Applying Artificial Neural Networks in Forecasting US Dollars-Indonesian Rupiah Exchange

Soleh Ardiansyah
Faculty of Computer System and Software Engineering
Universiti Malaysia Pahang
Gambang Kuantan, Pahang, Malaysia
ardiansyah.soleh@gmail.com

Mazlina Abdul Majid
Faculty of Computer System and Software Engineering
Universiti Malaysia Pahang
Gambang Kuantan, Pahang, Malaysia
mazlina@ump.edu.my

***Abstract*—This paper investigates artificial neural networks prediction modeling of foreign currency rates using Levenberg Marquardt (LM) learning algorithms. The models were trained from historical data using US Dollar (USD) currency rates against Indonesian Rupiah (IDR). The forecasting performance of the models was evaluated using a number of statistical measurements and compared. The results show that significant close prediction result can be made using simple architecture forecasting model. LM1 and LM6 model achieves closer prediction of the actual value than that other model. Both forecasting models attain significantly high rate of predicting correct directional change (above 80%). The effect of network architecture on the performance of the forecasting model is also presented.**

***Index Terms*— Neural network, forecasting, foreign exchange.**

Introduction

The foreign currency exchange rates take an important part in compulsive of the dynamics of the currency market [1], since it is the largest financial market in the world and becomes very profitable market with more than US$ 3.0 trillion of daily transaction [2]. That the reason why the appropriate prediction of currency exchange rate is a crucial factor for the success of many businesses, although the market is well known for its volatility and unpredictable. Theoretical models including both econometric and time series approaches have been widely used to model and forecast exchange rates such as ARCH, GARCH, autoregressive models, and chaotic dynamics applied to financial forecasting. However, linear indicator has always worked well on a linear movement but stops helplessly when dealing with nonlinear behavior of the market [3].

As decision-making tools, Artificial Neural Networks is well-known function approximates in prediction and system modeling, has recently shown its great applicability in time series analysis and forecasting. Neural network are able to learn pattern and relationship from the data itself, unlike the other techniques that construct functional form to represent relationship of the data. Artificial Neural Networks support multivariate analysis. Multivariate models can rely on greater information's, there is not only the lagged time series being forecast, but also other indicators (such as fundamental, technical, inter-marker etc). In addition, ANN is more effective in describing the dynamics of non-stationary time series since its unique adaptive properties, non-assumable, non-parametric, and noise-tolerant. As universal function approximates, ANN also can map any nonlinear function without a priori assumptions about the data [4]. Artificial Neural Network model for forecasting exchange rates have been investigated in a number of studies have found that neural networks are better than random walk models in predicting the Deutsche mark/US dollar (DEM/USD) exchange rate [5]. In comparison with the traditional forecasting methods such as Box-Jenkins ARIMA models or regression models, [6] there are many more modeling factors to be considered in neural networks [7]. In this paper, we will examine number of hidden nodes on the forecasting performance of the exchange rate between the US Dollar (USD)-Indonesia Rupiah (IDR).

The rest of this paper is organized as follow. In the section 2, the forecasting models and the review of related literature are summarized. The methodology and the simulation result of the model are described in section 3. Finally, conclusions are given in section 4.

Neural Network Forecasting Model

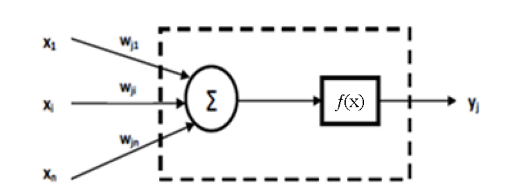
Neural network architecture

Artificial neural networks are massively parallel adaptive networks of simple nonlinear computing element called neuron which is intended to abstract and model some of the functionality of the human nervous system in the attempt to partially capture some of its computational strengths. Artificial neural network (ANN) is a type of an Artificial Intelligence technique that mimics the behavior of the human brain [9]. ANN can model linear and non-linear systems without need to make assumptions implicitly as in most traditional statistical approaches. ANNs can be grouped into feed-forward and feedback (recurrent) networks. In the former network, no loops are formed by the network connections, while one or more loops may exist in the latter. The most commonly used family of feed-forward networks is a layered network in which neurons are organized into layers with connections strictly in one direction from one layer to another [8].

