

New Approaches of NARX-Based Forecasting Model. A Case Study on CHF-RON Exchange Rate

Cătălina-Lucia COCIANU, Mihai-Șerban AVRAMESCU

The Bucharest University of Economic Studies

catalina.cocianu@ie.ase.ro, mihai_serban.avramescu@yahoo.com

The work reported in the paper focuses on the prediction of the exchange rate of the Swiss Franc-Romanian Leu against the US Dollar-Romanian Leu using the NARX model. We propose two new forecasting methods based on NARX model by considering both additional testing and network retraining in order to improve the generalization capacities of the trained neural network. The forecasting accuracy of the two methods is evaluated in terms of one of the most popular quality measure, namely weighted RMSE error. The comparative analysis together with experimental results and conclusive remarks are reported in the final part of the paper. The performances of the proposed methodologies are evaluated by a long series of tests, the results being very encouraging as compared to similar developments. Based on the conducted experiments, we conclude that both resulted algorithms perform better than the classical one. Moreover, the retraining method in which the network is conserved over time outperforms the one in which only additional testing is used.

Keywords: Exchange Rate Forecasting, NARX Neural Network, Retraining, Time-series, Mobile Average

1 Introduction

The prediction of time series is a very topic subject with a promising future in the context of globalization and the full implication of information technology in the financial field.

For the economic processes it is essential that the solutions offered by the literature be found in the conception of the applications and the understanding of the phenomena. The need to predict data in the form of time series is generally understood in the context of stock market speculation, more precisely in terms of potential profit or loss. The level of knowledge in the field is of real use in analyzing markets, stock prices, national economy evolution, in order to see possible collapses, scenarios or even frauds in time.

This paper aims to find a new viable solution by combining several elements of statistics, mathematics and neural computations.

The main purpose of this article is to study the variance of the Swiss franc exchange rate - RON according to the dollar-RON exchange rate. Taking into account the volume of credits granted in Swiss francs in Romania, the variance of this exchange rate according to the

most traded currency on the international market, achieving an accurate forecasting is extremely useful.

There are two prediction categories, based on the number of values to predict. The first is the one-step ahead prediction and the second is the multi-step ahead prediction. In the study [1] the two categories are highlighted by the fact that the long-term predicted values can be calculated only based on the forecasted predicted values. This observes a limitation consisting of the accumulation of errors and lack of concrete information. The time series that correspond to the exchange rates are by nature non-stationary. Since 1989, the paper [2] describes that the property of the volatility of trajectories generated by nonlinear systems makes the predictive value of future values extremely difficult.

In a recent study [3], the authors obtained better results using SVM (part of Analogy paradigm) in front of NARX (part of Connectivism paradigm) to predict a stock market. Another limitation is described in [4] where the authors note that the prediction performance of neural networks is very dependent on network structure and the training procedure. At the same time, the difficulty of identifying

data quality, data preprocessing, networking, and repeating this process until a good model is determined must not represent a discouragement [4]. Ahmed [5] predicts the currency exchange rate of a coin according to three other exchange rates using a neural approach, the results after the network was coaching by showing that ANN is an effective prediction method. Also, finding the right lag, the market prediction ability for the NARX network ranged between 99% (best result) and 47% (the worst result) [5]. Exogenous Autoregressive Nonlinear Networks have the advantage that they can be used for a very large range of prediction issues and the promising approach they are looking at. As can be seen from [6], a comparative analysis between ANN models and econometric models shows that MLFFNN (Multilayer Feed Forward Neural Network) and NARX (Nonlinear Auto Regressive Exogenous) are better methods of predictive efficiency. Another conclusion from this study is that for the NARX model the number of hidden layer neurons and network drive algorithms do not have a significant influence on the results obtained. This is an encouragement to focus on the optimizations that can be brought to the ANN networks without focusing on changing the core components.

