Accounting and Management Information Systems
Vol. 12, No. 4, pp. 578–594, 2013

SUPPORTING MANAGEMENT DECISIONS BY USING ARTIFICIAL NEURAL NETWORKS FOR EXCHANGE RATE PREDICTION

Laura Maria BADEA (STROIE)[1]
The Bucharest University of Economic Studies, Romania

ABSTRACT

Management Information Systems are meant to create methods for data management, leading to better decision making. By designing, implementing and using business information systems in innovative ways, the effectiveness and efficiency of every-day activities significantly increases. In the context of uncertainty and high volatilities resulted from the extended crisis, the economic environment became unpredictable, impeding business practitioners from correctly assessing the risks of their activities. These volatilities affected also the evolution of exchange rates, emphasizing even more their nonlinear nature that makes them difficult to model using traditional estimation methods. This paper highlights the benefits of using advanced systems such as Artificial Neural Networks, which provide good solutions to nonlinear problems, guiding business activities in an efficient manner. Different types of Multilayer Perceptron Neural Networks are compared in this study, based on results obtained in predicting EUR/RON exchange rate.

Management Information Systems, Artificial Neural Networks, exchange rate prediction, Multilayer Perceptron

JEL code: C45, D80

INTRODUCTION

Information Systems use Artificial Neural Networks (ANNs) to give the user predictive information on the market and guide management decisions in an effective way. These information technologies are integrated in a series of business

[1] *Correspondence address*: The Bucharest University of Economic Studies, Ion N. Angelescu Building, 6 Romană Square, Bucharest, District 1, Postal Code 010374, Tel. (+40) 21-319.19.00, Email: laura.maria.badea@gmail.com

intelligence tools available today, supporting activities from fields like risk management (Matoussi & Krichene, 2010; Karaa & Krichene, 2012), market segmentation (Štencl *et al.*, 2012) and financial analysis (Kia *et al.*, 2012; Khan & Gour, 2013). ANNs are mathematical representations containing several processing units also called neurons or nodes which act independently, but which grouped in a complex network create the picture of an artificial brain which is adaptable to a series of financial and economic problems. The operating mode of these techniques is based on a series of training-testing experiments which eventually should lead to an optimal solution.

Approaching the use of Artificial Intelligence techniques within Information Systems, Ali *et al.* (2010) used Artificial Neural Networks to extend the functionalities of Geographical Information Systems (GIS) towards modeling and simulation. Based on a case study which used ANNs to predict the distribution of the slum in Cairo, the authors showed that these modeling techniques have great prediction capabilities and can be used to support decision making processes. More recently, Klintong *et al.* (2012) also proved that ANNs provide fast, flexible and strong predictive abilities when selecting the product innovation development project. They showed that ANNs can provide guidance as decision support systems, by obtaining a 96.5% overall correct classification rate on a data set consisting of 87 product innovation projects introduced by entrepreneurial firm in Thailand.

Another important functionality of Artificial Neural Networks when integrated within Management Information Systems (MIS) is time series forecasting. Exchange rates play an important role for emerging economies, being part of many activities and driving a series of decisions and strategies. Therefore, a good estimation of near future FX rates could bring substantial benefits for business activities and trading actions. Traditional time series estimation techniques have often been limited by nonlinearities observed in the data series (Bellgard & Goldschmidt, 1999). This is where, Artificial Neural Networks bring added value, through their capacity to model nonlinear data and also through the increased accuracy outputs reached in forecasting applications. However, ANNs predictive power depend on a series of elements which define ANNs configuration process and which will be further described in this paper.

This study analyzes how differently configured Artificial Neural Networks perform in the process of estimating EUR/RON daily exchange rate. The paper is organized in the following sections: section one provides a brief literature review on Neural Networks performances in exchange rate forecasting applications. Section two describes the methodology approached in this study. Section three gives information on the analyzed data and sampling technique. Section four details the network configurations performed for ANNs and results obtained using different architectures, while the last section concludes on the main benefits, and also drawbacks when using Neural Networks in exchange rate forecasting applications, as MIS integrated techniques.

1. NEURAL NETWORKS AND EXCHANGE RATE PREDICTION

Often, traditional models involved in the process of FX forecasting were unable to deal with the noise and nonlinearities present in the data series. This is how, nowadays, even if they require more time for model configuring, usually resulted from trial and error experiments, ANNs have gained popularity through the higher performances they bring and higher flexibility in handling data.