MLPs are the most common type of feed-forward networks. MLP which has three types of layers: an input layer, an output layer and a hidden layer. Neurons in input layer only act as buffers for distributing the input signals x_i (i=1, 2, …,n) to neurons in the hidden layer. Each neuron j as shown in Figure1 in the hidden layer sums up its input signals x_i after weighting them with the strengths of the respective connections w_{ji} from the input layer and computes its output y_j as a function f of the sum.

$$y_j = f\left(\sum_{i=1}^{n} w_{ji}\, x_i\right) \tag{1}$$

f can be a simple threshold function or a sigmoidal, hyperbolic tangent or radial basis function.



ANN process

The output of neurons in the output layer is computed similarly. The backpropagation algorithm is the most commonly adopted MLP training algorithm. It gives the change Δw_{ji} the weight of a connection between neurons i and j as follows:

$$\Delta w_{ij} = \alpha\, \delta_j x_i \tag{2}$$

Where α is a parameter called the learning rate and δ_j is a factor depending on whether neuron j is an input neuron or a hidden neuron. For output neurons,

$$\delta_j = (\delta f / \delta net_j)\left(y_j^{(t)} - y_j\right) \tag{3}$$

and for hidden neurons

$$\delta_j = (\delta f / \delta net_j)\left(\sum_q w_{jq}\, \delta_q\right) \tag{4}$$

In Eq. (3), net j is the total weighted sum of input signals to neurons j and $y_j(t)$ is the target output for neuron j. As there are no target outputs for hidden neurons, in Eq. (4), the difference between the target and actual output of a hidden neurons j is replaced by the weighted sum of the δ_q terms already obtained for neurons q connected to the output of j. The process begins with the output layer, the δ term is computed for neurons in all layers and weight updates determined for all connections, iteratively. The weight updating process can happen after the presentation of each training data (data-based training) or after the presentation of the whole set of training (batch training). Training epoch is completed when all training patterns have been presented once to the MLP.

A commonly adopted method to speed up the training is to add a “momentum” term to Eq. (5) which effectively lets the previous weight change influence the new weight change:

$$\Delta w_{ij}(I+1) = \alpha\, \delta_j x_i + \mu\, \Delta w_{ij}(I) \tag{5}$$

Where Δw_{ij} (i+1) and Δw_{ij} (i) are weight changes in epochs (i+1) and (i), respectively, and μ is “momentum” coefficient [10].

Data collection and pre-processing

The data used in this study is the foreign exchange rate of US Dollar against Indonesian rupiah (IDR) from April 2010 to January 2013downloaded from Meta Trader Applications (see Figure 2). The data sets are divided into three sets, training, validation, and testing datasets by 70/15/15 principle where 70% of the data are used as a training datasets, 15% of the data are used as a validation datasets and the rest 15% of the data are used as a testing datasets. Training datasets are presented to the network during training, and the network is adjusted according to its error. Validation datasets are used to measure network generalization, and to halt training when generalization stops improving. Testing datasets have no effect on training and so provide an independent measure of network performance during and after training.

Historical exchange rate of USD/IDR

After data collection, data preprocessing procedures are conducted to train the ANNsmore efficiently. These procedures are: (1) solve the problem of missing data, (2) normalize data. The missing data are replaced by the average of neighboringvalues during the same day. Normalization procedure before presenting the input data tothe network is generally a good practice, since mixing variables with large magnitudes andsmall magnitudes will confuse the learning algorithm on the importance of each variableand may force it to finally reject the variable with the smaller magnitude [12].

Performance Measurement

The selection of an error measure as a benchmark depends on each situation. By only using one error measure for evaluating the prediction performance, it does not show the behavior of the prediction in a clear way [13-14]. Therefore, the use of more than one performance measure should be considered to provide robust evaluation of the prediction result and to achieve desirable goals [15-16]. In order to evaluate the performance of the ANNs models quantitatively and verify whether there is any underlying trend in performance of ANNs models, this study used statistical analysis involving the coefficient of determination (R2), and the mean square error (MSE). Regression R Values measure the correlation between outputs and targets. An R value of 1 means a close relationship, 0 a random relationship. Mean Squared Error is the average squared difference between outputs and targets. Lower values are better. Zero means no error. This criterion is the most popular measure that is used to evaluate the prediction performance.

$$MSE = \frac{1}{N} \sum_{t=1}^{N} (y_t - \hat{y}_t)^2 \quad (6)$$

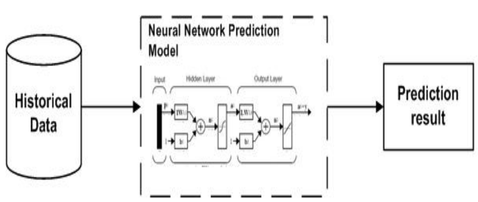
Where N is the number of prediction period; y_t is the actual value; and $\hat{y}_t$ is the predicted values.

