



Quickpropagation Architecture Optimization Based on Input Pattern for Exchange Rate Prediction from Rupiah to US Dollar

Harits Farras Zulkarnaen¹, Sukmawati Nur Endah²

^{1,2}Department of Informatics, Universitas Diponegoro, Indonesia
Email: ¹ harits.farras.z@gmail.com, ²sukma_ne@undip.ac.id

Abstract

Money exchange between countries was done by using exchange rates. One of the examples was the exchange between Rupiah and US Dollar. Exchange rates prediction to US Dollar was an attempt to assist all related economic actors to avoid losses during the process of decision making. The prediction could be done by using artificial neural network method. Quickpropagation was one of artificial neural network models considered suitable for prediction. Quickpropagation network architecture consisted of input layer, hidden layer, and output layer. The input layer of quickpropagation architecture could be determined by using autoregression (AR) for the input pattern. In this research, the authors aim to optimize the quickpropagation network architecture method using Nguyen-Widrow weight initialization to predict the Rupiah exchange rate to US Dollar. The research data were the exchange rate from the BI website from May 2017 to July 2017 with a total of 57 data. The test was performed by using K-Fold Cross Validation with $k = 11$ values for data without AR and $k = 8$ for AR data. The results show that quickpropagation method using AR has better performance than quickpropagation method without AR in terms of MSE training and testing. The best parameters are in alpha 0,6 and hidden neuron 5, with MSE training value 0,03272 and MSE testing 0,02873 for selling rate and at alpha 0,9 and hidden neuron 5, with MSE training value 0,03297 and MSE testing 0,02828 for buying rate with maximal epoch 100.000 and target error 0,05.

Keywords: Predicted Rupiah Exchange Rate, Selling Rate, Buying Rate, Quickpropagation, Autoregression (AR)

1. INTRODUCTION

Every country definitely has a tool for trading which is called money. Each of them has their own currency value. Exchange rate is needed so that money exchange between countries can be done. The differences in economic conditions of each country causes some impacts on the value of a country's currency to another. The differences can be either an increase in the exchange rate or a decline in the exchange rate value based on the state of a country's economics condition. One example is the difference in the Rupiah exchange rate to the US Dollar. The US Dollar is the official currency used in the United States. The US Dollar is also widely used internationally as a reserve currency outside the United States. The huge exchange rate of the Rupiah to US Dollar resulted in considerable material losses. This is due to lack of knowledge of economic actors. Therefore, it is important for economic actors to know the latest information

about the exchange rate, either the selling rate, the buying rate, or the middle rate in order to anticipate the movement of the Rupiah to US Dollar.

Prediction on currency rates is necessary in order to know the estimated value of the currency in the future, which keeps changing every day. So once the predicted results are obtained, economic actors can take strategic decisions and steps that should be taken in order to avoid losses. Prediction can be calculated using various methods; one of them is artificial neural networks. Artificial neural networks are a collection of interconnected elements of unit or node processing whose functionality is based on animal neurons. Network processing capabilities are stored in the value of connections between units called weights which is obtained through the adaptation process or the learning process from a set of training patterns [3].

One model of artificial neural network that can be used for prediction is quickpropagation. This method is comparatively better than other equivalent methods such as backpropagation where the quickpropagation method can perform calculations that provide fewer epoch than backpropagation method [7]. Furthermore, the other advantage of quickpropagation is the high percentage of accuracy rate produced compared with the results of backpropagation [5]. Quickpropagation has a weight change calculation using only local information on each weight, namely using the previous derivative error value and the current derivative error value, as well as the weight changes in the previous epoch [1].

The quickpropagation neural network has three layers: the input layer, the hidden layer, and the output layer. In the input layer, there is an approach using the analysis of the autoregression model (AR) to determine the number of inputs used in artificial neural networks. The autoregression model (AR) describes the time series value which has a relationship with the previous value and the error in a period determined from the amount of data available. AR model is formed based on the number of periods in the previous time. The use of AR for determining the number of inputs on artificial neural networks can be used for forecasting with time series data form [2].

Related to this research, there are many journals and studies that refer to the case study of “Quickpropagation Architecture Optimization Based on Input Pattern Using Autoregression for Exchange Rate Prediction from Rupiah to US Dollar”. One of them is the research conducted by Siana Halim (2000) with the title of “Application of Artificial Neural Network for Forecasting of Rupiah, Yen, Singapore Dollar, and Baht to US Dollar with Backpropagation”. The research of the prediction system model of Rupiah exchange rate to US Dollar by Arief Budiman Hutauruk (2010) by using quickpropagation resulted that quickpropagation has higher accuracy value than backpropagation. In addition to the quickpropagation and backpropagation, there is another approach that should be discussed, namely determining the input pattern using AR. So the research

proposal is to use the determination of input pattern with AR by using the prediction of quickpropagation neural network.