Yu *et al.* (2007) have analyzed the use of ANNs in forty-five exchange rate forecasting applications, published between 1993 and 2004. Reviews showed that 60% of these articles concluded on the fact that Artificial Neural Network models perform better than other methods, 36% of these provide mixed results, depending on the situation, while only 4%, namely 2 applications, revealed worse performances when using ANNs. Tang & Fishwich (1993), Yao *et al.* (1996), Wang & Leu (1996), Hill *et al.* (1996), Kamruzzaman & Sarker (2003a, 2003b), and many others proved that ANNs generate better results compared with traditional ARIMA models. Plasmans *et al.* (1997) used a neural network error correction model for JPY/USD, GBP/USD and DEM/USD exchange rates, revealing that ANNs significantly outperform both, the random walk model and a linear vector error correction model. Dunis and Williams (2002) showed that ANNs reach higher performances when forecasting EUR/USD exchange rate compared with conventional modeling techniques. Later, Dunis *et al.* (2008) analyzed different types of ANNs and have once again concluded that these provide better results than traditional time series methods. Panda *et al.* (2009) developed two neural network models with the purpose of forecasting exchange rates for USD against GBP, JPY and INR (Indian Rupee). They found that the results of the models using ANNs were superior to those obtained from using least mean square models. Recently, when forecasting the evolution of the Nigerian currency as related to EUR, USD, GBP and JPY, Philip *et al.* (2011) made a comparison between Multilayer Perceptron ANNs and hidden markov technique. Results indicated better results reached by ANNs, with 81.2% accuracy, compared with 69.9% in case of the hidden markov model. Thus, interest in Artificial Neural Networks has increased over the past decades, supported by the limitations reached when using conventional linear models, but also by the explosive expansion of computer usage.

2. METHODOLOGY

ANNs are nonlinear computational models inspired by the way in which the human brain functions. Such methods provide high flexibility in data modeling, of which they make no a priori assumptions, have great capacity in finding interconnections between data, and provide robust results in terms of predicted values. The back-propagation algorithm is one of the most popular learning methods for

Multilayered feed-forward Neural Networks. Adya and Collopy (1998) showed that 88% of the neural networks forecasting applications developed between 1988 and 1994 used error back-propagation as a learning rule. Described for the first time 1969 by Bryson and Ho, the back-propagation algorithm gained popularity beginning with 1986, when Rumelhart *et al.* (1986) presented this learning procedure for adjusting the network weights and minimizing the errors between the actual and the desired output vectors.

Let ($x_1,x_2,\ldots,x_p$) be the set of input vectors and ($y_1,y_2,\ldots,y_p$) the output vectors for $p=1,\ldots,N$ patterns from the training dataset. The input vector is multiplied by the weights which are afterwards iteratively adjusted using the back-propagation method to decrease the error measured as a difference between the desired and the actual outputs. The summation process for each unit of a layer is performed as follows:

$$y_p = g(W_o h_p + \theta_o) \quad (1)$$

$$h_p = f(W_h x_p + \theta_h) \quad (2)$$

where, W_o and W_h are the output and hidden layer matrices of weights,

h_p is the vector denoting the response of hidden layer for pattern p,

θ_o and θ_h are the output and hidden layer bias vectors, respectively,

f(.) is the hidden layer activation function, and *g(.)* is the output layer activation function.

Some of the most common activation functions used within exchange rate forecasting applications are logistic sigmoid and hyperbolic tangent sigmoid:

- Logistic sigmoid function: $f(x) = \dfrac{1}{1+e^{-x}}$ (3)
- Hyperbolic tangent function (tanh): $f(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$ (4)

According to Yu *et al.* (2007: 220), 53% of the forty-five analyzed ANNs articles performing forecasts on exchange rates used the back-propagation algorithm. They also mention that it is a common thing using the same type of activation function in all layers. Kalman and Kwasney (1992) suggested using hyperbolic tangent function, while Kaastra and Boyd (1995) sustain the use of sigmoid functions in the hidden layers (logistic or hyperbolic tangent) and linear function in the output layer. Philip *et al.* (2011) and Kia *et al.* (2012), for instance, used logistic activation function to forecast exchange rates. Yu *et al.* (2007: 116) showed that, on average, hyperbolic tangent activation function provides better outputs compared with logistic sigmoid function. However, depending on the data, results can be better when using one or the other type of activation function.

The error function that one seeks to minimize when using the back-propagation method is the sum of squares:

$$E = \frac{1}{2}\sum_{p}(d_p - y_p)^2 \tag{5}$$

where, d_p is the desired output vector for pattern p, and y_p is the actual output vector for pattern p defined in equation (1).

Depending on the method used for performing the weights adjustments, the back-propagation can use different types of algorithms. In this paper the gradient descent and the scaled conjugate gradient will be approached and results obtained in an exchange rate forecasting application will be compared in terms of errors.

2.1. Gradient descent method

This algorithm is also known as steepest descent method and it defines a way to adjust the weights between neurons based on achieving a gradient descent along the total error surface. Therefore, weights are modified based on the minimum of the local error surface. Defining the weights to a single neuron by a weight vector *w*, the update for epoch *t* is given by the following equations:

$$\Delta w_{k+1} = -\alpha \nabla E(w_k) + \beta \Delta w_k \tag{6}$$

where, Δw_{k+1} is the weight modification for epoch $k+1$

Δw_k is the weight modification from iteration k

$\nabla E(w_k)$ is the gradient of the weight vector resulted at iteration k

α is the learning rate

β is the momentum rate.

