is known as Wavelet Neural Network (WNN).

The WNN forecasting model does not actually accommodate inputs from seasonal lags. In this paper, we extend the WNN model for nonlinear-seasonal time series forecasting. The architecture of extended WNN model consists of one hidden layer with *Q* neurons and inputs that accommodate seasonal and near-seasonal lags from the wavelet and scaling coefficients, in addition to the inputs that propose by Renaud et al. [8]. The one-step forward forecast of the extended WNN model is written as

$$\hat{X}_{t+1} = f_2\left(\sum_{q=1}^{Q} \widehat{b}_q f_1\left(\begin{array}{l} \displaystyle\sum_{j=1}^{J}\sum_{k=1}^{A_j} \hat{a}_{j,k,q} w_{j,t-2^j(k-1)} + \sum_{k=1}^{A_j} \hat{a}_{J+1,k,q} v_{J,t-2^J(k-1)} + \sum_{j=1}^{J}\sum_{k=1}^{A_j} \hat{a}_{j,Sp,q} w_{j,t-Sp} \\ \displaystyle + \sum_{k=1}^{A_j} \hat{a}_{J+1,Sp,q} v_{J,t-Sp} + \sum_{j=1}^{J}\sum_{k=1}^{A_j} \hat{a}_{j,Sp-1,q} w_{j,t-(Sp-1)} + \sum_{k=1}^{A_j} \hat{a}_{J+1,Sp-1,q} v_{J,t-(Sp-1)} \\ \displaystyle + \sum_{j=1}^{J}\sum_{k=1}^{A_j} \hat{a}_{j,Sp+1,q} w_{j,t-(Sp+1)} + \sum_{k=1}^{A_j} \hat{a}_{J+1,Sp+1,q} v_{J,t-(Sp+1)} \end{array}\right)\right) \tag{8}$$

where S is number of significant seasonal, p is period of seasonal, $\hat{a}_{j,Sp,q}$ and $\hat{a}_{J+1,Sp,q}$ are connection weight from seasonal lag inputs in the input layer to neurons in the hidden layer, $\hat{a}_{j,Sp-1,q}$, $\hat{a}_{J+1,Sp-1,q}$, $\hat{a}_{j,Sp+1,q}$ and $\hat{a}_{J+1,Sp+1,q}$ are connection weight from near-seasonal lag inputs in the input layer to neurons in the hidden layer. f_1 is a sigmoid logistic activation function in the hidden layer and f_2 is a linear function in the output layer.

2.5. Methodology

There are seven steps performed in this research, namely:

1. Examine stationary of the data in the mean and in the variance. If the data are non stationary, transform it in order to stationary in the mean and in the variance.
2. Decompose the stationary data using MODWT of level $j = 1, 2, 3, 4, 5, 6$ and Haar wavelet filter.
3. Examine CCF between the simulation data and the wavelet and scaling coefficients at each of scale, to detect existence of the significant seasonal pattern.
4. Select the lags from wavelet and scaling coefficients as inputs.
5. Apply nonlinearity test using Terasvirta test for detecting nonlinearity relationship between input and output of the WNN model.
6. If there is nonlinearity relationship, determine the optimum WNN model. The number of neurons in the input layer is determined by the Wald test statistic and the number of neurons in the hidden layer is decided by the AIC value.
7. Choose the optimum model based on the smallest value of the Root Mean Square Error (RMSE) from the testing data.

For 4th step, there are two methods for selection inputs in WNN model, i.e.:

1. Inputs in WNN model are the lags of wavelet and scaling coefficients such as that proposed by Renaud et al. [8].
2. Inputs in WNN model accommodate some appropriate lags addition (i.e. seasonal and near-seasonal lags) of scaling and wavelet coefficients besides lags input selection that proposed by Renaud et al. [8].