Thus, the proposed research aims to optimize the architecture of quickpropagation by using AR based on the input pattern. This will be compared to the quickpropagation architecture without using AR.

2. METHODS

The problem solving steps in this study are presented in the form of process blocks in Figure 1.

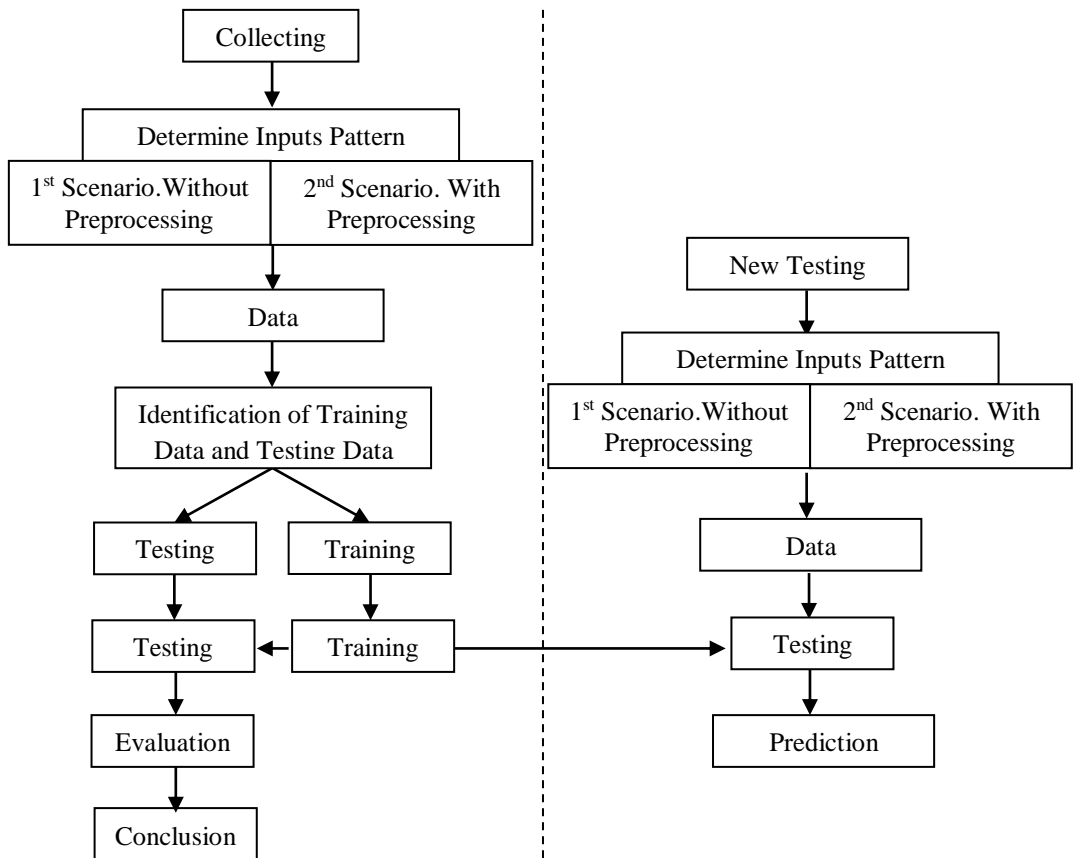


Figure 1. Methodology of process blocks

The explanation of each step of the process block steps in Figure 1 is presented in the next section.

2.1. Data Collection

Data collection for “Quickpropagation Architecture Optimization Based on Input Pattern Using Autoregression for Exchange Rate Prediction from Rupiah to US Dollar” is done by taking data from Bank Indonesia (BI) official website that is *www.bi.go.id*. The data taken are 58 data of selling rate and buying rate. This data was taken from May 2, 2017 to July 31, 2017.

2.2. Determination of Input Patterns

In the determination phase of data input pattern, data identification can be mapped in quickpropagation architecture. The process of determining the input pattern is divided into 2 scenarios. Scenario 1 is the determination of input pattern without using AR model based on the study of Novita A.S. (2010) that is converted to the time series by using the previous two days input. Scenario 2 is the determination of the input pattern using AR model and SPSS to find the AR model. Autoregression Model (AR) is a model where the current value of a set of times depends on the preceding values plus the random shock. So it can be seen that Z_t is regressed against p value of the previous Z . Autoregression model with order p , shortened as $AR(p)$ or $ARIMA(p, 0, 0)$ [9].

$$Z_t = \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \dots + \theta_p Z_{t-p} + a_t \quad (1)$$

Input patterns that have been obtained are then mapped into input units and output units so that they can be processed using quickpropagation architecture.