There are three ways to see the problem we proposed to analyze:

- (A) The ideal results. The general problem consists in obtaining an artificial neural network that successfully learns input data and has the ability to learn the patterns from data as accurately as possible so that the test result is as close as possible to the real, new and unknown values.
- (B) The limitations. There are a lot of factors that prevent the accuracy of predictions for instance: mathematical limitations (the use of probability-based algorithms), computational limitations (time-consuming, complex configurations, computational complexity, processing power), statistically limitations (non-stationarity of the time series, the need of large amounts of data)
- (C) What is our purpose. Based on the features

provided by the Matlab software, we optimized the NARX network using 3 models. In this study a new NARX scheme will be developed to determine the forex rate prediction. Thus, three methods will be compared for predicting: (1) the standard NARX method offered by Matlab software with an additional test method in which the network will be validated, (2) the previous method with an additional feature to can conserve the network if it obtains good results on training but poor results on testing and (3) the previous method with an additional feature to can made changes on the weight while the network is conserve.

The NARX model has a flexible architecture that combines simplicity (such as feed-forward neural networks) and time series prediction (recurrent neural networks) [7].

The rest of paper is organized as follows. The general NARX prediction model is briefly described in Section 2. The classic, most commonly used forecasting model together with the proposed techniques developed to improve the forecasting accuracy and generalization capacities of the trained neural network are presented in the third section of the paper. In Section 4 the quality measure used to evaluate the results and the acquisition of data analyzed in this case study are presented. The last section is dedicated to disseminating and discussing the results obtained and suggesting ideas for future work.

2 NARX Model

One of the most powerful class of discrete-time nonlinear systems in time series modeling is the nonlinear autoregressive with exogenous inputs (NARX) model. The NARX model computes a function of the delayed input and output values expressed by

$$y(t) = f\left(y(t - d_y), \dots, y(t - 1), u(t - d_u), u(t)\right)$$

where $y(t) \in R$, and $u(t) \in R$ are the input and the output of the network at time t , d_u and d_y are the input and the output order, while f is the nonlinear function. When the nonlinear function f can be approximated by a multi-layer perceptron, the resulting system

is called a NARX neural network.

The NARX model is part of the recurrent network class. These networks have the property to store past values and add the output of the hidden layer in part from the new network input.

According to [8], the convergence rate and learning ability of NARX neural networks are higher than other ANNs. This makes them suitable for multi-step forward prediction, as is the case in the present study.

From a formal point of view, the equation for the NARX model is:

$$y_{t+1} = f(y_t, y_{t-1}, \dots, y_{t-d_1}, u_t, u_{t-1}, \dots, u_{t-d_2}) \quad (3)$$

where y_{t+1} is the next predicted value of the dependent time series $\{y_t\}$, calculated based on the past values of the exogenous series $\{u_t\}$. f function is the global operation of the neural network through which the estimate is made. The values d_1 and d_2 represent the delays of the target and input and must respect the following properties: $d_1 > 0$, $d_2 \geq 0$ and $d_1 \geq d_2$.

The representation of the architecture of a NARX neural network is presented in Figure 2.

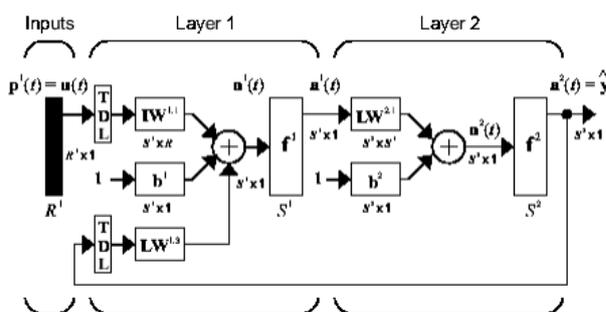


Fig. 1. NARX Architecture. Source: Beagle et al. (2017)

There are two possible configurations for training a NARX network. In the first case is the parallel-type architecture (4) in which the output is partially transformed from the network input. The second variant is the serial-parallel architecture (5) in which the actual output is used instead of the estimated output, which leads to the benefit of more accurate results.

$$\dot{y} = \hat{f}(y_t, y_{t-1}, \dots, y_{t-d_1}, u_t, u_{t-1}, \dots, u_{t-d_2}) \quad (4)$$

$$\dot{y} = \hat{f}(y_t, y_{t-1}, \dots, y_{t-d_1}, u_t, u_{t-1}, \dots, u_{t-d_2}) \quad (5)$$

To determine the number of neurons on the hidden layer, use formula (6). A large number of neurons will make learning too dependent on the network and the accuracy of the predicted results will suffer. At the same time, a small number of neurons may be insufficient for learning.

$$H = 2 \sqrt{N * (m + 2)} \quad (6)$$

where N is the output size and m is the size of the input.

