Predicting Five Currency Exchange Rates Using Five Different Learning Algorithms of Neural Network: A Case Study

Hamed Khadivar\textsuperscript{a}, Ameneh Khadivar\textsuperscript{b}, Mina Hadian Niasar\textsuperscript{c}

\textsuperscript{a}Department of Banking and Finance, Eastern Mediterranean University, Famagusta, Cyprus, \textsuperscript{b}Department of Management, Alzahra University, Tehran, Iran, \textsuperscript{c}Department of Industrial Engineering, University of Arkansas, Fayetteville, USA,

Abstract. One of the most important hopes in modern finance is finding the most accurate ways to forecast future values of exchange rates. The research provides some evidence about the efficiency of utilizing neural network models in forecasting foreign exchange rates. We used three distinct learning algorithms in our neural network models, namely, Scaled Conjugate Gradient (SCG), Standard Backpropagation (SBP), Backpropagation with Bayesian Regularization (BPR), Gradient Descent with Momentum (GDM) and resilient Backpropagation (RB), which were used for United States Foreign Exchange to predict five diverse currencies against the United States dollar. We chose four most important technical indicators based on the Forex experts’ opinion to feed neural network models. Finally, a general comparison between neural network models and a Box-Jenkins model (ARIMA) forecasting model based on the five performance metrics is offered.

Keywords: Neural Network, Neural Network Algorithm, FOREX, Exchange rate, Forecasting.

1. Introduction.

Neural Network is a result of previous researches about Artificial Intelligence which tried to understand and model the brain behavior. There are three main AI systems, namely, expert systems, neural networks and fuzzy-logic systems (VerDuin, 1995; Crowe and Vassiliadis, 1995).

Some researchers such as Bar and Feigenbaum (1981) and Buchanan and Shortlife (1983) categorized neural networks as a part of computer science which concerns to design intelligent computer systems that can imitate the characteristics of the human brain. These computer systems can think and solve problems. Also, Robert Hecht-Nielson (1990) defined neural networks as a computing system which is made up of highly interconnected nodes or processing elements. Although neural networks do not use an algorithm but they have a cyclic procedure specifying step-by-step performing approach that guarantees a correct or optimal solution.

The aim of neural network is to map a set inputs onto a corresponding set of outputs. It uses past information to learn and define the relationship between inputs and outputs. Then, by utilizing what it has learned it can forecast appropriate outputs for a new set of inputs.

Nowadays, exchange rates play an important role in international economics. Forecasting, the science and technology of mining data in order to extract previously unknown patterns, is a part of the overall process of knowledge discovery in
databases (KDD). In our computerized world, there are huge databases storing exchange rate history. However, due to the dynamic nature of business or economical crises dramatic changes in the exchange values have been observed. (Branson, 1997).

Since exchange rates are highly noisy, volatile and chaotic prediction of these rates is so challenging (Yaser and Atiya, 1996; Deboeck, 1994). A wide range of data mining methods has been applied to the forecasting of this time series. They range from simple statistical methods to more advanced methods such as neural networks and genetic algorithms. At the present time, artificial neural network is found to be a popular technique (Hill et al., 1996; Suarez-Farinas, 2004; Lam, 2004; Hansen and Nelson, 2003). Providing the bases of multivariate analysis, not only ANN can rely on grater information, but it can also combine other factors such as technical and fundamental indicators as its predictors. In addition, the capability of the approximating nonlinear function without any pre-assumption has made ANN a perfect tool for predicting foreign exchange rates (Cao and Tay, 2001).

Various studies such as Tang and Fishwich, 1993; Jhee and Lee, 1993; Wang and Leu, 1996; Hill et al., 1996 have been done to show that ANN models outperform Box- Jenkins model in forecasting high frequency series. Moreover, Weigend et al. (1992) proved that ANN is significantly better than the Random walk model. Kaastra and Boyd (1996) discussed about introduction of building financial and economic time series models via ANN. Zhang and Hu (1998) and Yao et al. (2000) provided some information about the ability of backpropagation neural networks for forecasting exchange rates. Klein and Rossin(1999) showed that the quality of input data has a great influence on the predictive models.