The learning rate controls the size of the step from each iteration, while the momentum rate is used to speed up the convergence process in flat regions, or to diminish the jumps in regions of high fluctuations, by adding a fraction of the previous weight change. However, gradient descent algorithm does not always lead to the optimal solution for the weight vector because the method does not approach the minimum of the total error. Moreover, gradient descent method provides a slow learning process whose convergence depends on the properly chosen values for the learning and momentum parameters.

2.2. Conjugate gradient algorithm

In the weight space, the classical gradient descent of the local error provides just the surface of error minimization. Instead, the second derivative of the error generates the minimum point in the error surface. Therefore, weights are modified to evaluate each of the local errors. This modification is achieved through the second derivative of the local error which is obtained with respect to weights, individually for the output and the hidden layers, but also in combination. This leads to generating a difference matrix for the weight modification. Until the minimum of the local error is not obtained, the weights adjustment process proceeds in an iterative manner.

As mentioned before, one of the drawbacks of using steepest descent algorithm is that of a slow convergence process due to performing a high number of steps, which not always grant reaching the global minimum solution. In order to overcome this limitation, conjugate gradient method provides a simple and fast solution, proceeding in a direction which is conjugate to the direction from the previous step. Originally proposed by Hestenes and Stiefel (1952), conjugate gradient method uses the gradient only to calculate the first error descent direction, all other directions being conjugate with the previous ones. This means that the gradient changes only in magnitude, not in direction. Within the conjugate gradient method, the weights are changed after the following scheme:

$$w_{k+1} = w_k + \alpha_k d_k \tag{7}$$

$$d_k = -g_k + \beta_k d_{k-1} \tag{8}$$

$$g_k = \nabla E(w_k) \tag{9}$$

Where, α_k is the size of step k;

β_k is the coefficient that gives the proportion of the previous search direction within the current direction

d_k and d_{k-1} are the conjugate directions from iteration *k* and *k-1*

Nevertheless, conjugate gradient method can be considered a variation of the steepest descent algorithm with learning rate and momentum that are adjusted at each iteration, the learning rate being determined by line minimization and the momentum by using β_k which controls the search direction. Unlike standard conjugate gradient methods, scaled conjugate gradient algorithm does not perform a line search of the conjugate directions to identify the direction of the next step. Instead, it makes a simple approximation of s_k (the scaling factor) based on which the following step is computed. This method was introduced by Moller (1993) which suggests performing the following stages when adjusting the weights by the scaled conjugate gradient method:

i) The weight vector w_1 is initialized and scalars $0 < \sigma \leq 10^{-4}, 0 < \lambda_1 \leq 10^{-6}$, $\overline{\lambda}_1 = 0$. Let $\overline{g}_1 = \overline{d}_1 = -E^{'}(\overline{w_1})$, k=1 and success=true.

ii) If success=true, then find the second-order information as follows: $\sigma_k = \dfrac{\sigma}{|d_k|}$, $s_k = \dfrac{\nabla E^T(w_k + \sigma_k d_k) - \nabla E^T(w_k)}{\sigma_k}$, $\delta_k = d_k{}^T s_k$

iii) Set $\delta_k = d_k + (\lambda_k - \overline{\lambda_k})|d_k|^2$

iv) If $\delta_k \leq 0$, make the Hessian positive definite $\overline{\lambda_k} = 2(\lambda_k - \dfrac{\delta_k}{|d_k|^2})$, $\delta_k = -\delta_k + \lambda_k |d_k|^2$, $\lambda_k = \overline{\lambda_k}$

v) Determine the step size: $\mu_k = d_k^T g_k$, $\alpha_k = \dfrac{\mu_k}{\delta_k}$

vi) Determine the comparison parameter: $\nabla_k = 2\delta_k \dfrac{[E^T(w_k) - E^T(w_k + \alpha_k d_k)]}{\mu_k{}^2}$

vii) If $\nabla_k \geq 0$, then a successful reduction of the error can be performed: $w_{k+1} = w_k + \alpha_k d_k$, $g_{k+1} = -\nabla E(w_{k+1})$, $\overline{\lambda_k} = 0$, success=true

viii) If k mod N=0, then restart the algorithm with $d_{k+1} = g_{k+1}$, else set a new conjugate direction $\beta_k = \dfrac{|g_{k+1}|^2 - g_{k+1} g_k}{\mu_k}$, $d_{k+1} = g_{k+1} + \beta_k d_k$

If $\nabla_k \geq 0.75$ then reduce the scale parameter $\lambda_k = \dfrac{1}{4}\lambda_k$

Else, set $\overline{\lambda_k} = \lambda_k$, success=false

ix) If $\nabla_k < 0.25$, then increase the scale parameter as follows: $\lambda_k = \lambda_k + \dfrac{\delta_k(1-\nabla_k)}{|d_k|^2}$

x) If the steepest descent direction $g_k \neq 0$, then set k=k+1 and return to step ii), else end and return to w_{k+1} as the desired minimum.