The Levenberg-Marquardt algorithm is used to train the network. It provides a numerical solution for the problem of minimizing the smallest squares [3] and the experimental results show the effectiveness of ANN using the LM algorithm [10].

3 Forecasting Technique

The study focuses on getting results through the three methods mentioned in the first section.

In Figure 2 are the status charts for the algorithm in which the configuration is tested on additional data until obtaining a desired result. The software development is based on recursive functions. Considering the following components of the methodology: CN () - Function to create the neural network, TRN () - Function to test the neural network and TN () - Function to test the neural network, the representation of the flow corresponding to the first model is presented in Figure 3.

One way to improve the performances of the forecasting methods is to consider additional information obtained base on technical analysis. Note that technical analysis considers the analysis of market-based information such as price and volume to analyze future price developments.

Therefore, we include additional exogenous variables that, on one hand, could help to better train the network and, on the other hand, could have the ability to describe the future price movements.

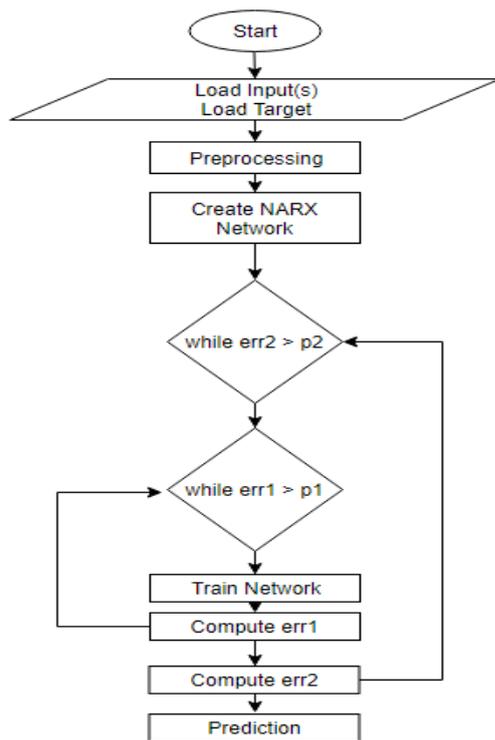


Fig. 2. Extra data test method

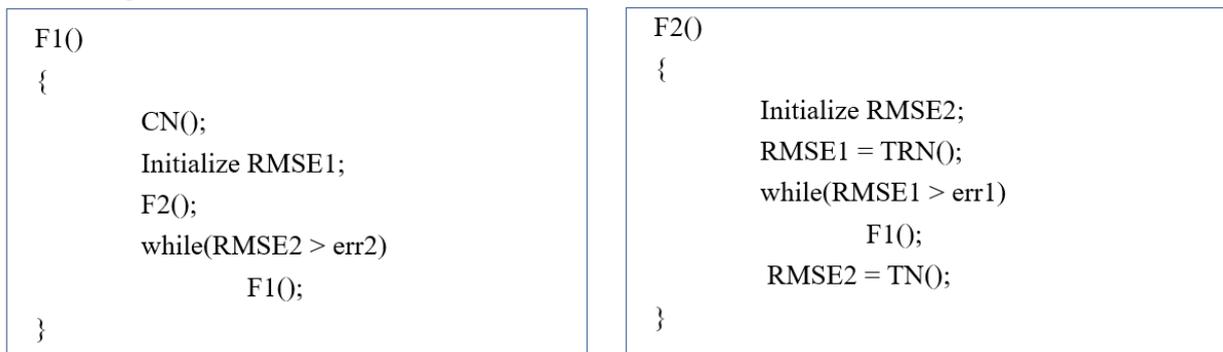


Fig. 3. Recursive Flow for Model 1

The second proposed model for the problem consists in conserving the properties of the neural network (weights and biases) and try to improve its generalization capabilities by re-training it.