Result and Discussion

Building the model

At this stage, the designer specifies the number of hidden layers, neurons in each layer transfer function in each layer, training function, weight/bias learning function, and performance function. In this paper, multilayer perceptron (MLP) and Levenberg Marquardt (LM) training algorithm are used to investigate the influence of neural network architecture on prediction performance. All of the models were trained with different number of hidden neuron to predict USD against IDR.

This study used 9 period of moving average as technical data. The advantage of moving average is its tendency to smooth out some of the irregularity that exits between market days [11]. In our model, we used moving average values of past 9 days to feed to the neural network to predict the next day rate. The neural network model has 6 inputs for six indicators, one hidden layer and one output unit to predict exchange rate. Yao et al. [11] has reported that increasing the number of inputs does not necessarily improve the performance. Figure 3 shows the forecasting model of ANN.



ANN Forecasting Model

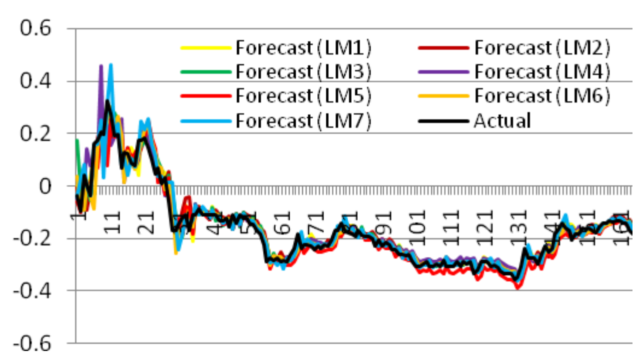
Simulation Results

A neural network model was trained with nine inputs representing the nine days period of USD/IDR exchange rates, a hidden layer and an output unit to predict the exchange rate. The final set of weights to which a network settles down (and hence its performance) depends on a number of factors, e.g., initial weights chosen, different learning parameters used during training and the number of hidden neuron. This study trained 7 models of neural networks with different architecture. The number of hidden units was variedbetween10~40 and the training were terminated at iteration number between 500-1000. The network that generated the best result of the trials in each architecture was presented in this section. This study measured the performance on the testing, validation, and testing data to investigate how well the neural network forecasting model captured the underlying trend of the movement of currency. The forecasting performance of the best trial is reported in Table 1. The result shows that Levenberg Marquardt (LM) training algorithm performs better in term of all performance measure in almost all of the models.

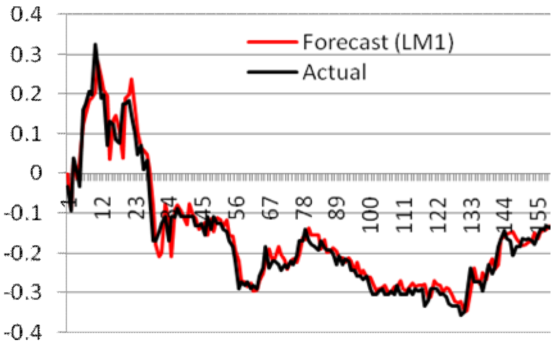
Table Type Styles

Model	MSE			R		
	Train	***Val.***	***Test***	***Train***	***Val.***	***Test***
LM1	1.89E-03	2.93E+02	4.06E-03	9.96E-01	9.95E-01	9.93E-01
LM2	1.20E-03	3.06E-03	6.82E-03	9.98E-01	9.94E-01	9.87E-01
LM3	1.36E-03	5.92E-03	7.75E-03	9.97E-01	9.88E-01	9.85E-01
LM4	1.78E-03	4.77E-03	5.20E-03	9.97E-01	9.91E-01	9.90E-01
LM5	1.48E-03	4.78E-03	8.51E-03	9.97E-01	9.89E-01	9.85E-01
LM6	1.66E-03	3.27E-03	3.90E-03	9.97E-01	9.94E-01	9.93E-01
LM7	1.06E-03	4.82E-03	7.85E-03	9.98E-01	9.91E-01	9.83E-01