Kamruzzaman and Sarker (2004) have shown that when forecasting exchange rates, scaled conjugate gradient performs better than the standard steepest descent approach.

3. DATASET AND SAMPLING TECHNIQUE

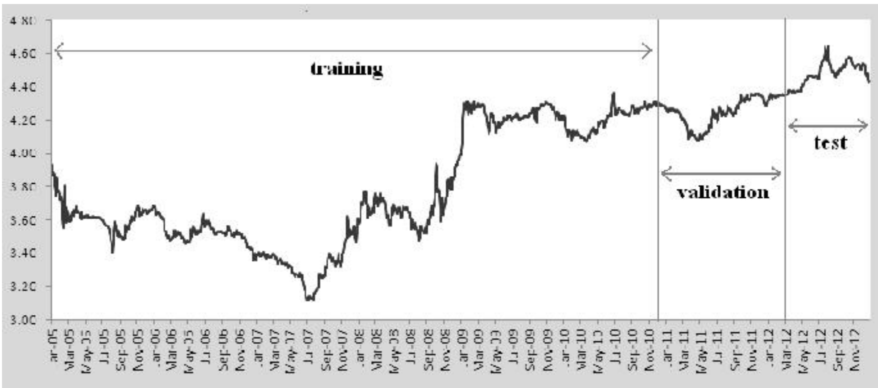
The analyzed database consists of daily closing values for EUR/RON exchange rate from the time period 3rd of January 2005 – 31st of December 2012, available on the National Bank of Romania website. Missing values resulted from non-transactional days, such as legal holidays, were replaced by prices from previous available days.

According to Nucu (2011: 129), from 2000 until 2004, EUR/RON FX rate increased by more than 100%. This was until a certain stability was reached between 2004 and 2007, when fluctuations significantly decreased. As visible from Figure 1, EUR/RON exchange rate presents a nonlinear evolution between January 2005 and December 2012, with periods of decreases, and also with severe increases as the one from November 2008 – March 2009, when the international economic crisis has installed within the Romanian economy too. Over the past two years, given the stressed political context, the confidence in RON's economic power has significantly decreased, EUR/RON exchange rate reaching the highest historical value in August 2012, namely 4.6481.

For evaluation purposes the dataset was split into three databases as follows:
- Training set – 75% of the initial dataset, which was used for model development;
- Validation set – 15% of the initial dataset, used for model assessment;
- Test set – the remaining 10% of the initial dataset, which gives a completely out-of-sample reassessment of the model.

Considering that an FX rate forecasting model would serve for future predictions, the splitting rule was based on the chronological dimension, meaning that the oldest 75% of the prices would fall into training set, the following 15% of the exchange rates would be included in the validation sample, and the most recent 10% would be part of the test dataset. The evolution of EUR/RON exchange rate and the splitting intervals are available in Figure 1.

Figure 1. **EUR/RON exchange rate evolution**



Given the earlier presented splits performed on the initial dataset, passing from one dataset to another, we observe that the training set is marked by high volatilities, the spread between the minimum and the maximum prices being 1.26 RON compared with validation and test samples, which have lower spreads, of about 0.28 RON each (Table 1). However, this is somehow natural, because larger timeframes usually provide higher fluctuations, especially when switches from economic growth to recession take place, such as the one captured by the training set.

Table 1. **Basic statistics by data samples**

Data sets	**Timeframe**	**Average**	**Min**	**Max**	**St. dev.**	**No. of daily exchange rates**
Training	3rd Jan 2005 – 31st Dec 2010	3.7688	3.1112	4.3688	0.3494	1565
Validation	3rd Jan 2011 – 14th Mar 2012	4.2561	4.0735	4.3627	0.0858	313
Test	15th Mar 2012 – 31st Dec 2012	4.4835	4.3672	4.6481	0.0669	208
Total		3.9132	3.1112	4.6481	0.3986	2086

4. MODEL CONFIGURATION AND RESULTS

For model building, it was chosen the Multilayer Perceptron (MLP) using, on turns, one of the following two types of learning algorithms: the standard back-propagation algorithm using gradient descent approach (SBP) and the scaled conjugate gradient method (SCG) presented in section two of this paper.

4.1. Forecasting time horizon

According to Huang *et al.* (2003) ANNs perform better than random walk when a forecasting period of 1, 3 or 5 days is considered. For forecasting periods longer than 8-10 days, however, like Chun and Kim (2003) have showed, ANNs provide poor results. In the present paper, the goal is to predict one step ahead based on one previous observed exchange rate.