In this paper, we used five ANN models, including Scaled Conjugate Gradient (SCG), Standard Backpropagation (SBP), Backpropagation with Bayesian Regularization (BPR), Gradient Decent with Momentum (GDM) and resilientBackpropagation (RB), and a statistical model, namely, Box- Jenkins model in order to predict the exchange rates of the Unites State Dollar (USD) against five other currencies such as Great British Pond (GBP), Japanese Yen (JPY), Australian Dollar (AUD), Canadian Dollar (CAD) and Swiss Franc (CHF). The average of bid and ask closing rate of 3000 days are used for building, 1000 days data for validating and 389 days data for testing of the models. Moreover, most important technical indicators of FOREX such as Relative Strength Index (RSI), Stochastic Oscillator, Moving Average Convergence Divergence (MACD) and Exponential Moving Average are used to feed our models. The outcomes of these models were compared with Box- Jenkins model based on the Normalized Mean Square Error (NMSE), Root Mean Absolute Error (RMSE), Mean Absolute Error (MAE), Directional Symmetry (DS). As it was expected, all ANNs performed more accurately than Box- Jenkins model. Moreover, SCG and BPR results were more precise than SBP.

2. Box- Jenkins Models

ARIMA is a sub-type of Box- Jenkins methods of forecasting (Cao and Tay, 2001). These techniques do not have any pre-assumption about patterns in historical data. There is a repetitive cycle of identifying appropriate model among other models. Then, the approach tries to understand whether the potential model describes historical data accurately. The repetitive cycle comes to work again if the particular model could not fit to the time series, and it will improve on the original one. This process repeated until a satisfactory model is found.

A class of Box- Jenkins models for a stationary time series is the Auto Regressive Integrated Moving Average. ARIMA models predict future values of a time series by a linear combination of its past values and a series of errors. This group of Box- Jenkins model includes the AR model with only autoregressive terms, the MA models with only moving average terms, and the ARIMA models with both autoregressive and moving-average terms. The Bob-Jenkins methodology allows the analyst to select the model
that best fits the data. The details of AR, MA and ARIMA models can be found in the Jarrett paper (1991).

2.1 Neural Networks

ANNs are highly parallel computing approaches imitating the biological brain, which store intelligence in their many interconnecting weights. These variable weights connect nodes (neurons) both in parallel and in sequence. The mechanism hierarchically analyzes vector input through the network of nodes and weights, arriving at a vector output. Neural networks have various advantages for modeling systems containing noisy, fuzzy and uncertain elements. They learn models by repeating through a large number of exemplar vectors. Learning can take place through internal grouping (self organizing or competitive learning) or through paired training sets (supervised learning).

The human brain is composed of hundred billions of neurons and each neuron is an independent biological information processor unit. Usually, each neuron is cooperating with ten thousands surrounding neurons and they build an extremely complex architecture. The human brain is also highly fault tolerant as we continue to function perfectly though neurons are constantly dying. The exceptional characteristics of human brain serve as strong motivation for the idea of building an intelligent machine modeled after biological neuron.

ANN models are very simplified scheme of our understanding of biological neurons. Each neuron’s input fibre called dendrite receives excitatory signals through thousands of surrounding neurons’ output fibre called axon. Figure 1 shows a simple artificial neural network model. Each neuron receives an input $x_j$ from other neuron $j$ which is multiplied by the connection strength called weight $ω_j$ to produce total net input as the weighted sum of all inputs as shown below.

$$net = \sum_j ω_j x_j$$

![Figure 1. An artificial neuron](image)

As shown in figure 2, neural networks consist of three layers, namely input layer, output layer and hidden layers. The neurons of the input layer act as buffers for dispersing the input signal $x_i$ to neurons in the hidden layer. Each neuron $j$ in the hidden layer adds its input signals $x_i$ after multiplying them by the strengths of the respective connection weights $w_{ji}$ and computes its output $y_j$ as a function of the sum, i.e.,

$$y_j = f(\sum w_{ji} x_i)$$
2.2 Training Algorithm

Training a network consists of adjusting its weights using a training algorithm. We used different models as training algorithm. Through performing Scaled Conjugate Gradient a search is performed along conjugate directions that has faster convergence than steepest descent directions (Hagan et al., 1996). In the steepest descent search, novel directions are perpendicular to the old direction. This method is a zigzag path and one step can be mostly undone by the next. In CG method, a new search direction spoils as little as possible the minimization achieved by the previous one and the step size is adjusted in each iteration. The general procedure to determine the new search direction is to combine the new steepest descent direction with the previous search direction so that the current and previous search directions are conjugate as governed by the following equations.