This concept is based on the idea of not losing a well-trained network that does not get good

Beside the USD series considered for prediction, our models include the following technical indicators: the Open, the High and the Low values of USD and the moving average of CHF series. We take into account these technical indicators based on the studies reported in [3].

A moving average is an indicator that shows the average value of a security's price over a specified period of time. The N -period MA for the asset's price is computed by:

$$MA_N(t) = \frac{P_t + P_{t-1} + \dots + P_{t-N+1}}{N}$$

where P_t is the current price, and N is the number of time periods. The moving average has the role of analyzing the trend of the time series. In our approach N is set to 5: the value for current day is computed as the average of the last 5 days.

results on the additional test and trying to improve it.

Future data will be predicted after finding a NARX configuration that falls within the expected error limits. The flow is represented in Figure 4. The value for iteration limitation is chosen arbitrary, in our case the value is 10.

```

F1()
{
    CN();
    Initialize RMSE1;
    i = 0;
    F2();
    while(RMSE2 > err2){
        while (i < 10) {F2();}
        F1();}
}
    
```

```

F2()
{
    Initialize RMSE2;
    RMSE1 = TRN();
    while(RMSE1 > err1)
        F1();
    RMSE2 = TN();
    i++;
}
    
```

Fig. 4. Recursive Flow for Model 2

The third proposed model for the problem consists in conserving the most of the properties of the neural network (weights and biases) in order to re-establish the same network. This concept is based on the idea of forcing an optimal configuration by modifying a small part of a well-trained network. The way in which changes are made to a value of the weight on the input layer is represented in Figure 5.

```

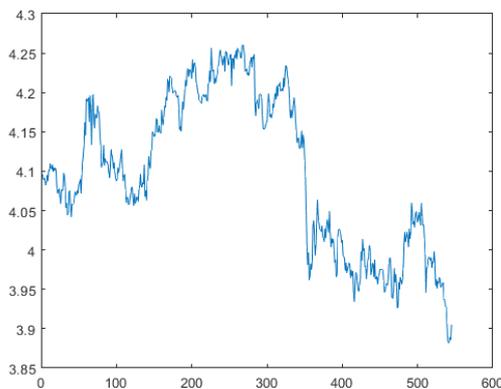
a = netc.IW{1,1};
poz1 = randi([1 nr_h_neurons],1,1);
a(poz1,1) = a(poz1,1) + unifrnd(-0.1000 , 0.1000);

netc.IW{1,1} = a;
    
```

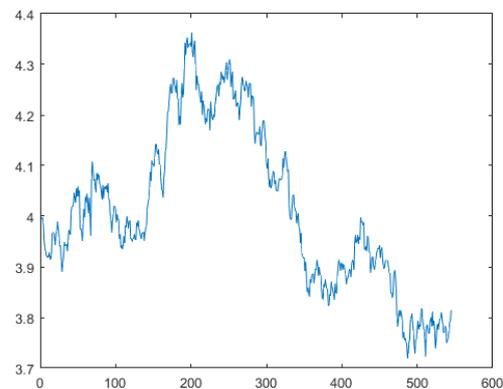
Fig. 5. Changes on weights level for Model 3 to the period April 2016 - March 2018 and were collected from the database of the National Bank of Romania [11]. Their charts are as follows:

4 Error measurements

The data on which the case study is based is the history of the exchange rate between CHF-RON and USD-RON. The records correspond



(a)



(b)

Fig. 6. Exchange Rate Daily Data for Swiss Franc (a) and for US Dollar (b)

The computation of the training error and the prediction are usually done by referring to the mean square error (RMSE), which is proven to be an appropriate metric for time series with a sufficiently large number of observations [12].

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_p - y_r)^2} \quad (7)$$

where:

n = the size of the time series

y_p = the current element of the learned/predicted series

y_r = the current element of the real series

This evaluation of the forecasting accuracy using RMSE is used in our study to determine performance on tests on unknown data for all three methods. The developing of the proposed methods includes testing of their learning capabilities of training data and testing the prediction capacities. Both of these procedures use an additional measure to assign a more significant weight to the end of the time series that coincides with the beginning of what the prediction will be. In other words, we propose the following scenario: a particular network configuration is considered suitable if the error is within the limits set for the training and additional data, forcing the error to be

reduced by increasing the weights from the beginning of the time series to the end of it. The weighted average weighted square root formula for calculating the error is given by the following equation

$$\text{WRMSE} = \sqrt{(\sum_{i=1}^n w_i)^{-1} \sum_{i=1}^n w_i (y_p - y_r)^2} \quad (8)$$

where n , y_p and y_r have the same significance of (7) and w_i is the weights vector.