The data pattern to be established in scenario 1 consists of today's data and the previous day's data as inputs, and one target which is the data on the next day. The first layer is an input containing one bias and 2 input neurons. The second layer is a hidden layer consisting of one bias and hidden neurons as much as p . The third layer is the output layer or target with one output neuron. The number of neurons in the hidden layer is a variable that can be changed. Data input is in the form of time series data per day with data input pattern that is x_{t-1}, x_{t-2} . The resulting output is the value of x_t which is the value of x on the next day. Network architecture can be seen in Figure 2.

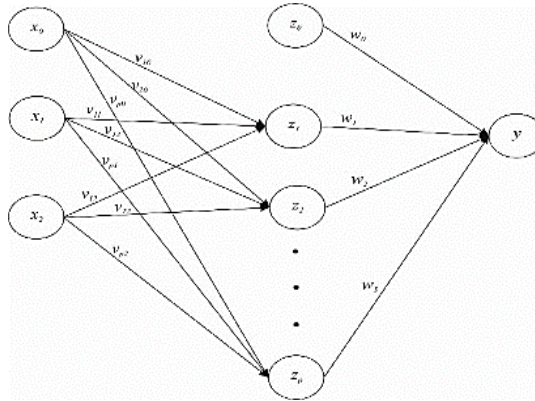


Figure 2. Quickpropagation architecture without AR

The data pattern to be formed in scenario 2 consists of today's data as input and a target that is the data on the next day. The first step to do is to plot the original data. Furthermore, it is imperative to conduct stationary test to the mean. If the prob value is less than 0.05, then the data is stationary to the mean. Then find the ACF (autocorrelation function) and PACF (partial autocorrelation function) graphs using SPSS and look at the PACF to determine the lag that affects the AR. The architecture formed is AR (1) or AR ordo 1 so that input data is in the form of time series data per day with data input pattern that is x_{t-1} . The resulting output is the value of x_t which is the value of x on the next day. Network architecture can be seen in Figure 3.

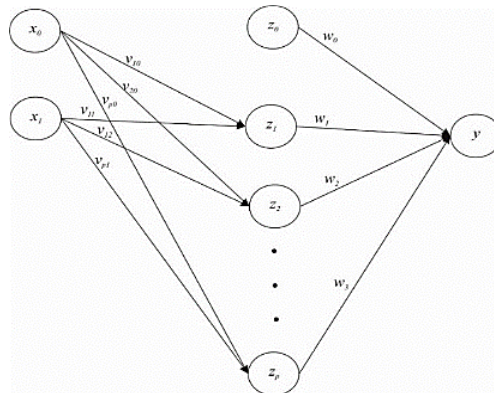


Figure 3. Quickpropagation architecture with AR

2.3. Data Normalization

Once the pattern is formed, the data will be normalized. Normalization of data is done so that all data values have the same range as the activation function used. The activation function used is the binary sigmoid activation function with range [0,1]. The equation for data normalization is as follows.

$$x' = \frac{0.8(x-a)}{b-a} + 0.1 \quad (2)$$

2.4. The Identification of the Training Data and the Test Data

The data will be divided into training data and test data. The training data are used in the training process, while the test data are used to validate models in the form of accurate testing of trained Quickpropagation. K-Fold Cross Validation is used to divide all data into training data and test data. K-Fold Cross Validation method is performed by dividing the dataset randomly into a subset [6]. K-Fold Cross Validation performs iterations of k times for training and testing data. Every iteration, one subset is used for testing while the remaining subset is used for training.

In this research, K-Fold is used with k = 11 value for scenario 1. Thus, all data is divided into 11 subsets of same size, then iteration is done. After that, the mean value of error from 11 iterations of MSE result is calculated. Furthermore, k = 8 is used for scenario 2 so that all data is divided into 8 subset of the same size, then 8 iterations are done. After that, the average value of error from 8 iterations of MSE result is calculated.

2.5. Training and Testing Process

This research uses quickpropagation training method. The quickpropagation algorithm is the development of a standard backpropagation algorithm based on the Newton method. All processes are the same as the standard backpropagation process, but each weight is stored in an equation of $\partial E / \partial w(t-1)$; the derivative error value is calculated during the previous epoch training in conjunction with the difference in the current weight value and the previous weighting value. In the quickpropagation algorithm, there are 3 phases comprising forward propagation phase, backward propagation phase, and weight changes.

The calculation of weight changes uses only local information on each weight, namely by using the derivative value of the previous error and the current derivative error value, as well as the weight changes in the previous epoch [1]. The explanation of the three phases of quickpropagation is as follows [8].

