Using MLP and RBF Neural Networks to Improve the Prediction of Exchange Rate Time Series with ARIMA

Arash Negahdari Kia, Mohammad Fathian, and M. R. Gholamian

Abstract—**In this paper , a new hybrid model for predicting the exchange rate time series is introduced, using the multilayer perceptron (MLP) and radial basis function (RBF) neural networks to reduce the error of autoregressive integrated moving average (ARIMA) method. The hybrid model tries to detect the error of the linear statistical method and then model this error with MLP neural network. Again the remainder error is modeled with RBF neural network to reduce the final error of the hybrid model. In this two level process of error modeling, it will be proved that the final result of prediction and modeling is better than the results that could be achieved by a single ARIMA method or a single MLP or RBF neural network.**

Index Terms—**ARIMA; multilayer perceptrons; radial basis functions; time series forecasting**

I. Introduction

The new exchange rate market was made on the ruins of the Bretton-Woods system in 1973. According to Bretton-Woods agreement the rate of foreign currencies was based on the resources of gold in each country was called a fixed rate system. But the new system of exchange rate is based on the supply and demand of each currency and the exchange rate is float and its movements are more complicated. The new system makes the exchange rate market very risky for the traders, giving them the chance of getting a lot of profit or too much loss [1]. This makes the necessity of methods for predicting the market and modeling the exchange rate movements more than before to reduce the risk for the traders. Predicting and modeling of exchange rate time series have gained the attention of many researchers and the usefulness of machine learning and artificial intelligence techniques like neural networks have been discussed in many works [1]-[6].

Financial time series have high non-linearity, high noise and high complexity and it makes them hard cases for being modeled and predicted [7].

Yao and Tan (2000) showed that a simple feed forward artificial neural network can be used for forecasting exchange rates time series and the results showed that it could be better than conventional methods like ARIMA. They said that financial data cannot be modeled using simple linear ARIMA based modeling methods and because of the high complexity of financial time series neural networks could be a good modeling method to use for these kinds of data [6]. White (1989) showed that we can use neural networks as a tool for helping conventional statistical methods to get better results in forecasting [8]. Refenes *et al.* (1984) said, we have reached our limit to achieve better results for modeling complex time series like financial time series with statistical methods [2]. The linear nature of most statistical methods is the weakness of them in modeling financial data. For capturing the non-linearity of financial time series we have to using other techniques like neural networks is necessary.

In this paper the researchers try to solve the non-linearity problem of exchange rate time series by using the help of two most important kinds of neural networks to capture the non-linearity that had not been found by the ARIMA model. The neural networks that are used are MLP (Multilayer Perceptron) and RBF (Radial Basis Function). MLP and RBF are used to catch the benefits of both in capturing non-linearity, for the prediction.

In section 2 of the paper the researchers briefly introduce ARIMA, MLP and RBF methods which are like the building bricks of the hybrid model and then in section 3 the hybrid model building process is described in detail. Section 4 talks about the evaluation methods in the literature of financial time series prediction. In this section, the evaluation method used in the research is introduced for proving that the model is useful and it works better than the single models. Section 5 describes the data preparation step of the research and how the dataset was made ready for the ARIMA and for the learning step of the neural networks. In section 6 the results of using the proposed model are shown and discussed and finally section 7 concludes and discusses further researches and future works.

II. A Brief Description of ARIMA, MLP, and RBF

In the literature, there are two types of meta-learning method: first using a simple model with different datasets as input and then ensembling the outputs to achieve better results. The second is using different types of modeling for a single dataset to use the benefits and potentialities of every single model in the hybrid model, again for getting better results in modeling and prediction [3]. In this research the second type of approach is used. Before going through the model building process it is necessary to describe ARIMA, MLP, and RBF in brief.

A. ARIMA (Auto-Regressive Integrated Moving Average)

Autoregressive integrated moving average is the most famous and important method in statistics for linearly modeling and predicting the time series. Three important factors are in ARIMA, as its name says: auto regression, integration, and moving average. Every ARIMA model is represented with the structure of ARIMA (p, d, q), where p is the auto regression factor, d is the integration factor, and q is the moving average factor [9]. The general ARIMA model is represented in (1):

Manuscript received April 10, 2012; revised May 22, 2012.
The authors are with Iran University of Science and Technology, School of Industrial Engineering, Iran (e-mail: arash.nkia@gmail.com).

$$P(B)(1-B)^d(X_t-\mu)=Q(B)e_t$$

$$P(B)=1-\sum_{i=1}^{p}\varphi_i B^i$$

$$Q(B)=1-\sum_{j=1}^{q}\theta_j B^j \quad (1)$$

In (1), B is called the backshift operator and $P(B)$ and $Q(B)$ are polynomial backshifts. $X(t)$ is the time series that we want it to be modelled. μ is the mean of the time series and φ and θ are the model parameters. At last, p, d, and q, as mentioned before, are the ARIMA factors.