5 Experimental results

For this study, the time series stationarity check was performed with the Kwiatkowski-Phillips-Schmidt-Shin test, finding themselves to be non-stationary.

To determine the delay (optimal delay) the PACF function was used, the results of which are presented in the table below:

Lag	0	1	2	3	4	5	6	7	8	9
CHF	1.0000	0.9906	-0.0754	0.0921	0.0460	-0.0018	0.0408	-0.0386	0.0027	0.0631
USD	1.0000	0.9939	0.0087	0.0475	0.0206	-0.0117	-0.0025	-0.0416	-0.1304	-0.0617

In both cases, the sudden fall occurs after lag 1, which means we can consider the two series an optimal delay of 2.

We started with an arbitrary values for the conditions and decrease its successive and moderate. For each run of the program we save the values for the train error (RMSE1), for the test error on additional data (RMSE2) and for the test error on the new data.

Model 1. In the first case, we will test the standard NARX model by dividing the data to 70% for training, validation and testing, and the prediction will be done on the remaining of 30%. The best result obtained by the network in represented in Figure 7.

The results are presented in the table below:

	R1	R2	R3	R4	R5
RMSE1	0.0266	0.0192	0.0197	0.0163	0.0192
RMSE2	0.0272	0.0269	0.0234	0.0229	0.0178
RMSE3	0.0627	0.0279	0.0285	0.0263	0.0277
initial err1	0.0280	0.0260	0.0220	0.0200	0.0180
initial err2	0.0300	0.0280	0.0250	0.0230	0.0200
number of while1 iterations	39	34	102	64	271
number of while2 iterations	15	12	58	10	68

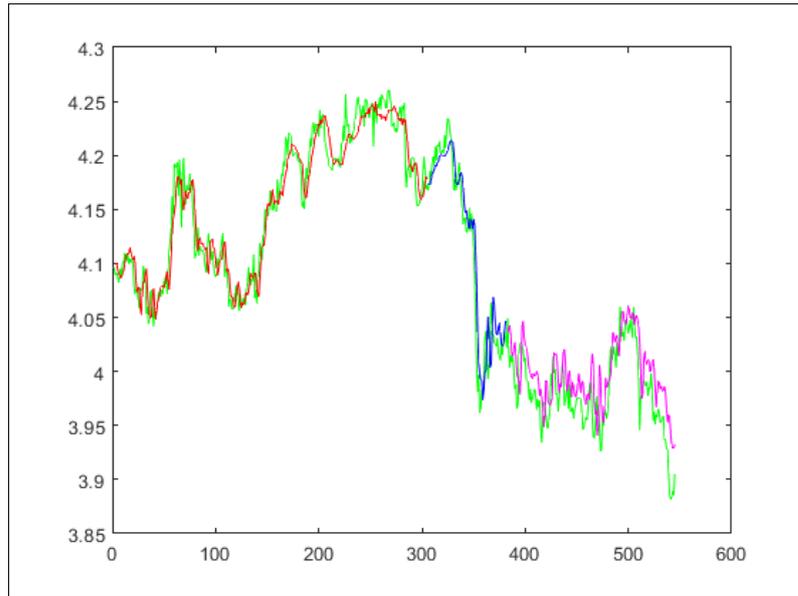


Fig. 7. Results for the first model. RMSE = 0.0263

Model 2. The idea behind this model is to conserve the properties of the neural network (weights and biases) and try to improve its generalization capabilities by training it again

	R1	R2	R3	R4	R5
RMSE1	0.0171	0.0208	0.0202	0.0194	0.0179
RMSE2	0.0295	0.0261	0.0246	0.0211	0.0198
RMSE3	0.0598	0.0436	0.0386	0.0291	0.0227
initial err1	0.0280	0.0260	0.0220	0.0200	0.0180
initial err2	0.0300	0.0280	0.0250	0.0230	0.0200
number of while1 iterations	223	198	498	683	508
number of while2 iterations	39	26	56	166	78