4.2. Activation functions

Following the recommendation of Kaastra and Boyd (1995) in terms of activation functions, logistic sigmoid and hyperbolic tangent functions were used on turns in the hidden layer, and linear activation function was chosen for the output layer. Considering that logistic sigmoid and hyperbolic tangent functions provide outputs within the intervals [0;1] and [-1;1] respectively, the initial exchange rates were normalized using the formula below:

$$NM_t = \frac{EUR/RON_t - MIN_{EUR/RON}}{MAX_{EUR/RON} - MIN_{EUR/RON}}(\omega - \varepsilon) + \varepsilon \tag{10}$$

where, NM_t is the normalized value for point *t*

EUR/RON_t is the initial exchange rate for time point *t*,

$MIN_{EUR/RON}$ is the lowest price observed on EUR/RON,

$MAX_{EUR/RON}$ is the highest value from the database

ω is the highest bound of the interval

ε is the lowest bound of the interval

Depending on the activation function used, the upper and lower bounds were set at [0.1;0.9] for logistic sigmoid and at [-0.9; 0.9] for hyperbolic tangent, in order to avoid working near the asymptote.

4.3. Performance metrics

Two measures were chosen for determining the performance of the developed models, namely the Mean Squared Error (MSE) and the Directional Success Rate (DS) which reveals the success rate of the prediction in terms of direction. These are presented below:

$$MSE = \frac{\sum_{t=1}^{n}(EUR/RON_t - EUR/RON^{estimated}{}_t)^2}{n} \tag{11}$$

where, n is the number of periods,

$EUR/RON^{estimated}{}_t$ is the estimated exchange rate for time point *t*

$$DS = \frac{1}{N}\sum_{i=1}^{N} ds_t \tag{12}$$

where,

$$\begin{cases} ds_t = 1, if(EUR/RON_t - EUR/RON_{t-1})* \\ \quad *(EUR/RON^{estimated}{}_t - EUR/RON^{estimated}{}_{t-1}) \geq 0 \\ ds_t = 0, if(EUR/RON_t - EUR/RON_{t-1})* \\ \quad *(EUR/RON^{estimated}{}_t - EUR/RON^{estimated}{}_{t-1}) < 0 \end{cases} \tag{13}$$

4.4. Stopping rule

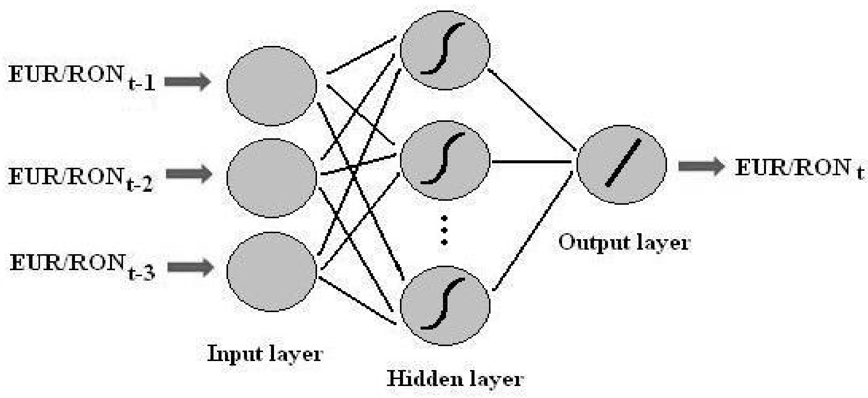
The stopping rule was set to either of the following events, meaning that whichever happens first, leads to stopping the training process:

i) A variation of the average error for 20 consecutive cycles below 0.0000001, avoiding this way the "over-fitting" phenomenon from happening;
ii) A maximum number of 1000 iterations.

4.5. Input nodes

The number of input nodes was set to three. Thus, to forecast the exchange rate for period *t*, the exchange rates observed over the previous three days were taken into consideration, namely from time points *t-1, t-2 and t-3*. Figure 2, provides an image of the MLP architecture before selecting the number of hidden nodes.

Figure 2. **MLP Architecture**



4.6. Hidden layers

Regarding the number of hidden layers, Knerr *et al.* (1992) showed that one hidden layer suffices most of the times, providing accurate outputs. This of course happens if the number of hidden nodes is properly selected. For the forecasting process described in this paper, one hidden layer was selected. From this point onwards, the abbreviation MLP 3-N-1 will be used to define a Multilayer Perceptron with three layers, namely an input layer containing three nodes, an output layer having one node, and one hidden layer containing N nodes.