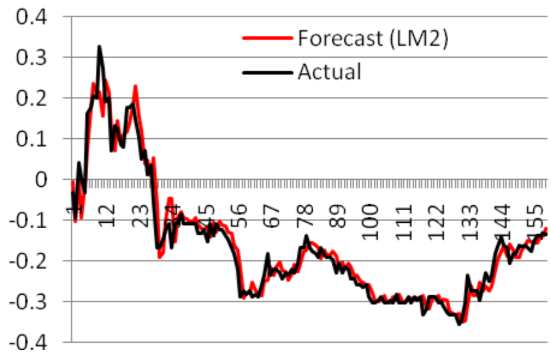
The comparative diagrams showing the output forecast by ANN model against actual time seriesover9 days prediction period for 7 models are shown in Figure 4(a)-(h). Fig 4(a) shows the forecasting of USD/IDR by all the models. The plots show that theLM1 and LM6 forecasting model more closely follows the actual rate. Both forecasting models attain significantly high rate of predicting correct directional change (above 80%). From figure 4(a), we can see that generally the network is trained in good condition. Using the trained network, we can obtain mean forecasting error of each models is 0.006101925, 0.000992487, 0.003275768, 0.00628393, 0.004087708, 0.001997213, 0.006321315 respectively.



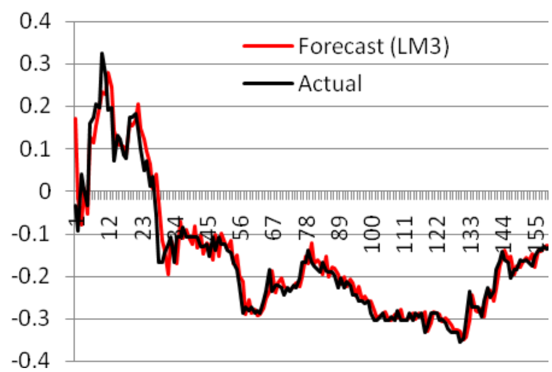
(a) Forecasting of USD/IDR by all the models against actual value



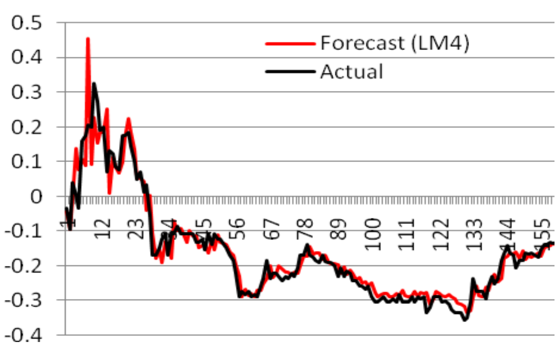
(b) Forecasting of USD/IDR by LM1 models against actual value



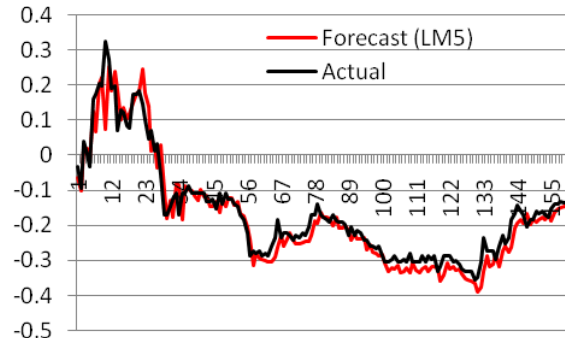
(c) Forecasting of USD/IDR by LM2 models against actual value



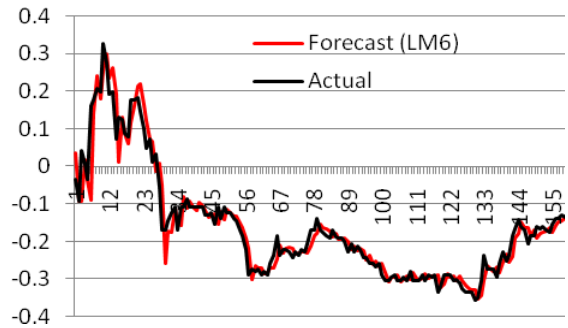
(d) Forecasting of USD/IDR by LM3 models against actual value



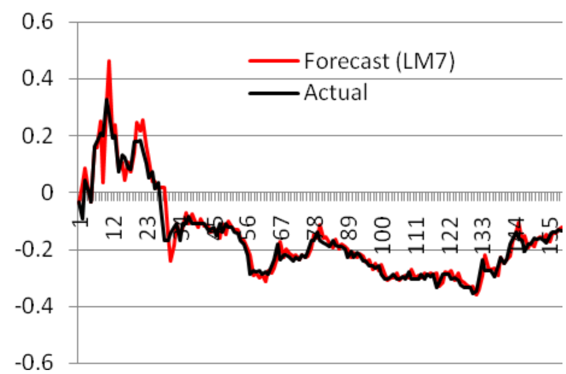
(e) Forecasting of USD/IDR by LM4 models against actual value



(f) Forecasting of USD/IDR by LM5 models against actual value



(g) Forecasting of USD/IDR by LM6 models against actual value



(h) Forecasting of USD/IDR by LM7 models against actual value

(a)-(h) Forecasting of different architecture of neural network model.