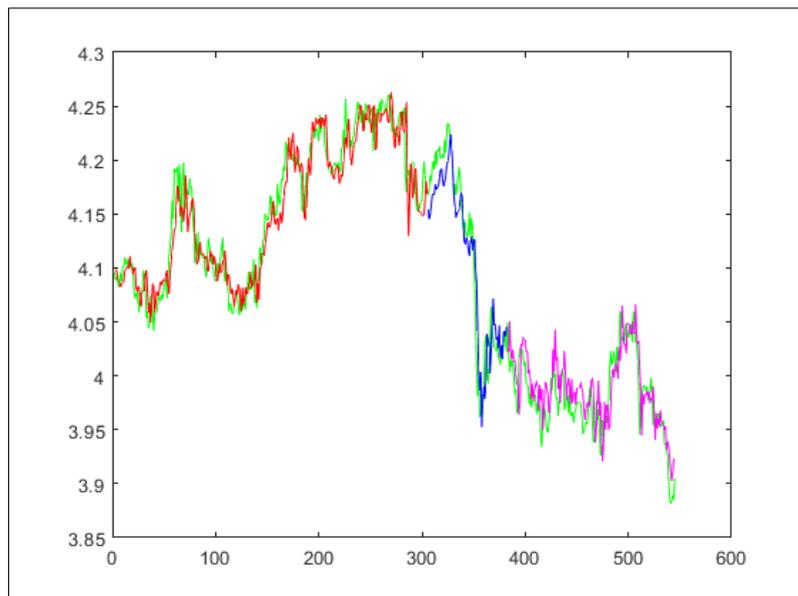


Fig. 8. Results for the second model. RMSE = 0.0227

The prediction obtained with the second model is more accurate overall, the error has

decreased by 13.7% compared to the first model.

Model 3. The third proposed model for the

	R1	R2	R3	R4	R5
RMSE1	0.0174	0.0258	0.0243	0.0192	0.0178
RMSE2	0.0254	0.0262	0.0204	0.0212	0.0192
RMSE3	0.0389	0.0381	0.0265	0.0223	0.0199
initial err1	0.0280	0.0260	0.0220	0.0200	0.0180
initial err2	0.0300	0.0280	0.0250	0.0230	0.0200
number of while1 iterations	2	79	322	673	792
number of while2 iterations	2	65	203	542	611

problem consists in conserving the most of the properties of the neural network (weights and biases) in order to re-establish the same network.

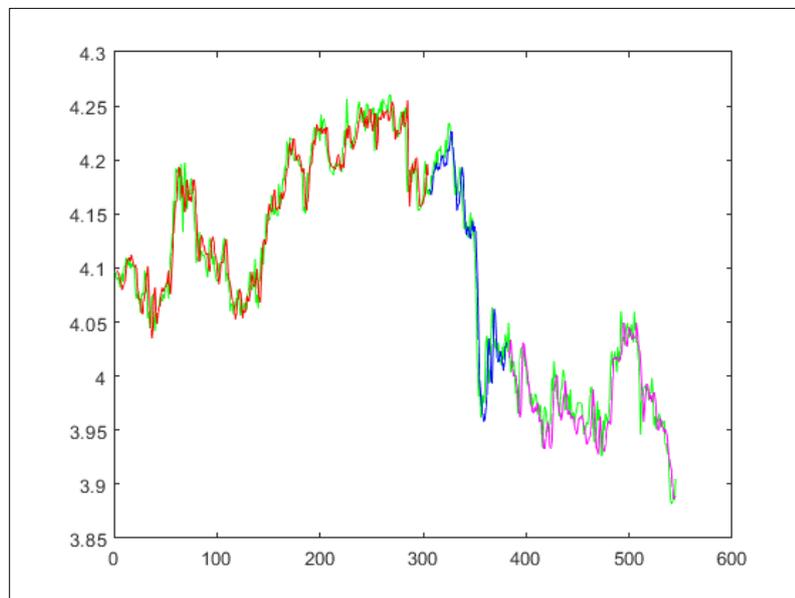


Fig. 9. Results for the third model. RMSE = 0.0199

The prediction obtained with the third model is more accurate overall and we can observe a very good fitting at the end of prediction. The error has decreased by 24.7% compared to the first model and by 12.3% compared to the second model.