Step 0 : Initialize all weights with small random numbers, $epoch = 1$, determine the learning rate (α), specify the number of units on the hidden layer (p), and specify the termination conditions. The termination conditions are the maximum value of max epoch and target error.

Step 1 : If $epoch \neq \max epoch$ and the target error $< MSE$, do step 2 - 9.

Step 2 : For each pair of training data (1 to a where a is the amount of training data), do steps 3 - 8.

Phase I : Forward Propagation

Step 3 : Each input unit receives the signal and passes it to the hidden unit above it.

Step 4 : Calculate all outputs in the hidden layer z_j ($j = 1, 2, \dots, p$)

$$z_net_j = v_{j0} + \sum_{i=1}^n x_i v_{ji} \quad (3)$$

$$z_j = f(z_net_j) = \frac{1}{1 + e^{-z_net_j}} \quad (4)$$

Step 5 : Calculate all outputs networks in the unit y_k ($k = 1, 2, \dots, m$)

$$y_net_k = w_{k0} + \sum_{j=1}^p z_j w_{kj} \quad (5)$$

$$y_k = f(y_net_k) = \frac{1}{1 + e^{-y_net_k}} \quad (6)$$

Phase II : Backward Propagation

Step 6 : Calculate the derrivative error weight in

$$\frac{\partial E}{\partial w_0} = p = (y - t)y(1 - y) \quad (7)$$

$$\frac{\partial E}{\partial w_0} = p \cdot z_1 = (y - t)(y)(1 - y)(z_1) \quad (8)$$

Step 7 : Calculate the derrivative error weight in y . Do this process for all of the data set in one epoch.

$$\frac{\partial E}{\partial v_{j0}} = q_j = p \cdot w_j \cdot z_j(1 - z_j) \quad (9)$$

$$\frac{\partial E}{\partial v_{ji}} = q_j \cdot x_i = p \cdot w_j \cdot z_j(1 - z_j)x_i \quad (10)$$

Step 8 : Calculate the total of derrivative error in one epoch

$$d_m = \sum_{n=1}^N \left[\frac{\partial E}{\partial w_m} \right]_n \quad (11)$$

$$d_m = \sum_{n=1}^N \left[\frac{\partial E}{\partial v_m} \right]_n \quad (12)$$

Step 9 : Calculate each of weight changes

$$\Delta w_j^m = \frac{d_m w_j}{d_{m-1} w_j - d_m w_j} \cdot \Delta w_j^{m-1} \quad (13)$$

$$\Delta v_{ji}^m = \frac{d_m v_{ji}}{d_{m-1} v_{ji} - d_m v_{ji}} \cdot \Delta v_{ji}^{m-1} \quad (14)$$

Step 10 : Calculate the weight changes in each weight at the first epoch.

$$\Delta w_j^m = -\alpha \cdot d_m w_j \quad (15)$$

$$\Delta v_{ji}^m = -\alpha \cdot d_m v_{ji} \quad (16)$$

Phase III : weight changes

Step 11 : Calculate all the weight changes

Changes in the weight leading to the output layer.

$$w_{kj}(\text{baru}) = w_{kj}(\text{sekarang}) + \Delta w_{kj} \quad (17)$$

$$k = 1, 2, \dots, m ; j = 0, 1, \dots, p$$

Changes in the weight leading to the hidden layer.

$$v_{kj}(\text{baru}) = v_{kj}(\text{sekarang}) + \Delta v_{ji} \quad (18)$$

$$j = 1, 2, \dots, p ; i = 0, 1, \dots, n$$

Step 12 : Update the epoch value

$$epoch = epoch + 1$$

and calculate the mean squared error

$$MSE = \frac{\sum_{k=1}^a (t_k - y_k)^2}{a} \quad (19)$$

After obtaining the final weights of the training process, the final weights are used in the testing process to test the accuracy of the network to be used for prediction. The calculations on the testing process are only up to forward propagation stage and the accuracy value is calculated with MSE.

3. RESULTS AND DISCUSSION

The experimental results were conducted using 3 scenarios. The purpose of scenario 1 and 2 is to find the optimal architecture of each method of quickpropagation in each scenario. The optimal architecture of each scenario is determined by the value of MSE resulting from neural network testing procedures, where the optimal architecture is the architecture that produces the lowest MSE test value. Meanwhile, scenario 3 aims to compare the results of

scenario 1 and 2 based on the value of MSE training and MSE test scores. These values can be analyzed to get the best scenario and method from scenario 1 and 2 done in this research.

1st Scenario

1st Scenario aims to find optimal architecture in the process of predicting Rupiah to US Dollar using quickpropagation method without using preprocessing method i.e. autoregression (AR) with initial weighting built using Nguyen-Widrow. The analysis process is performed by using alpha or learning rate combinations from 0.1 to 0.9 with a rise of 0.1 and using a combination of the number of hidden neurons from 1 to 5. From a combination of alpha and hidden neuron parameters and using a target error of 0.05 and the epoch target 100.000, then the process is done as much as 45 training processes. The results from the scenario 1 are shown in Figure 4 and 5.