The ANNs is able to produce correct output in response to an unseen input pattern is influenced by a number of factors: 1) the size of the training dataset, 2) the network architecture, and 3) the complexity of the problem. Practically we have no control on the problem complexity, and in our simulation the size of the training set is fixed. This leaves the generalization ability, i.e., performance of the model dependent on the architecture of the corresponding network.

Conclusion and Future Works

This paper has presented and compared seven different neural network forecasting models to perform USD/IDR currency exchange forecasting. LM1 and LM6 model achieves closer prediction of the actual value than other models. As Medeiros et al. argues in [17], there are evidences that favor linear and nonlinear models against random walk, and nonlinear models stand a better chance when nonlinearity is spread in time series.

A neural network model with improved learning technique is thus a promising candidate for forex prediction. Results in this study show that LM neural network model achieves very close prediction on training phase in terms of MSE. The forecasting models attain significantly high rate of predicting correct directional change (above 80%). Many researchers argued that directional change of the metrics may be a better standard for determining the quality of forecasting.

Acknowledgment

This work was supported under the research grant No Vote GRS120339, Universiti Malaysia Pahang, Malaysia.

References

[1] Gencay R. (1999). Linear, non-linear, and essential foreign exchange rate prediction with simple technical trading rules. Journal of International Economics 47, 91–107.

[2] Soleh,A. and Mazlina, A.M. and Zain, J.M. (2012). Hybrid Neural Network and Decision Tree for Exchange Rates Forecasting. In the Proceeding of International Conference on Computational Science and Information Management (ICoCSIM), 29–35.

[3] M.L. Seliem. Foreign exchange forecasting using artificial neural network as a data mining tool. M. Sc. Thesis, University of Louisville, Kentucky, 2004.

[4] Cao L.andTay F. (2001). Financial Forecasting Using Support Vector Machines. Neural Computing & Application, 10, 184-192

[5] Weigend A.S., B.A. Huberman and Rumelhart D.E. (1992). Predicting sunspots and exchange rates with connectionist networks: In Nonlinear modeling and forecasting. Addison-Welsey, Redwood City, 395-432

[6] Wang J.H. and Leu J.Y. (1996). Stock Market Trend Prediction Using ARIMA-based Neural Networks. Proceeding of IEEE International Conference on Neural Networks, 4, 2160-2165.

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

[8] Jain, A., Mao, J., and Mohiuddin, K.(1996). Artificial Neural Networks: A Tutorial.IEEE Computer, 29(3,) 31–44.

[9] Haykin, S. (2009). Neural Networks and Learning Machines: 3rd edition. Pearson EducationInc.,New Jersey

[10] Jayawardena, A., Achela, D., and Fernando, K. (1998). Use of Radial Basis Function Type Artificial Neural Networks for Runoff Simulation. Computer-Aided Civil and Infrastructure Engineering, 13, 91–99.

[11] S.-C. Huang, P.-J. Chuang, C.-F. Wu, and H.-J. Lai.(2010).Chaos based support vector regressions for exchange rate forecasting. Expert Systems with Application, 37, 8590–8598.

[12] Tymvios, F., Michaelides, S. and Skouteli, C.(2008). Estimation of Surface solar radiation with artificial neural networks. InV. Badescu (Eds.): Modeling Solar Radiation at the Earth Surface,Springer, 221–256.

[13] M.P. Clements and D.F. Hendry.(1993).On the limitations of comparing mean square forecast errors. Journal of Forecasting, 12 (8), 617–637.

[14] J.S. Amstrong and F. Collopy.(1992).Error measures for generalizing about forecasting methods: Empirical comparison. International Journal of Forecasting 8, 62–80.

[15] L.J.A. Rodrigues, P.S.G. de MattosNeto, and T.A.E. Ferreira.(2009).A prime step in the time series forecasting with hybrid methods: the fitness function choice. In the Proceeding of International Joint Conference on Neural Networks (IJCNN), 2703–2710.

[16] L.J. Tashman.(2000).Out-of-sample tests of forecasting accuracy: analysis and review. International Journal of Forecasting, 16(4), 437–450.

[17] M.C. Medeiros, A. Veiga and C.E. Pedreira. (2001). Modeling exchange rates: smooth transitions, neural network and linear models. IEEE Transactions Neural Networks, 12(4), 755–764