3. RESULTS AND DISCUSSION

Figure 1 show the time series plot from the simulation data. The simulation data lies between -6 and 6, also indicate that the mean of the data are around 0. Thus, the simulation data are stationary in mean. The result of Box-Cox analysis shows that the simulation data are stationary in variance. Hence, the result of the examination is stationary data in both the mean and variance as a condition of modeling WNN. The first step has been met. The second step, the stationary simulated data are decomposed using MODWT and is done with Haar wavelet filter and level $j = 1, 2, 3, 4, 5, 6$ as the initial processing of the data.

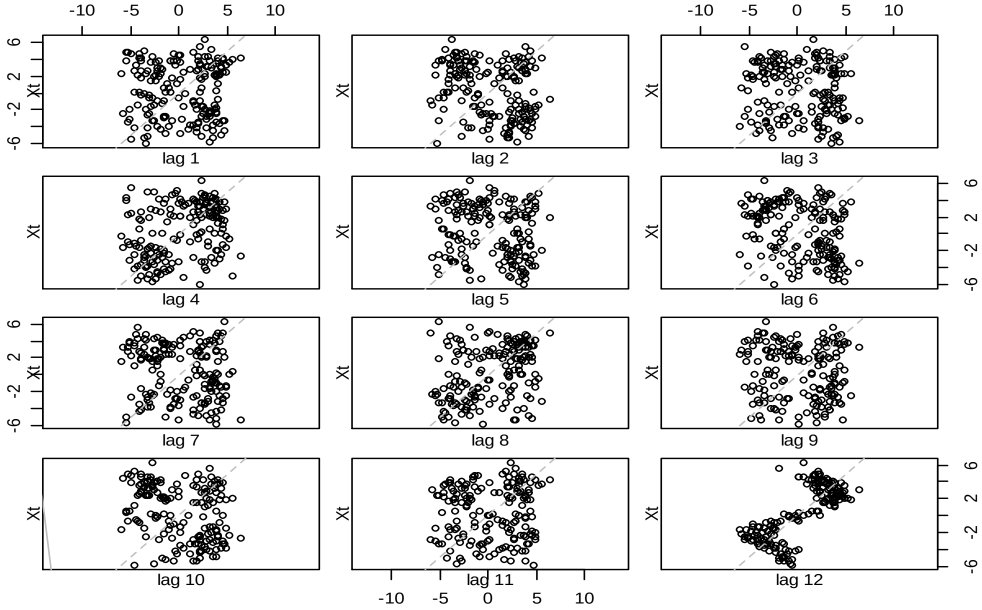

Figure 2. Plot of the training data from simulation data with its lags (from lag 1 to lag 12)