2nd Scenario

2nd Scenario aims to find optimal architecture in the process of predicting Rupiah to US Dollar using quickpropagation method with preprocessing method i.e. autoregression (AR) and initial weighting built using Nguyen-Widrow. The analysis process is performed by using alpha or learning rate combinations from 0.1 to 0.9 with a rise of 0.1 and using a combination of the number of hidden neurons from 1 to 5. From a combination of alpha and hidden neuron parameters and using a target error of 0.05 and the epoch target 100.000, then the process is done as much as 45 training processes. The results from the scenario 2 are shown in Figure 6 and 7.

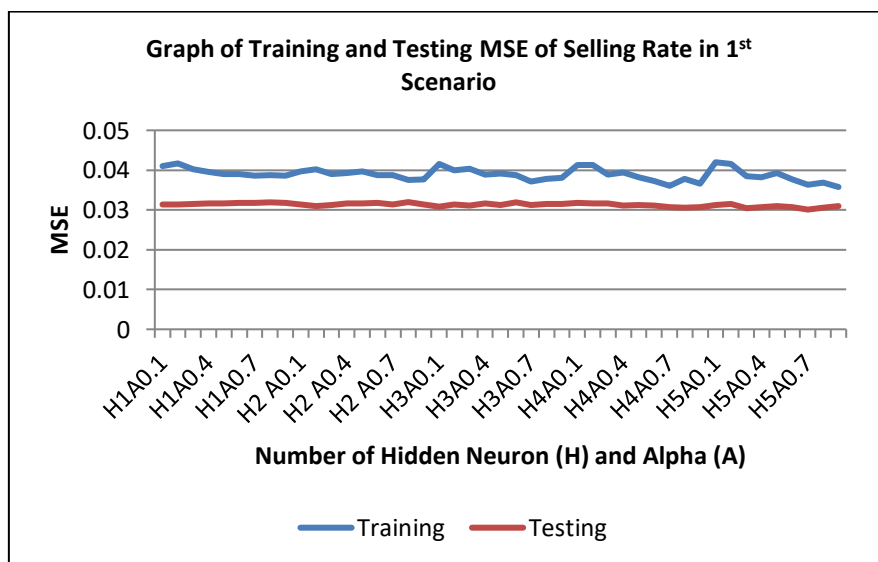


Figure 4. Graph of training and testing MSE of selling rate in 1st scenario

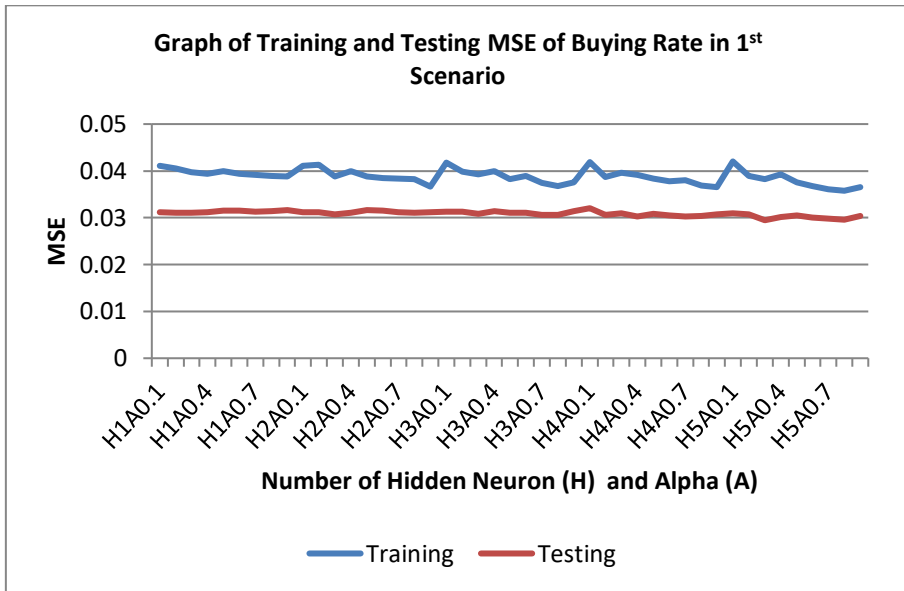


Figure 5. Graph of training and testing MSE of buying rate in 1st scenario

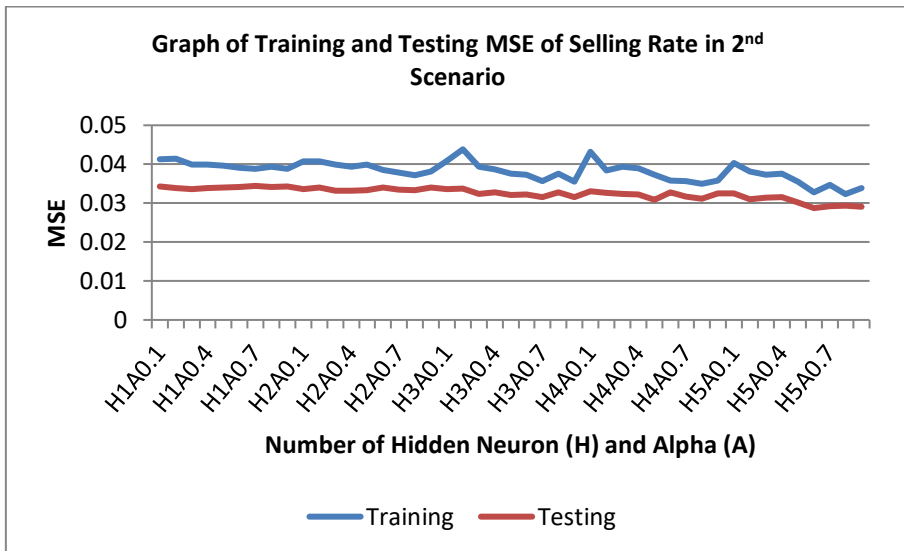


Figure 6. Graph of training and testing MSE of selling rate in 2nd scenario

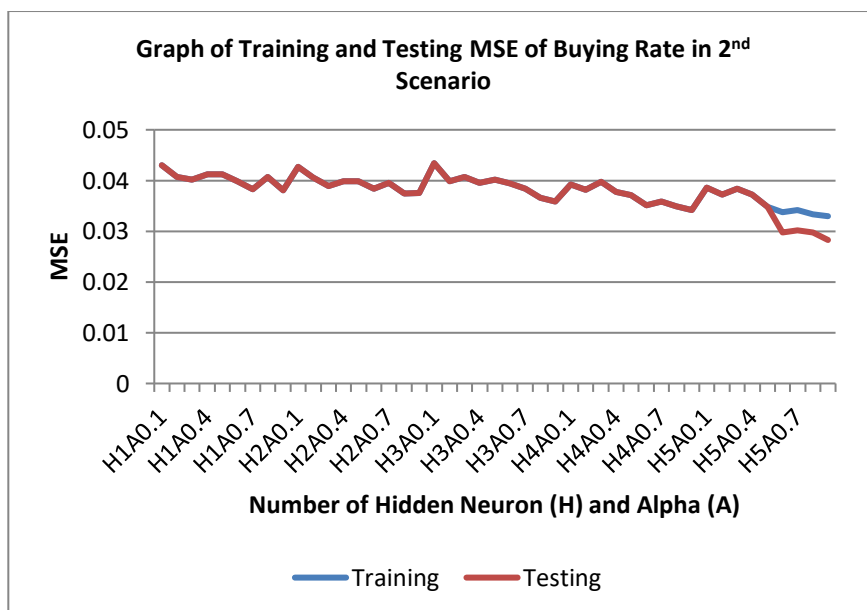


Figure 7. Graph of training and testing MSE of buying rate in 2nd scenario

3rd Scenario

3rd Scenario is an analytical scenario that aims to compare the research results between the prediction of Rupiah to US Dollar using 1st scenario and 2nd scenario. The comparisons are divided into two, namely the comparison of the training process with the MSE training indicator as well as the result of the testing process with the MSE testing. From 3rd scenario, it can be seen which scenario has the best performance in predicting Rupiah to US Dollar. The result of scenario 3 is shown in Figure 8 and Figure 9 for comparison of training, and Figure 10 and Figure 11 for comparison of testing.

In the best network architecture in scenario 1, the value of training MSE obtained is 0.03637 at the selling rate and 0.03827 at the buying rate. In the best network architecture in 2nd scenario, it produces training MSE of 0.03272 at the selling rate and 0.03297 at the buying rate. The 2nd scenario produces the lowest score of training MSE compared to the results of training MSE in 1st scenario seen from MSE results of the best architectural training. This shows that 2nd scenario is the best scenario compared to 1st scenario without AR when viewed from the value of training MSE.

In the best network architecture in scenario 1, the value of testing MSE obtained is 0.03008 at the selling rate and 0.02947 at the buying rate. In the best network architecture in 2nd scenario, it produces testing MSE of 0.02873 at the selling rate

and 0.02828 at the buying rate. The 2nd scenario produces a lower score of testing MSE compared to the results of testing MSE in 1st scenario seen from MSE results of the best architectural testing.