B. MLP (Multilayer Perceptron)

Multilayer perceptron neural networks are a class of nonlinear models that have one or more than one inputs and some hidden layers that connect the inputs to one or more than one outputs in a nonlinear manner. The nonlinearity comes from the hidden layers of the neural network which can model very complex functions [10]. A simple single layer perceptron is stated as (2):

$$output=f(\sum_{i=0}^{n}w_i\Phi(x)) \quad (2)$$

In (2) x is the input vector given to the single layer perceptron and Φ is called the activation function. w_i is the weight of the edge i, in the neural network and f is the output function.

There are many activation functions for neural networks. The most famous ones are log-sigmoid and hyperbolic tangent. This research used log-sigmoid activation function for the multilayer perceptron neural network model as stated in (3):

$$f(a)=\frac{1}{1-e^{-a}} \quad (3)$$

In the multilayer perceptron every hidden layer can transfer its output to another hidden layer in an orderly manner. For example a two layer perceptron function could be expressed as follow [11].

$$output_k=f(\sum\nolimits_j w_{kj}^{(2)}g(\sum\nolimits_i w_{ij}^{(1)}x)) \quad (4)$$

In (4) k, is the number of outputs or elements in output vector. f is the output function. g is the activation function in hidden layer and $w^{(1)}$ and $w^{(2)}$ are the weight vectors for the first and the second hidden layer.

C. RBF (Radial Basis Function)

The RBF neural network is very different from other types of neural networks. It consists only of three layers, the input layer, hidden layer, and the output layer. Each node in the hidden layer calculates the distance between itself and the input and then a radial basis function changes the result. The final result of the radial basis function will be the output of the hidden layer and multiplies to a weight and then is fed to the output layer of RBF. In the output node all the inputs from the hidden layer are summed together and it yields the final output of the RBF neural network [5]. The calculation procedure of an RBF neural network is presented in (5).

$$output=bias+\sum_{i=1}^{n}w_i\Phi(d_i) \quad (5)$$

In (5) w_i is the weight of the edge i, which transfers the result of the hidden layer to the output layer and bias is a value which is added to the output at the end as an independent weight value. The function Φ is the radial basis function and d_i is the distance between the input and the center of the node in the hidden layer. For calculating the distance, we can use any kind of distance measurement like Euclidian distance as it is shown in (6). C_j, is the center of the node j in the hidden layer. There are some different kinds of radial basis function but in this research the Gaussian radial basis function is used as it is presented in (7).

$$d_i=\sqrt{\sum_{j=1}^{n}(x_j-C_{ij})^2} \quad (6)$$

$$\Phi(d_i)=e^{-d_i^2/r^2} \quad (7)$$

III. Model Building Process

The process of model building and the hybrid model is proposed by the researcher. First by using the ARIMA method for predicting the exchange rate time series, the difference between the real exchange rate, which is called *realex*, and the predicted exchange rate, which is called *arimaout*, will be found and defined as $error_1$,as it is presented in (8) and the process is shown in Fig. 1.

$$realex-arimaout=error_1 \quad (8)$$

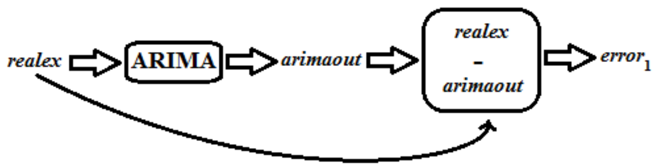

Fig. 1. Hybrid model's first block. Calculating the error of ARIMA in prediction.

Then $error_1$ will be modeled by MLP neural network and the result will be called *mlperror* as it is shown in Fig. 2.

Fig. 2. Hybrid model's second block. Using a multilayer perceptron to model the prediction error of ARIMA

It can be assumed that *mlperror* is a good approximation of the error of ARIMA model or $error_1$. This error comes from the nonlinearity and complexity that could not be modelled by ARIMA and the multilayer perceptron neural network tries to model it. So by (8) we can say that:

$$realex = arimaout + error_1 \quad (9)$$

As mentioned before we can replace the ARIMA error by the approximation of it from MLP modeler and it results us the equation (10).

$$realex \cong arimaout + mlperror \quad (10)$$

In the next stage of the proposed hybrid model we can assume that there will be still some errors left that could not be captured by the MLP modeler. In this stage the hybrid model tries to use the benefits of an RBF neural network to catch the reminder error that could not be modeled by MLP because of its different nature and structure from RBF. This time the output of ARIMA model will be added to the output of the MLP and the remainder of this result and the real exchange rate will be called $error_2$ as shown in (11).

$$error_2 = realex - (arimaout + mlperror) \quad (11)$$

$$realex = arimaout + mlperror + rbferror \quad (12)$$

Multilayer perceptron has the problem of sometimes getting stuck in the local minima and it is really slow in the learning process but the situation is not this way for RBF neural networks [5], [12]. They are good approximators for any given function [5] and therefore the hybrid model uses RBF at the last stage of error modeling to model the last remaining error which was called $error_2$ in (11). This process is shown in Fig. 3.