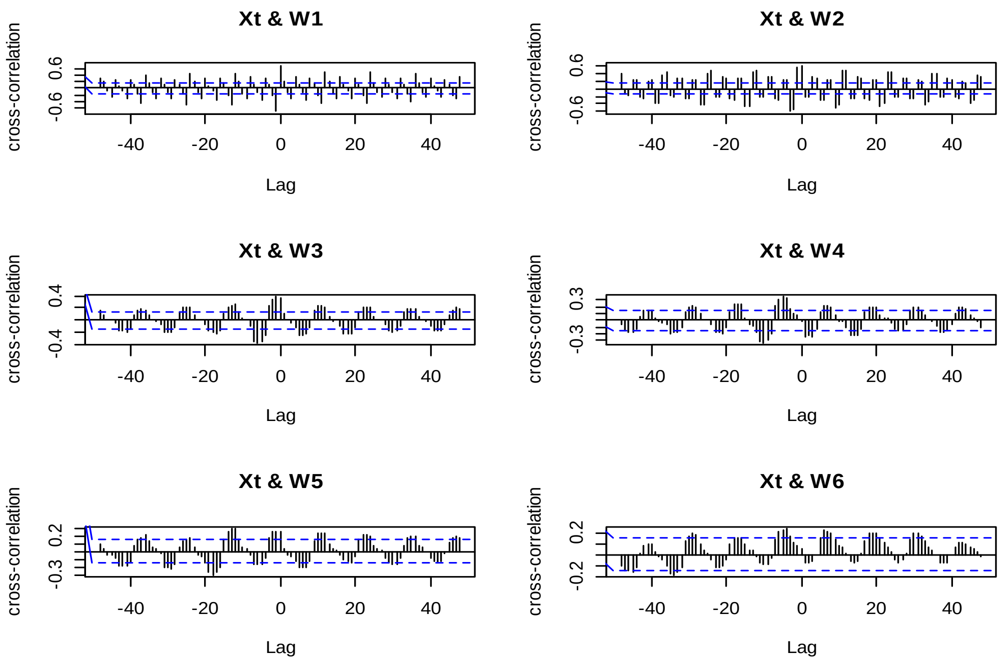

Figure 3. Cross Correlation between the training data from simulation data and the wavelet coefficients at level $j = 1, 2, 3, 4, 5, 6$**.**

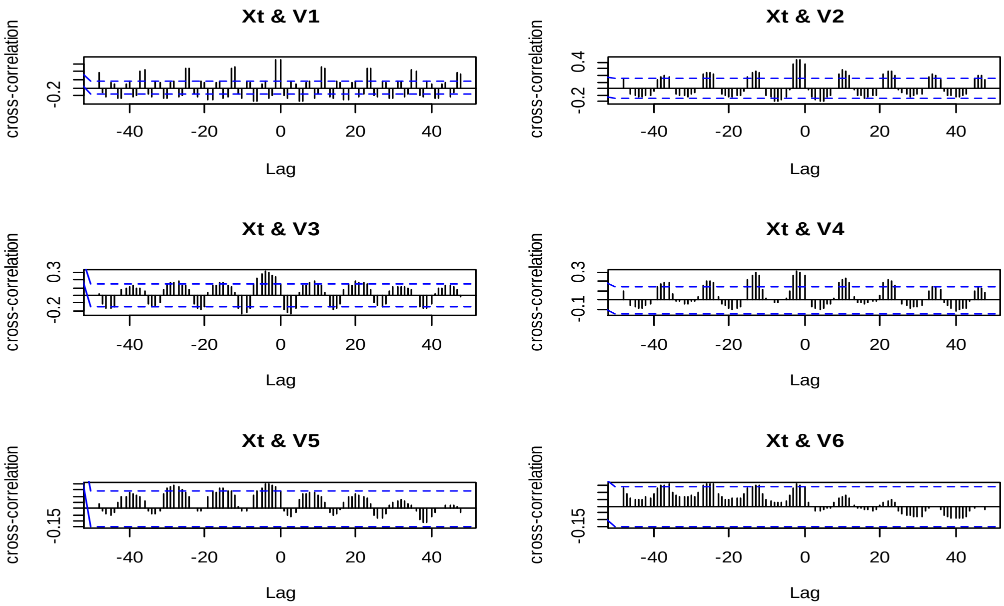

Figure 4. Cross Correlation between the training data from simulation data and the scaling coefficients at level $j = 1, 2, 3, 4, 5, 6$**.**

The third step is to detect existence of the significant seasonal pattern. Figure 2 is the lag plot of the simulation data. At 1^{st}-11^{th} lag, the plot of the simulation data has random pattern, whereas at 12^{th} lag, the data forms nonlinear pattern. So the simulation data are nonlinear and seasonal at 12^{th} period.

Plot of CCF between the simulation data and the wavelet coefficients at level $j = 1, 2, 3, 4, 5, 6$ is shown by Figure 3. At each level of MODWT, the plot of CCF forms seasonal pattern such as pattern of the simulation data. It means that the seasonal or near-seasonal lags of the wavelet coefficient have potential influence on the simulation data. The seasonal or near-seasonal lags are viewed significant at level $j = 1, 2$ and enough significant at level $j = 3, 4$.

Plot of CCF between the simulation data and the scaling coefficients at level $j = 1, 2, 3, 4, 5, 6$ is shown by Figure 4. At each of level MODWT, the plot of CCF forms seasonal pattern such as pattern of the simulation data. It means that the seasonal or near-seasonal lags of the scaling coefficient have potential influence on the simulation data. The seasonal or near-seasonal lags is also viewed significant at level $j = 1, 2$ and enough significant at level $j = 3, 4$.