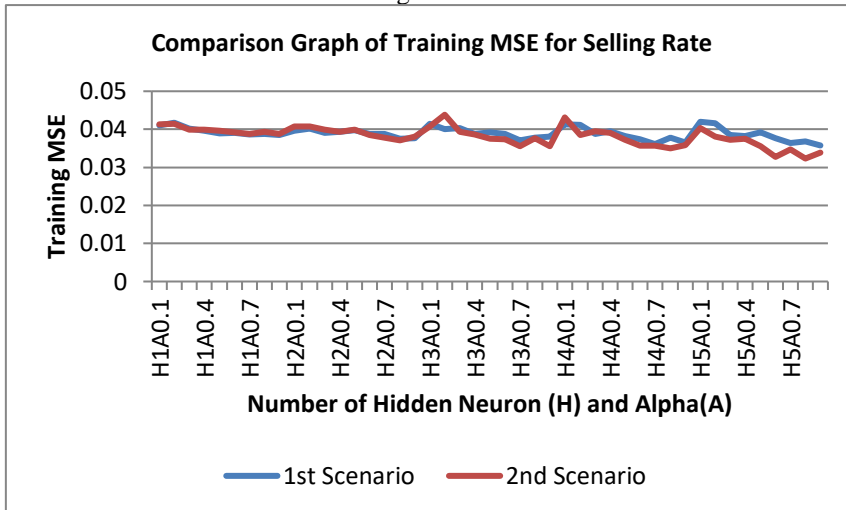


Figure 8. Comparison graph of training MSE for selling rate

So it can be concluded that 2nd scenario using quickpropagation method with Nguyen-Widrow initialization weight and using preprocessing AR resulted in the lowest testing MSE and is the best scenario compared to 1st scenario without using AR reviewed from testing MSE.