4.7. Hidden nodes

Having set the number of input nodes and hidden layers, the following important step becomes selecting the number of hidden neurons. As Jha (2009) pointed out, there is usually no need using a number of hidden neurons greater than twice the number of input nodes. In the selected architecture, this means a maximum number of six hidden neurons. However, Ramlall (2010) advocates for the need to carefully select the number of hidden neurons, as otherwise we risk obtaining an "over-fitted" model or, on the contrary, a poor performing one. Nevertheless, as there is generally no pre-specified "best architecture", depending each time on the analyzed data, the number of hidden neurons is selected based on trial and error comparisons. In this paper, the number of hidden nodes was varied between three and six using, on turns, logistic sigmoid and hyperbolic tangent activation functions in the hidden layer. These variations in the model architecture were performed using both, SBP and SCG algorithms. When using SBP, the learning rate and momentum rate parameters were set at 0.1 each, as no significant changes resulted when modifying their values.

4.8. Results

In choosing the number of hidden neurons, the approach was based on selecting that MLP architecture which provides the lowest MSE on the out-of-sample test set. This brings the ranking available in Table 2. Results indicate that the number of hidden neurons depends on the neural network architecture. Oscillating both, the learning algorithm and the activation function from the hidden layer, results are better in some cases when using less hidden nodes and in some others when using a larger number of hidden nodes. However, the best results in terms of MSE and DS were reached when using SCG learning algorithm with hyperbolic tangent activation function and with five nodes in the hidden layer (MLP SCG 3-5-1 Tanh).

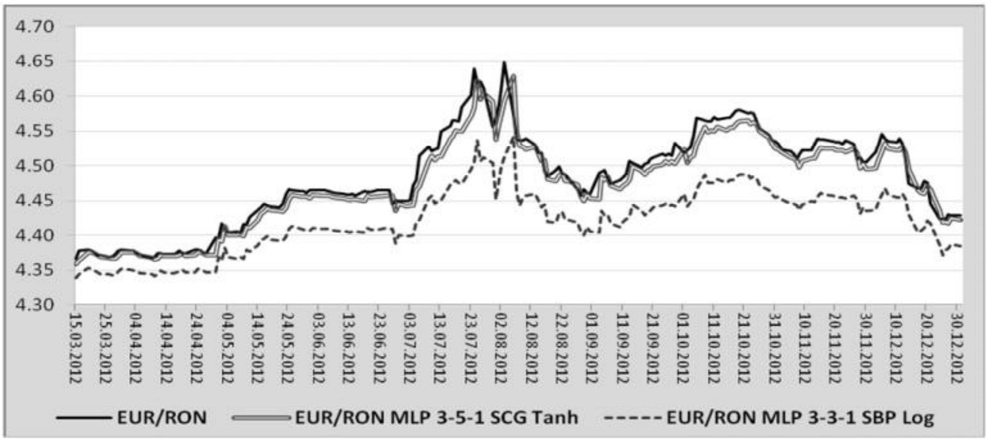
When learning with the SBP, the best performance in terms of MSE and SD is reached by a MLP with three hidden neurons using a logistic sigmoid activation function in the hidden layer (MLP SBP 3-3-1 Log). However, even if the statistic DS reaches slightly better results for SBP on the training and test datasets, the value for MSE is significantly higher for SBP and this shows that forecasts using SCG reach smaller deviations from the true exchange rates. For similar results on DS metric (62.98%), SCG still generates a MSE that is over ten times lower on the test set compared with SBP method (MLP SCG 3-3-1 Tanh compared with MLP SBP 3-3-1 Log).

Table 2. **Results for different neural networks architectures**

Ranking by test MSE	MLP architecture	Training MSE	Training DS	Validation MSE	Validation DS	Test MSE	Test DS	Activation function in the hidden layer
1	**MLP SCG 3-5-1 Tanh**	**0.000319**	**54.55%**	**0.000132**	**51.12%**	**0.000236**	**60.58%**	**Tanh**
2	MLP SCG 3-4-1 Tanh	0.000322	55.19%	0.000133	52.08%	0.000274	62.02%	Tanh
3	MLP SCG 3-6-1 Tanh	0.000319	55.31%	0.000134	51.44%	0.000304	60.10%	Tanh
4	**MLP SCG 3-3-1 Tanh**	**0.000351**	**55.63%**	**0.000136**	**52.72%**	**0.000317**	**62.98%**	**Tanh**
5	MLP SCG 3-4-1 Log	0.000406	53.65%	0.000144	52.08%	0.000346	58.65%	Logistic
6	MLP SCG 3-3-1 Log	0.000594	50.58%	0.000185	52.08%	0.000465	52.40%	Logistic
7	MLP SCG 3-5-1 Log	0.000397	55.12%	0.000209	51.76%	0.002221	59.13%	Logistic
8	**MLP SBP 3-3-1 Log**	**0.033725**	**55.89%**	**0.000792**	**51.12%**	**0.004429**	**62.98%**	**Logistic**
9	MLP SBP 3-6-1 Tanh	0.025697	55.44%	0.000722	50.48%	0.004538	62.02%	Tanh
10	MLP SBP 3-4-1 Tanh	0.026177	55.38%	0.000742	50.48%	0.004539	62.02%	Tanh
11	MLP SBP 3-5-1 Tanh	0.026018	55.51%	0.000730	50.48%	0.004637	62.02%	Tanh
12	MLP SBP 3-4-1 Log	0.040994	55.95%	0.000921	51.44%	0.004812	62.98%	Logistic
13	MLP SBP 3-5-1 Log	0.047455	55.89%	0.001057	51.12%	0.005587	62.50%	Logistic
14	MLP SBP 3-3-1 Tanh	0.025140	55.44%	0.000774	50.48%	0.005838	62.02%	Tanh
15	MLP SBP 3-6-1 Log	0.054514	56.27%	0.001198	51.44%	0.006325	62.02%	Logistic
16	MLP SCG 3-6-1 Log	0.006252	50.90%	0.000999	52.08%	0.011999	52.88%	Logistic