Table 1. The result of significant-inputs selection by using the 1st method.

MAR; Level	Significant inputs of reduced model	AIC	R Square	RMSE-training	RMSE-testing
1; 1	$w_{1(t-1)}$, $v_{1(t-1)}$	1599.8111	0.1326	2.11323	3.36731
1; 2	$w_{2(t-1)}$, $v_{2(t-1)}$	1528.5521	0.1715	2.06533	3.39139
1; 3	$w_{2(t-1)}$, $v_{3(t-1)}$, $w_{3(t-1)}$	1529.8749	0.1718	2.06487	3.38551
1; 4	$w_{2(t-1)}$, $v_{4(t-1)}$, $w_{1(t-1)}$, $w_{3(t-1)}$	1452.2688	0.2153	2.01000	3.49811
1; 5	$w_{2(t-1)}$, $w_{3(t-1)}$, $w_{1(t-1)}$, $w_{5(t-1)}$	1415.1118	0.2355	1.98386	3.61055
1; 6	$w_{2(t-1)}$, $w_{6(t-1)}$	1537.3273	0.1667	2.07129	3.46634
2; 1	$w_{1(t-3)}$, $v_{1(t-3)}$	1329.4469	0.2729	1.93609	2.67900
2; 2	$w_{1(t-3)}$, $v_{2(t-5)}$	1295.1816	0.2877	1.92173	2.67730
2; 3	$w_{1(t-3)}$, $w_{3(t-1)}$, $w_{1(t-1)}$	1166.8653	0.3497	1.84300	2.62105
2; 4	$w_{1(t-3)}$, $w_{3(t-1)}$, $w_{1(t-1)}$, $v_{4(t-17)}$	971.2984	0.4136	1.71890	2.67350
2; 5	$w_{1(t-3)}$, $w_{5(t-33)}$, $w_{1(t-1)}$, $w_{3(t-1)}$, $w_{2(t-1)}$	875.8430	0.4283	1.71600	2.67997
2; 6	$w_{1(t-3)}$, $w_{5(t-33)}$, $w_{1(t-1)}$, $w_{5(t-1)}$	712.9469	0.4077	1.75087	4.35620

Table 2. Comparison of various WNN models by using the 1st method.

MAR; Level	Number of inputs from FM*	Reduced Model				
		Number of inputs	Number of seasonal lag inputs	Number of near-seasonal lag inputs	Number of neurons in HL**	Number of parameter
1; 1	2	2	0	0	1	5
1; 2	3	2	0	0	1	5
1; 3	4	3	0	0	1	6
1; 4	5	4	0	0	1	7
1; 5	6	4	0	0	1	7
1; 6	7	2	0	0	1	5
2; 1	4	2	0	0	1	5
2; 2	6	2	0	0	1	5
2; 3	8	3	0	0	1	6
2; 4	10	4	0	0	1	7
2; 5	12	5	0	0	1	8
2; 6	14	4	0	0	1	7

Note : * FM = Full model
**HL = Hidden Layer.

The fourth step is to select the lags from wavelet and scaling coefficients as inputs. The first method, inputs are the lags of scale and wavelet coefficients similar to that proposed by Renaud et al. [8] as in Equation (7). The second method, the inputs accommodate some appropriate lags addition (i.e. seasonal and near-seasonal lags) of scaling and wavelet coefficients besides lags input selection that proposed by Renaud et al. [8] as in Equation (8). The fifth step, the result of Terasvirta test shows that nonlinearity relationship between inputs-output of the WNN model is significant.