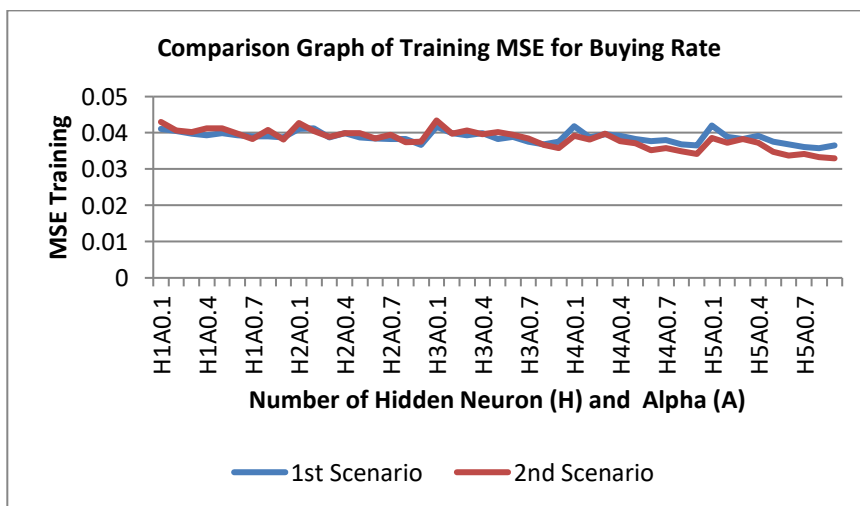


Figure 9. Comparison graph of training MSE for buying rate

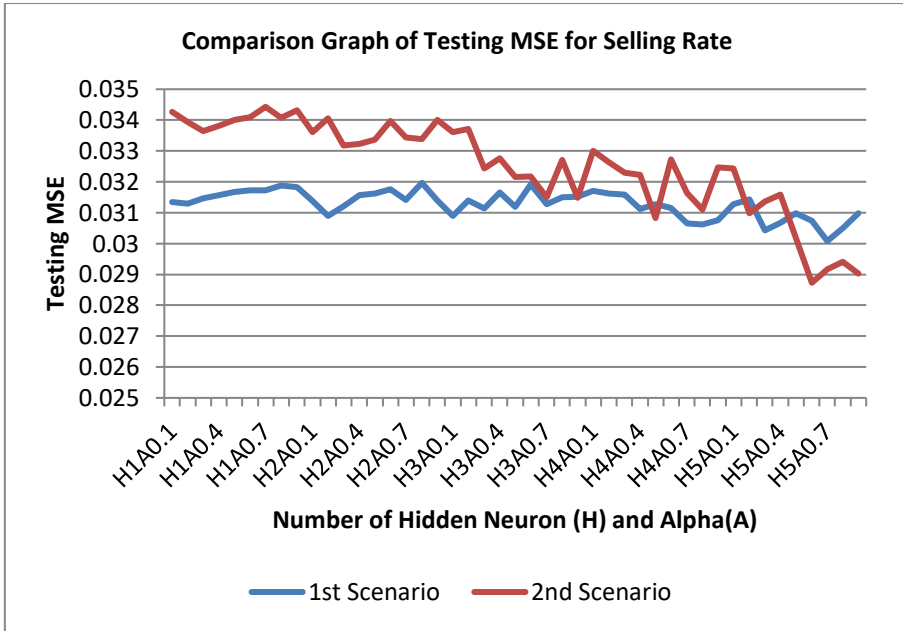


Figure 10. Comparison graph of testing MSE for selling rate

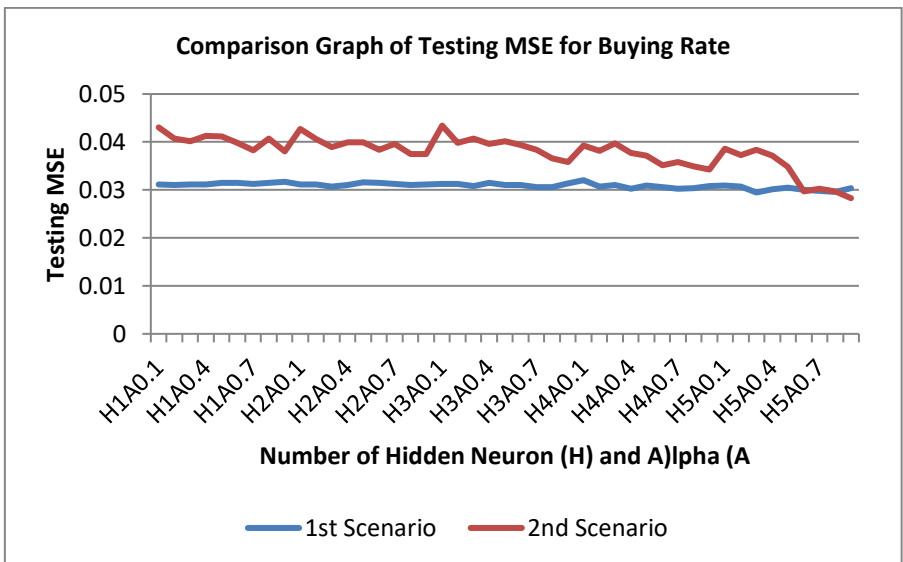


Figure 11. Comparison graph of testing MSE for buying rate

4. CONCLUSION

The conclusion that can be drawn from the research on "Quickpropagation Architecture Optimization Based on Input Pattern Using Autoregression For Exchange Rate Prediction of Rupiah to US Dollar" in terms of value of training MSE and testing MSE is that quickpropagation method using Nguyen-Widrow weighing initialization and preprocessing AR is the best method to do prediction. The best network architecture of quickpropagation method using initialization of Nguyen-Widrow weight and preprocessing AR is at alpha of 0.6 and hidden neuron as much as 5, with value of training MSE of 0.03272 and testing MSE of 0.02873 for selling rate, and at alpha of 0.9 and hidden neuron as much as 5, with value of training MSE of 0.03297 and testing MSE of 0.02828 for buying rate.

5. REFERENCES

- [1] Fahlman, S. E. (1988). An Empirical Study of Learning Speed in Back-Propagation Networks, United States: Carneige Mellon University. Gurney, K., 1997. *An Introduction to Neural Networks*. London: UCL Press Limited.
- [2] Filik, U. B. & Kurban, M. (2007). A New Approach for the Short-Term Load Forecasting with Autoregressive and Artificial Neural Network Models. *International Journal of Computational Intelligence Research*, 3 (1): 66-71.
- [3] Gurney, K. (2003). *An Introduction to Neural Networks*. London: UCL Press.
- [4] Harinowo, Cyrillus. (2007). *Manajemen Aktiva Pasiva Bank Devisa*. Jakarta: Grasindo.
- [5] Hutauruk, A. B., Jondri & Rismala, R. (2010). Perancangan Model Sistem Prediksi Nilai Tukar Rupiah terhadap Dollar Amerika. *eProceedings of Engineering*, 1(1): 764-770.
- [6] Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *International Joint Conferences on Artificial Intelligence*, 14 (2): 1137-1145.
- [7] Mirzadeh, H. & Najafizadeh, A. (2008). Correlation between processing parameters and strain-induced martensitic transformation in cold worked AISI 301 stainless steel. *Material's characterization*, 59: 1650-1654.
- [8] Jek, S.J.. (2005). *Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan Matlab*. Yogyakarta: ANDI
- [9] Samarasinghe, S. (2006). *Neural Networks for Applied Sciences and Engineering*. New York: Auerbach Publications.