Figure 3 provides a comparison between the true EUR/RON exchange rate from test dataset and the forecasted values using the best MLP models as emerged from Table 2, i.e. MLP SCG 3-5-1 Tanh, and MLP SBP 3-3-1 Log. Results highlight that SCG learning algorithm reaches outputs much closer to the real observed EUR/RON exchange rates compared with SBP method.

Figure 3. **EUR/RON observed vs. forecasts on the test set**



Making predictions on EUR/RON exchange rate with scaled conjugate gradient method resulted thus into significantly lower MSEs compared with standard gradient descent training method on in-sample data, but also on out-of-sample dataset, providing a high generalization capacity and a flexible approach for advanced Information Systems.

CONCLUSIONS

Time series estimation is an important functionality of MIS applications, especially for making predictions on exchange rates which are part of many financial and economic activities. The more advanced the forecasting methods are, the more accurate results become and thus, the effectiveness of decision making process increases in terms of success and rapidness. Artificial Neural Networks are complex methods incorporating the necessary features to be used in exchange rates predictions. They provide good results to nonlinear problems by incorporating some of the human brain functionalities. Nevertheless, Neural Networks development is a complex process, requiring several model configurations, training, testing and then comparing results using the selected performance metrics. The network configuration elements and essentially the training algorithm were shown in this study to have a significant influence on the final model results. With a proper model selection, ANNs can generate highly accurate predictions of exchange rates, providing support for better managerial decisions when integrated within MIS.

Further steps and research directions regarding the evaluation of ANNs in exchange rate forecasting applications should focus on evaluating the effect of external qualitative factors such as political events at national and international level, and also on using other training algorithms.

REFERENCES

Adya, M. & Collopy, F. (1998) "How Effective are Neural Networks at Forecasting and Prediction. A Review and Evaluation", *Journal of forecasting*, vol. 17: 481-495

Ali, A.L., Hegazy, O. & Eldien, M.N. (2010) "A Framework for integration between Artificial Neural Network & Geographical Information System, Slum prediction as the case study", *International Journal of Electrical & Computer Sciences,* vol. 10, no. 01: 25-39

Bellgard, C. & Goldschmidt, P. (1999) "Forecasting Across Frequencies: Linearity and Non-Linearity", University of Western Australia Research Paper, *Proceedings of the International Conference on Advanced Technology*, Australia

Bryson, A.E. & Ho, Y.C. (1969) *Applied Optimal Control. Optimization, Estimation, and Control*, Blaisdell Publishing Company, USA

Chun, S.H. & Kim, S.H. (2003) "Impact of Momentum Bias on Forecasting Through Knowledge Discovery Techniques in the Foreign Exchange Market", *Expert Systems with Applications*, vol. 24:115-122

Dunis, C.L. & Williams M. (2002) "Modelling and Trading the EUR/USD Exchange Rate: Do Neural Network Models Perform Better?", *Derivatives Use, Trading and Regulation*, vol. 8, no. 3: 211–239

Dunis, C.L., Laws, J. & Sermpinis, G. (2008) "Higher Order and Recurrent Neural Architectures for Trading the EUR/USD Exchange Rate", *CIBEF Working Paper*

Hestenes, M. R. & Stiefel, E. (1952) "Methods of Conjugate Gradients for Solving Linear Systems", *Journal of Research of the National Bureau of Standards*, vol. 49: 409–436

Hill, T., O'Connor, M. & Remus, W. (1996) "Neural Network Models for Time Series Forecasts", *Management Science*, vol. 42: 1082-1092

Huang, W., Lai, K.K., Nakamori, Y. & Wang, S.Y. (2003) "An Empirical Analysis of Sampling Interval for Exchange Rate Forecasting with Neural Networks", *Journal of Systems Science and Complexity*, vol. 16, no. 2: 165-176

Jha, G.K. (2009) *Artificial Neural Networks*, Indian Agricultural Research Institute, PUSA, New Delhi

Kaastra, I. & Boyd, M.S. (1995) "Forecasting Futures Trading-Volume Using Neural Networks", *The Journal of Futures Markets*, vol. 15, no. 8: 953-970