The sixth step is to determine the optimum WNN model at each j^{th} level. For the first method, the result of Wald test is shown in Table 1 at 3rd column and the number of neurons in the input layer based on Wald test is shown in Table 2 at 3rd column. The number of neurons in the hidden layer based on AIC is shown in Table 2 at 6th column. For the first method, the WNN model at 4th level-MAR order 2 has 10 inputs (before Wald test), 4 inputs (after Wald test), i.e. $w_{1(t-3)}$, $w_{3(t-1)}$, $w_{1(t-1)}$, $v_{4(t-17)}$, 1 neuron in the hidden layer, 7 parameters, with AIC, R^2, RMSE at training and testing data are 971.2984, 0.4136, 1.71890 and 2.67350, respectively.

Table 3. The result of significant-inputs selection by using the 2nd method.

MAR; Level	Significant inputs of reduced model	AIC	R Square	RMSE-training	RMSE-testing
1; 1	$w_{1(t-24)}$, $v_{1(t-24)}$	417.2512	0.7373	1.14989	1.45397
1; 2	$\boldsymbol{w_{1(t-24)}, w_{1(t-11)}, w_{2(t-11)}, v_{2(t-11)}}$	**303.2165**	**0.8134**	**0.96903**	**1.44103**
1; 3	$w_{1(t-24)}$, $w_{1(t-11)}$, $w_{2(t-11)}$, $w_{3(t-11)}$, $v_{3(t-11)}$	305.1799	0.8134	0.96897	1.44104
1; 4	$w_{1(t-24)}$, $w_{1(t-11)}$, $w_{2(t-11)}$, $w_{3(t-11)}$, $w_{4(t-11)}$	311.6957	0.8092	0.97982	1.44198
1; 5	$w_{1(t-24)}$, $w_{1(t-11)}$, $w_{2(t-11)}$, $w_{3(t-11)}$, $w_{4(t-11)}$	311.6957	0.8092	0.97982	1.44198
1; 6	$w_{1(t-24)}$, $w_{1(t-11)}$, $w_{2(t-11)}$, $w_{3(t-11)}$, $w_{4(t-11)}$	311.6957	0.8092	0.97982	1.44198
2; 1	$w_{1(t-24)}$, $v_{1(t-24)}$	417.2512	0.7373	1.14989	1.45397
2; 2	$\boldsymbol{w_{1(t-24)}, w_{1(t-11)}, w_{2(t-11)}, v_{2(t-11)}}$	**303.2165**	**0.8134**	**0.96903**	**1.44103**
2; 3	$w_{1(t-24)}$, $w_{1(t-11)}$, $w_{2(t-11)}$, $w_{3(t-11)}$, $v_{3(t-11)}$	305.1799	0.8134	0.96897	1.44104
2; 4	$w_{1(t-24)}$, $w_{1(t-11)}$, $w_{2(t-11)}$, $w_{3(t-11)}$, $w_{4(t-11)}$	311.6957	0.8092	0.97982	1.44198
2; 5	$w_{1(t-24)}$, $w_{2(t-12)}$, $w_{1(t-12)}$, $w_{1(t-11)}$, $w_{3(t-12)}$, $w_{4(t-12)}$	267.2713	0.8343	0.92394	1.44402
2; 6	$w_{1(t-24)}$, $w_{2(t-24)}$, $w_{5(t-24)}$, $w_{2(t-23)}$	293.7791	0.7629	1.10775	1.96514

Table 4. Comparison of various WNN models by using the 2nd method.

MAR; Level	Number of inputs from FM*	Reduced Model				
		Number of inputs	Number of seasonal lag inputs	Number of near-seasonal lag inputs	Number of neurons in HL**	Number of parameter
1; 1	14	2	2	0	1	5
1; 2	21	4	1	3	1	7
1; 3	28	5	1	4	1	8
1; 4	35	5	1	4	1	8
1; 5	42	5	1	4	1	8
1; 6	49	5	1	4	1	8
2; 1	16	2	2	0	1	5
2; 2	24	4	1	3	1	7
2; 3	32	5	1	4	1	8
2; 4	40	5	1	4	1	8
2; 5	48	6	5	1	1	9
2; 6	56	4	3	1	1	7

Note : * FM = Full model
**HL = Hidden Layer.