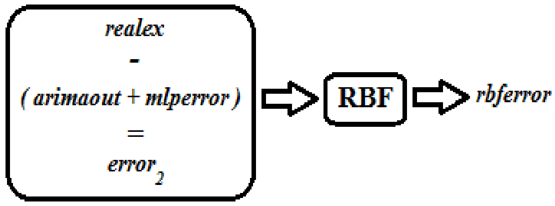

Fig. 3. Hybrid model's third block. modeling the reminder error of ARIMA and MLP with RBF.

The final result of the proposed hybrid model comes from the summation of the error modeled by RBF and the error modeled by MLP with the time series that was modeled by ARIMA as presented in (12). The final complete hybrid model is shown in Fig. 4.

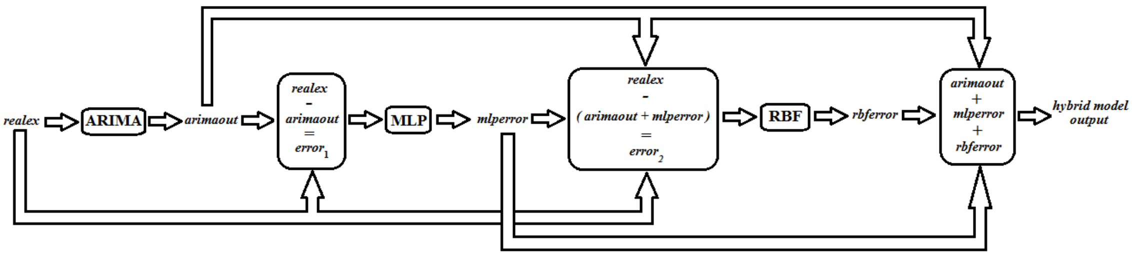

Fig. 4. Hybrid model's third block. Modeling the reminder error of ARIMA and MLP with RBF.

IV. Evaluation Method

In the literature there are two important criteria for evaluating financial time series modellers and predictors [6]. The first criterion for evaluation is the root mean square error (RMSE) between the predicted time series and original time series. The RMSE is defined as below in (13):

$$RMSE = \sqrt{\frac{\sum_{t=1}^{T}(x_t - \hat{x}_t)^2}{T}} \quad (13)$$

In (13), x_t is the value of the original time series and $\hat{x}_t$ is the value of the predicted time series in time t. T is the number of values in the time series, or in the test set of the time series. The RMSE shows the error in terms of the level and not the direction.

The second evaluation method which is more important for the financial traders is the directional success of the prediction. This means that if the original series goes up in the value, the predicted series does the same and if the original series goes down in the value the predicted series shows a downward movement. Traders do the trading by predicting if the price of a product or the value of an exchange rate goes up or down. That's why the directional success of the model is a more important issue in commercial world. The directional success or as we call it from now on, the D_{stat} is calculated as presented in (14), and it shows the success of the prediction in terms of direction.

$$D_{stat} = \frac{1}{T}\sum_{t=1}^{T} a_k$$
$$if\,(x_{t+1} - x_t)(\hat{x}_{t+1} - x_t) \geq 0 \;then\; a_k = 1 \quad (14)$$
$$else\; a_k = 0$$

V. Data Preparation for Empirical Analysis

In this research the time series of EUR/USD (Euro to US Dollar) exchange rate is used for the model design, validation and testing. The data come from the Federal Reserve Bank of St. Luis, economic research center's website [13]. The authors can also present the data to any reader upon request.

There are 3440 daily exchange rate data in the dataset from 1 April 2001 to 31 July 2010. The days without data were padded with the average exchange rate value of the last three days. For the neural network learning purpose we used up to 7 days before as input of the network to predict the next day. We used the 7 days delay because of the weekly seasonal behavior of the exchange rate market.

Data partitioning is very important in achieving better results for any modeling method [7]. Yao and Tan used a 7:2:1 ratio for training set, validation set, and test set in their research and they achieved good prediction results [6]. We also used this ratio for partitioning our data into training, validation, and test set.

VI. Empirical Results and Analysis

Directional success is a good measure to show how acceptable a model will be in the real commercial market [4]. That's because traders make decisions by predicting the direction of price or in our case an exchange rate. If the rate goes higher the next day, it's time to buy or to hold and if the rate goes down it's time to sell or not to buy. We also compare the proposed hybrid model to simple ARIMA and simple MLP and RBF in terms of error level with RMSE.

The ARIMA model that best modeled our exchange rate time series was ARIMA (1, 1, 4). The multilayer perceptron that fitted our model was MLP (7-6-5-1). It means the network had 7 inputs, the exchange rate of 7 days before and two hidden layers with 3 and 2 nodes and 1 output node. The RBF that fitted our model was RBF (7-9-1). It means this RBF neural networks had 7 input nodes which was the data of 7 days before the day we wanted to forecast and 9 nodes in the hidden layer and 1 output node. The MLP for modeling the first error series was MLP (7-4-2-1) and the RBF for modeling was RBF (7-11-1). The learning parameters for learning algorithm of MLP neural network were eta = 0.3 and alpha = 0.9, and for the RBF neural network, the learning parameters were eta = 0.4 and alpha = 0.9. The best learning parameters were chosen by trial and error. After modeling the time series with single ARIMA, MLP, and RBF method and then modeling it with our proposed hybrid model the evaluation results were as follows in Table I.

TABLE I: Evaluation Results of Different Prediction Models

EUR/USD	Hybrid Model	Single ARIMA	Single MLP	Single RBF
RMSE	0.0068	0.0081	0.0076	0.0085
D-stat	71%	65%	67%	59%

The results from the Table I show that our hybrid model works better than the other single models both in terms of error level and directional status.

VII. Conclusion and Further Research

In this research the researchers showed that by using the help of neural networks which are nonlinear modelers and predictors, we can improve the prediction of ARIMA method which is in nature a linear modeler. Improvement in forecasting the financial time series and in our special case the exchange rate time series can help the traders of FOREX (Foreign Exchange) market in their trading, so by better prediction of the future movements of exchange rate they can reduce the trading risk.

Using other machine learning techniques in error modeling of the hybrid model like support vector machines or self-organizing map neural networks or evaluating the model with other exchange rate time series instead of EUR-USD can be a good option for future works.

References

[1] M. Amiri *et al,* "An integrated eigenvector–DEA–TOPSIS methodology for portfolio risk evaluation in the FOREX spot market," *Expert Systems with Applications,* vol. 37, pp. 509–516, 2010.

[2] A. N. Refenes, A. Zapranis, and G. Francis, "Stock performance modeling using neural networks: a comparative study with regression models," *Neural Network*, vol. 5, pp. 961-970, 1994.

[3] L. Yu, S. Wang, and K. Lai, "A neural-network-based nonlinear metamodeling approach to financial time series forecasting," *Applied Soft Computing*, vol. 9, pp. 563-574, 2009.

[4] L. Yu, S. Wang, and K. Lai, "A novel nonlinear ensemble forecasting model incorporating GLAR nad ANN for foreign exchange rates," *Computers & Operations Research*, vol. 32, pp. 2523, 2005.

[5] L. Yu, K. Lai, and S. Wang, "Multistage RBF neural network ensemble learning for exchange rate forecasting," *Neurocomputing*, vol. 71, pp. 3295-3302, 2008.

[6] J. Yao, C. J. Tan, "A case study on using neural networks to perform technical forecasting of FOREX," *Neurocomputing* vol.34, pp. 79-98, 2000.

[7] J. W. Hall, "Adaptive selection of US stocks with neural networks, in: G.J. Deboeck (Ed.), Trading on the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets," Wiley, New York, pp. 45–65. 1994.

[8] H. White, "Learning in artificial neural networks: a statistical perspective," *Neural Computing*, vol. 1, pp. 425-464, 1989.

[9] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, "Time Series Analysis: Forecasting and Control," *Prentice-Hall*, Englewood Cliffs, NJ, 1994.

[10] J. W. Kay and D. M. Titterington, "Statistics and neural networks: advances at the interface," First Edition, Oxford, Oxford University Press, 1999.

[11] M. Bildirici and O. O. Ersin, "Improving forecast of GARCH family models with the artificial neural networks: An application to the daily returns in the Istanbul Stock Exchange," *Expert systems with applications*, vol. 36, pp. 7355-7362, 2009.

[12] S. Chen, S. A. Billings, C. F. N. Cowan, and P. M. Grant, "Nonlinear systems identification using radial basis functions," *Int. J. Syst. Sci*. vol. 21, pp. 2513–2539, 1990.

[13] Federal Resevrve Bank of St. Luis, Economic Research Data. [Online]. Available: http://research.stlouisfed.org/fred2/categories/15.