\[ \omega_{k+1} = \omega_k + \alpha_k p_k \]
\[ p_k = -E'(\omega) + \alpha_k p_{k+1} \]

In these equations, \( p_k \) shows the weight vector in \( k \)-th iteration, \( p_k \) and \( p_{k+1} \) are the conjugate directions in successive iterations. Moreover, \( \alpha_k \) and \( \beta_k \) are calculated in each iteration. An important drawback of CG algorithm is the requirement of a line search in each iteration which is computationally expensive. Moller introduced the SCG to avoid the time-consuming line search procedure of conventional CG. SCG needs to calculate Hessian matrix which is approximated by

\[ E''(\omega_k) p_k = \frac{E'(\omega_k + \sigma_k p_k) - E'(\omega_k)}{\sigma_k} + \lambda_k p_k \]

\( E' \) is the first and \( E'' \) is the second derivative of \( E \). \( p_k \), \( \sigma_k \) and \( \lambda_k \) are the search direction, parameter controlling the second derivation approximation and parameter regulating indefiniteness of the Hessian matrix. Considering the machine precision, the value of \( \sigma \) should be as small as possible (<10^{-4}).

**Backpropagation** uses steepest gradient descent technique in order to minimize the sum-of-squared error \( E \) over all training data. During training, each...
desired output $d_j$ is compared with actual output $y_j$ and $E$ is calculated as the sum of squared error at the output layer. Then the model will try to minimize $E$ through improving in weights and allocating more appropriate numbers. The weight $\omega_j$ is updated in the $n$-th training cycle according to the following equation.

$$\Delta \omega_j(n) = -\eta \frac{\partial E}{\partial \omega_j} + \alpha \Delta \omega_j(n-1)$$

**BPR:** A well established neural network model should have a small error on out of sample data, not only on sample data alone. To produce a network with better generalization ability, MacKay [14] proposed a method to restrict the size of network parameters by regularization. Regularization technique forces the network to settle to a set of weights and biases having smaller values. This causes the network response to be smoother and less likely to overfit and capture noise. In regularization technique, the cost function $F$ is defined as:

$$F = \gamma E + \frac{1-\gamma}{n} \sum_{j=1}^{n} \omega_j^2$$

where $E$ is the sum-squared error and $(\gamma < 1.0)$ is the performance ratio parameter, the magnitude of which dictates the emphasis of the training. A large $\gamma$ will drive the error $E$ small whereas a small $\gamma$ will emphasize parameter size reduction at the expense of error and yield smoother network response. The optimum value of $\gamma$ can be determined using Bayesian regularization in combination with Levenberg-Marquardt algorithm.

### 2.3 Neural Network Forecasting Model

In order to predict Forex time series both of the technical and fundamental analysis should have the potential to be used. Since, it has proven that markets are less efficient technical analysis has attracted academic attention recently. Exchange rate like many other economic time series models exhibits its own trend, cycle, season and irregularity. In this study, we used Relative Strength Index (RSI), Stochastic Oscillator, Moving Average Convergence Divergence (MACD) and Exponential Moving Average moving average as technical data. These indicators decrease the effect of irregularities in neural network models. We calculated these indicators for past days and then fed them to the neural network to predict the future day's rate. So the neural network model has four inputs (indicators), one hidden layer and one output unit to predict exchange rate. Historical data are used to train the model. Once trained the model is used for forecasting.
3. Data gathering:

The exchange rate of 5 different currencies against the United States Dollar from December 2000 to December 2012 is collected through Metha Trader4 software. We considered the exchange rates of Great British Pond (GBP), Japanese Yen (JPY), Australian Dollar (AUD), Canadian Dollar (CAD) and Swiss Franc (CHF). For each one of these currencies we obtained the average of bid and ask closing rate (midpoint) of 4389 daily data, and used 3000 days’ for learning model, 1000 days’ data for validating and 389 days’ data for testing. The plots of historical rates of these currencies are shown figure 3.