Kalman, B.L. & Kwasney, S.C. (1992) "Why Tanh? Choosing a Sigmoidal Function", *Proceedings of the International Joint Conference on Neural Network*, Baltimore, MD IEEE, New York

Kamruzzaman, J. & Sarker, R.A. (2003a) "Forecasting of Currency Exchange Rates Using ANN: A Case Study", *Proc. of IEEE Int. Conf. on Neural Networks & Signal Processing* (ICNNSP03), 793-797

Kamruzzaman, J. & Sarker, R.A. (2003b) "Comparing ANN Based Models with ARIMA for Prediction of Forex Rates", *ASOR Bulletin*, vol. 22: 2-11

Kamruzzaman, J. & Sarker, R.A. (2004) "ANN-Based Forecasting of Foreign Currency Exchange Rates", *Neural Information Processing - Letters and Reviews*, vol. 3, no. 2: 49-58

Karaa, A. & Krichene, A. (2012) "Credit-Risk Assessment Using Support Vectors Machine and Multilayer Neural Network Models: A Comparative Study Case of a Tunisian Bank|", *Journal of Accounting and Management Information Systems*, vol. 11, no. 4: 587-620

Khan, A.U. & Gour, B. (2013) "Stock Market Trends Prediction Using Neural Network Based Hybrid Model", *International Journal of Computer Science Engineering and Information Technology Research*, vol. 3, no. 1: 11-18

Kia, A.N., Fathian, M. & Gholamian, M. R. (2012) "Using MLP and RBF Neural Networks to Improve the Prediction of Exchange Rate Time Series with ARIMA", *International Journal of Information and Electronics Engineering*, vol. 2, no. 4: 543-546

Klintong, N., Vadhanasindhu, P. & Thawesaengskulthai, N. (2012) "Decision Support System using Artificial Neural Network for Managing Product Innovation", *IJCSI International Journal of Computer Science Issues*, vol. 9, issue 2, no. 2: 114-122

Knerr, S., Personnaz, L. & Dreyfus, G. (1992) "Handwritten Digit Recognition by Neural Networks with Single-Layer Training", *IEEE Transactions on Neural Networks*, vol. 3, no. 6: 962-968

Majhi, R., Panda, G. & Sahoo, G. (2009) "Efficient Prediction of Exchange Rates with Low Complexity Artificial Neural Network Models", *Expert Systems with Applications*, vol. 36, no. 1: 181-189

Matoussi, H. & Krichene, A. (2010) "Credit Risk Evaluation of a Tunisian Commercial Bank: Logistic Regression vs Neural Network Modelling", *Journal of Accounting and Management Information Systems*, vol. 9, no. 1: 92-119

Moller, M.F. (1993) "A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning", *Neural Networks*, vol. 6: 525-533

Nucu, A.E. (2011) "The Relationship between Exchange Rate and Key Macroeconomic Indicators. Case Study: Romania", *The Romanian Economic Journal*, vol. 41: 127 -145

Philip, A.A., Taofiki, A.A. & Bidemi, A.A. (2011) "Artificial Neural Network Model for Forecasting Foreign Exchange Rate", *World of Computer Science and Information Technology Journal*, vol. 1, no. 3: 110-118

Plasmans, J.E.J., Weeren, A.J.T.M., & Dumortier, F. (1997) *Exchange Rate Modeling by Multivariate Nonlinear Cointegration Analysis using Artificial Neural Networks,* The Pennsylvania State University CiteSeerX Archives

Ramlall, I. (2010) "Artificial Intelligence: Neural networks Simplified", *International Research Journal of Finance and Economics*, vol. 39: 105-120

Rumelhart, D., Hinton, G. & Williams, R. (1986) "Learning Representations by Backpropagation Errors", *Nature*, vol. 323: 533-536

Štencl, M., Popelka, O. & Šťastný, J. (2012) "Forecast of Consumer Behaviour Based on Neural Networks Models Comparison", *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis*, vol. 60: 437-442

Tang, Z. & Fishwich, P. A. (1993) "Backpropagation Neural Nets as Models for Time Series Forecasting", *ORSA Journal on Computing*, vol. 5, no. 4: 374-385

Wang, J. H. & Leu, J. Y. (1996) "Stock Market Trend Prediction Using ARIMA-Based Neural Networks", *Proc. of IEEE Int. Conf. on Neural Networks*, vol. 4, 2160-2165

Yao, J., Poh, H. & Jasic, T. (1996) "Foreign Exchange Rates Forecasting with Neural Networks", National University of Singapore Working Paper, *Proceedings of the International Conference on Neural Information Processing*, Hong Kong

Yu, L., Wang, S., Huang, W. & Lai, K.K. (2007) "Are Foreign Exchange Rates Predictable? A Survey from Artificial Neural Networks Perspective", *Scientific Inquiry: A Journal of International Institute for General Systems Studies*, vol. 8, no. 2: 207-228