For the second method, the result of Wald test is shown in Table 3 at 3rd column and the number of neurons in the input layer based on Wald test is shown in Table 4 at 3rd column. The number of neurons in the hidden layer based on AIC is shown in Table 4 at 6th column. For the second method, the WNN model at 5th level-MAR order 1 has 42 inputs (before Wald test), 5 inputs (after Wald test), i.e. $w_{1(t-24)}$, $w_{1(t-11)}$, $w_{2(t-11)}$, $w_{3(t-11)}$, $w_{4(t-11)}$, 1 seasonal lag input, 4 near-seasonal lag inputs, 1 neuron in the hidden layer, 8 parameters, with AIC, R^2, RMSE at training and testing data are 311.6957, 0.8092, 0.97982 and 1.44198, respectively.

Finally, the optimum WNN model architecture is determined based on the smallest RMSE value from the testing data. Moreover, these two methods show that the optimum WNN model architecture is WNN at the 2nd level-MAR and order 1 or 2 with RMSE equal to 1.44103.

4. CONCLUSION

Based on the above results, it can be concluded that:

1. The optimum WNN model architecture is yielded based on the extended WNN model architecture that accommodates the seasonal and near-seasonal lags of the wavelet and scaling coefficients as input units.
2. The optimum WNN model architecture lies at 2nd level-MAR order 1 or order 2, have 4 inputs, i.e. $w_{1(t-24)}$, $w_{1(t-11)}$, $w_{2(t-11)}$, $v_{2(t-11)}$ (all of inputs are seasonal and near-seasonal lag inputs), 1 neuron in the hidden layer, 7 parameters, with AIC, R^2,

RMSE at training and testing data are 303.2165, 0.8134, 0.96903 and 1.44103, respectively.

5. REFERENCES

[1] Daubechies, I., 1992, Ten Lectures on Wavelets. 1st Ed., SIAM: Society for Industrial and Applied Mathematics, USA.

[2] Faraway, J. and Chatfield, C., 1998, "Time Series Forecasting with Neural Network: A Comparative Study Using the Airline Data," Applied Statistics, 47, pp. 231–250.

[3] Günther, F and Fritsch, S., 2010, "neuralnet: Training of Neural Networks," The R Journal, 2(1), pp. 30-38.

[4] Mallat, S. G., 1989, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," IEEE Trans, Pattern Anal. Machine Intelligence, 11, pp. 674–693.

[5] Murtagh, F., Starck, J.L. and Renaud, O., 2004, "On Neuro-Wavelet Modeling," Dec. Support System, 37, pp. 475-484.

[6] Nason, G.P. and Sachs., R. V., 1999, "Wavelets in Time Series Analysis," Phil. Trans. R. Soc. London, A, 357, pp. 2511-2526.

[7] Percival, D.B and Walden, A.T., 2000, Wavelets Methods for Time Series Analysis, Cambridge: Cambridge University Press, Chap. 4-5.

[8] Renaud, O., Stark, J.L. and Murtagh, F., 2003, "Prediction Based on a Multiscale Decomposition," Int. Journal of Wavelets, Multiresolution and Information Processing, 1, 217-232.

[9] Riedmiller, M. and Braun, H., 1993,"A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm," Proceedings of the IEEE International Conference on Neural Networks (ICNN), pp. 586-591.

[10] Shin, T. and Han, I., 2000, "Optimal Signal Multi-Resolution by Genetic Algorithms to Support Artificial Neural Networks for Exchange-Rate Forecasting," Expert Systems with Applications, 18, pp. 257–269.

[11] Suhartono, Ulama, B.S.S. and Endharta, A.J.,2010, "Seasonal Time Series Data Forecasting by Using Neural Networks Multiscale Autoregressive Model," American Journal of Applied Sciences, 7 (10), pp.1372-1378.

[12] Teraesvirta, T., Lin, C. F. and Granger, C. W. J., 1993, "Power of the Neural Network Linearity Test," Journal of Time Series Analysis, 14, pp, 209-220.