Figure 3. Historical Data Plot

4. Performance Criteria

The outputs of neural network models and ARIMA model were compared based on the five major statistical metrics including Normalized Mean Square Error (NMSE), Mean Absolute Error (MAE), Root Mean Squared Error and Directional Symmetry (DS).

The Normalized Mean Square Error is an estimate of the overall deviations between predicted and measured values. NMSE generally shows the most striking differences among models. If a model has a very low NMSE, then it is well performed both in space and time. On the other hand, high NMSE values do not necessarily mean that a model is completely wrong. That case could be due to time and/or space shifting. It is defined as:
The Mean of Absolute Error measures the average magnitude of the errors in a set of forecasts, without consideration their direction. It measures the accuracy for continuous variables. In other word, the MAE is the average over the verification sample of the absolute values of the differences between forecasts and the corresponding observations. It is a linear score which means that all the individual differences are weighted equally in the average.

$$NMSE = \frac{\sum (x_k - \hat{x}_k)^2}{\sum (x_k - \bar{x}_k)^2} = \frac{1}{\sigma^2 N} \sum (x_k - \hat{x}_k)^2$$

Root Mean Squared Error is a quadratic scoring rule which measures the average magnitude of the error. It shows the difference between forecast and corresponding observed values are each squared and then averaged over the sample. Finally, the square root of the average is taken. Since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors.

$$MAE = \frac{1}{N} \left| x_k - \hat{x}_k \right|$$

$$RMS = \left( \frac{1}{n} \sum_{i=1}^{n} (\text{model}_i - \text{observed}_i)^2 \right)^{\frac{1}{2}}$$

All of the NMSE, MAE and RMSE are negative–oriented scores meaning that lower values show more accurate models. Finally, DS measures correctness in predicting directions as shown below:

$$DS = \frac{100}{N} \sum d_k,$$

$$d_k = \begin{cases} 1 & \text{if } (x_k - x_{k-1})(\hat{x}_k - \hat{x}_{k-1}) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

5. Experimental Results

We performed our forecasting by using the neural network toolbox of MATLAB containing all of the proposed learning algorithm, namely, Scaled Conjugate Gradient (SCG), Standard Backpropagation (SBP), Backpropagation with Bayesian Regularization (BPR), Gradient Decsent with Momentum (GDM), and resilient Backpropagation (RB). Number of hidden units have a great influence on the network's results. According to the experts' opinion and similar studies we used various numbers of hidden layers ranged from 2 to 10 for each algorithm. Finally, we trained 83 various networks for each algorithm and extracted best results at the end based on the previously defined performance metrics. Results are shown below:
Table 1. Performance metrics for different models