6 Conclusions

NARX neural networks have real potential in learning and understanding stock market developments. Our main goal is to optimally configure them to create more and more performing versions. An idea of future work emerged from the present study is the ability to determine at the level of weights which ones bring for a network the best quality for a version of it and their dynamic preservation according to this criterion. Moreover, since

the conducted tests proved good results, they entail the hope that further and possibly more sophisticated extensions can be expected to improve it. Among several possible extensions, some work is still in progress concerning the use of different output functions for the hidden and output neurons.

References

- [1] A. Sorjamaa, et al., *Methodology for long-term prediction of time series*, Neurocomputing (2007), doi:10.1016/j.neucom.2006.06.015.
- [2] Jose A. Scheinkman, Blake LeBaron, *Non-linear Dynamics and Stock Returns*, The Journal of Business, Vol. 62. No. 3 (1989), pp. 311-337.

- [3] Cătălina-Lucia COCIANU, Hakob GRI-GORYAN, Machine Learning Techniques for Stock Market Prediction. A Case Study of OMV Petrom, Economic Computation and Economic Cybernetics Studies and Research, Issue 3/2016, Vol. 50, pp. 63-82.
- [4] Malkareddy, Krishna & Boiroju, Naveen. (2008). *Forecasting Foreign Exchange Rates Using Time Delay Neural Networks*, 267-273.
- [5] Abou-Zaid, Ahmed & Stokes, Adam. (2012). *Forecasting foreign exchange rates using artificial neural networks: A trader's approach*. International Journal of Monetary Economics and Finance. 5. 370-394. 10.1504/IJMEF.2012.052502.
- [6] Chaudhuri T. D. et al., *Artificial Neural Network and Time Series Modeling Based Approach to Forecasting the Exchange Rate in a Multivariate Framework*, Journal of Insurance and Financial Management, Vol. 1, Issue 5 (2016) 92-123
- [7] A.Y. Hatata, M. Eladawy, *Prediction of the true harmonic current contribution of nonlinear loads using NARX neural network*, Alexandria Eng. J. (2017).
- [8] Eugen Diaconescu, *The use of NARX Neural Networks to predict Chaotic Time Series*, Wseas Transactions on Computer Research, Volume 3, (2008), ISSN: 1991-8755.
- [9] Beagle, M., Hagan, M. and Demuth, H., (2017) *Neural Network Toolbox: User's Guide*, The MathWorks, Inc., Natick, MA.
- [10] Sadig Mammadli, Financial time series prediction using artificial neural network based on Levenberg-Marquardt algorithm, 9th International Conference on Theory and Application of Soft Computing, Computing with Words and Perception, ICSCCW 2017, Hungary.
- [11] Interactive Exchange Rate Database, National Bank of Romania (2018). Online link: <http://www.bnro.ro/Baza-de-date-interactiva-604.aspx>.
- [12] Yan Xiu, Wei Zhang, *Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature*, Geosci. Model Dev., 7, (2014), pg. 1247–1250, DOI:10.5194/gmd-7-1247-2014.



Catalina-Lucia COCIANU, Professor, PhD, currently working with Bucharest University of Economic Studies, Faculty of Cybernetics, Statistics and Informatics, Department of Informatics and Cybernetics in Economy. Competence areas: machine learning, statistical pattern recognition, digital image processing. Research in the fields of pattern recognition, data mining, signal processing. Author of 20 books and more than 100 papers published in national and international journals and conference proceedings.



Mihai-Șerban AVRAMESCU graduated the Faculty of Cybernetics, Statistics and Economic Informatics in 2015. He has graduated the Economic Informatics Master program organized by the Bucharest University of Economic Studies, in 2017. In present he is earning his PhD degree in Economic Informatics with the Doctor's Degree Thesis: "Analyzing and predicting economic data using Machine Learning techniques".