<table>
<thead>
<tr>
<th>Currency</th>
<th>Learning Algorithm</th>
<th>Performance Metrics</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>NMSE</td>
<td>MAE</td>
<td>RMSE</td>
</tr>
<tr>
<td>USD/GBP</td>
<td>SCG</td>
<td>0.0925</td>
<td>0.0002</td>
<td>0.1545</td>
</tr>
<tr>
<td></td>
<td>SBP</td>
<td>0.1556</td>
<td>0.0013</td>
<td>0.1865</td>
</tr>
<tr>
<td></td>
<td>BPR</td>
<td>0.1382</td>
<td>0.0014</td>
<td>0.2964</td>
</tr>
<tr>
<td></td>
<td>GDM</td>
<td>0.1915</td>
<td>0.0054</td>
<td>0.1923</td>
</tr>
<tr>
<td></td>
<td>RB</td>
<td>0.1797</td>
<td>0.0038</td>
<td>0.1734</td>
</tr>
<tr>
<td>USD/JPY</td>
<td>SCG</td>
<td>0.0119</td>
<td>0.0022</td>
<td>0.1512</td>
</tr>
<tr>
<td></td>
<td>SBP</td>
<td>0.0537</td>
<td>0.0096</td>
<td>0.1568</td>
</tr>
<tr>
<td></td>
<td>BPR</td>
<td>0.0741</td>
<td>0.0023</td>
<td>0.1931</td>
</tr>
<tr>
<td></td>
<td>GDM</td>
<td>0.1463</td>
<td>0.0019</td>
<td>0.1927</td>
</tr>
<tr>
<td></td>
<td>RB</td>
<td>0.092</td>
<td>0.0049</td>
<td>0.1333</td>
</tr>
<tr>
<td>USD/AUD</td>
<td>SCG</td>
<td>0.0387</td>
<td>0.002</td>
<td>0.1521</td>
</tr>
<tr>
<td></td>
<td>SBP</td>
<td>0.0312</td>
<td>0.0031</td>
<td>0.1846</td>
</tr>
<tr>
<td></td>
<td>BPR</td>
<td>0.0786</td>
<td>0.0059</td>
<td>0.1712</td>
</tr>
<tr>
<td></td>
<td>GDM</td>
<td>0.1475</td>
<td>0.0011</td>
<td>0.2271</td>
</tr>
<tr>
<td></td>
<td>RB</td>
<td>0.0645</td>
<td>0.0072</td>
<td>0.2448</td>
</tr>
<tr>
<td>USD/CAD</td>
<td>SCG</td>
<td>0.0611</td>
<td>0.0057</td>
<td>0.0119</td>
</tr>
<tr>
<td></td>
<td>SBP</td>
<td>0.1185</td>
<td>0.005</td>
<td>0.0537</td>
</tr>
<tr>
<td></td>
<td>BPR</td>
<td>0.1125</td>
<td>0.0019</td>
<td>0.0741</td>
</tr>
<tr>
<td></td>
<td>GDM</td>
<td>0.1387</td>
<td>0.0011</td>
<td>0.1463</td>
</tr>
<tr>
<td></td>
<td>RB</td>
<td>0.1798</td>
<td>0.0005</td>
<td>0.0921</td>
</tr>
<tr>
<td>USD/CHF</td>
<td>SCG</td>
<td>0.0214</td>
<td>0.0091</td>
<td>0.0899</td>
</tr>
<tr>
<td></td>
<td>SBP</td>
<td>0.0463</td>
<td>0.0077</td>
<td>0.1111</td>
</tr>
<tr>
<td></td>
<td>BPR</td>
<td>0.0754</td>
<td>0.0013</td>
<td>0.1634</td>
</tr>
<tr>
<td></td>
<td>GDM</td>
<td>0.1492</td>
<td>0.0054</td>
<td>0.201</td>
</tr>
<tr>
<td></td>
<td>RB</td>
<td>0.1232</td>
<td>0.0088</td>
<td>0.2397</td>
</tr>
</tbody>
</table>
### Table 2. Performance metrics for ARIMA

<table>
<thead>
<tr>
<th>Currency</th>
<th>Performance Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>USD/GBP</td>
<td>1.7121 0.0451 0.4132 37.143</td>
</tr>
<tr>
<td>USD/JPY</td>
<td>1.0259 0.0386 0.3526 40.019</td>
</tr>
<tr>
<td>USD/AUD</td>
<td>1.2158 0.0268 0.3759 41.198</td>
</tr>
<tr>
<td>USD/CAD</td>
<td>0.9537 0.0113 0.2533 45.547</td>
</tr>
<tr>
<td>USD/CHF</td>
<td>1.0257 0.0951 0.2896 40.001</td>
</tr>
</tbody>
</table>
Figure 4. Forecasting of different currencies by ARIMA over 389 days.
**Figure 5.** Forecasting of USD against AUD by SCG over 389 days

**Figure 6.** Forecasting of USD against CAD by SCG over 389 days

**Figure 7.** Forecasting of USD against CHF by SCG over 389 days

**Figure 8.** Forecasting of USD against GBP by SCG over 389 days
6. Conclusion:

In this study we used 5 different models of neural network to forecast the exchange rate of the United State Dollar against five major currencies, namely, Great British Pond (GBP), Japanese Yen (JPY), Australian Dollar (AUD), Canadian Dollar (CAD) and Swiss Franc (CHF). As it is shown in table 1 and table 2 all of the neural network based models outperformed ARIMA according to Normalized Mean Square Error (NMSE), Root Mean Absolute Error (RMSE), Mean Absolute Error (MAE), Directional Symmetry (DS). Among those five neural networks based models, Scaled Conjugate Gradient (SCG) was the most precise model. Moreover, generally it can be concluded that the accuracy of Standard Backpropagation (SBP), Bckpropagation with Bayesian Regularization (BPR), Gradient Descent with Momentum (GDM), and resilient Backpropagation (RB) are decreasing respectively.

Reference: