An Artificial Neural Network for Data Forecasting Purposes

Cătălina-Lucia COCIANU, Hakob GRIGORYAN
Bucharest University of Economic Studies, Bucharest, Romania
catalina.cocianu@ie.ase.ro, grigoryanhakob90@yahoo.com

Considering the fact that markets are generally influenced by different external factors, the stock market prediction is one of the most difficult tasks of time series analysis. The research reported in this paper aims to investigate the potential of artificial neural networks (ANN) in solving the forecast task in the most general case, when the time series are non-stationary. We used a feed-forward neural architecture: the nonlinear autoregressive network with exogenous inputs. The network training function used to update the weight and bias parameters corresponds to gradient descent with adaptive learning rate variant of the backpropagation algorithm. The results obtained using this technique are compared with the ones resulted from some ARIMA models. We used the mean square error (MSE) measure to evaluate the performances of these two models. The comparative analysis leads to the conclusion that the proposed model can be successfully applied to forecast the financial data.
Keywords: *Neural Network, Nonlinear Autoregressive Network, Exogenous Inputs, Time Series, ARIMA Model*

1 Introduction

Predicting stock price index and its movement has been considered one of the most challenging applications of time series prediction. According to the efficient market theory proposed in [1], the stock price follows a random path and it is practically impossible to make a particular long-term global forecasting model based on historical data. The ARIMA and ANN techniques have been successfully used for modelling and forecasting financial time series. Compared with ANN models, which are complex forecasting systems, ARIMA models are considered to be much easier techniques for training and forecasting.

An important feature of neural networks is the ability to learn from their environment, and, through learning to improve performance in some sense. One of the new trends is the development of specialized neural architectures together with classes of learning algorithms to provide alternative tools for solving feature extraction, data projection, signal processing, and data forecasting problems respectively [2].

Artificial neural networks have been widely used for time series forecasting and they have shown good performance in predicting stock market data. Chen et al., [3], introduced a neural network model for time series forecasting based on flexible multi-layer feed-forward architecture. F. Giordano et al., [4], used a new neural network-based method for prediction of non-linear time series. Lin et al.,[5], applied artificial neural network to predict Taiwan stock index option price. Z. Liao et al., [6], applied stochastic time effective neural network to develop a forecasting model of global stock index. Mohamed et al., [7], used neural networks to forecast the stock exchange movements in Kuwait Stock Exchange. Cai et al., [8], used neural networks for predicting large scale conditional volatility and covariance of financial time series.

In the recent years, a series of studies have been conducted in the field of financial data analysis using ARIMA models for financial time series prediction. Meyler et al, [9] used ARIMA models to forecast Irish Inflation. Contreras et al, [10] predicted next day electricity prices using ARIMA methodology. V. Ediger et al, [11] used ARIMA model to forecast primary energy demand by fuel in Turkey. Datta [12] used the same Box and Jenkins methodology in forecasting inflation rate in the Bangladesh. Al-Zeaud [13] have used ARIMA model for

DOI: 10.12948/issn14531305/19.2.2015.04

modelling and predicting volatility in banking sector.
The paper is organized as follows. In the second section of the paper, we briefly present the ARIMA model for prediction. Next, the nonlinear autoregressive network with exogenous inputs aiming to forecast the closing price of a particular stock is presented. The ANN-based strategy applied for data forecasting is analysed against the ARIMA model, and a comparative analysis of these models is described in the fourth section of the paper. The conclusions regarding the reported research are presented in the final part of the paper.

2 ARIMA Model

The Auto-Regressive Integrated Moving Average (ARIMA) model or Box-Jenkins methodology [14] is a statistical analysis model. It is mainly used in econometrics and statistics for time-series analysis. ARIMA model uses time series data to predict future points in the series. A non-seasonal ARIMA model is denoted by ARIMA(p, d, q), where p, d, q are non-negative integers referring to the Auto-Regressive (AR), Integrated (I) and Moving Average (MA) parameters, respectively.
ARIMA model generalizes the Auto-Regressive Moving Average ARMA (ARMA) model. Let $\phi_p$and Θ_q be the autoregressive and moving average polynomials respectively,

$$\phi_p(x) = 1 - \varphi_1 x - \cdots - \varphi_p x^p$$
$$\Theta_q(x) = 1 + \theta_1 x + \cdots + \theta_q x^q$$

The process $\{X_t, t = 0, \pm 1, \pm 2, \ldots\}$ is an ARMA(p,q) process if $\{X_t\}$ is stationary and for each *t* the following relation holds

$$\phi_p(B)X_t = \Theta_q(B)W_t \qquad (1)$$

where B is the backward shift operator defined by

$$B^i X_t = X_{t-i}, \quad i = 0, \pm 1, \pm 2, \ldots$$

and $\{W_t\}$ is white noise $\sim N(0, \sigma^2)$. The process is causal if there exists a sequence of constants $\{\alpha_i\}$ such that $\sum_{i=0}^{\infty} \alpha_i < \infty$ and

$$X_t = \sum_{i=0}^{\infty} \alpha_i W_{t-i}, t = 0, \pm 1, \pm 2, \ldots \qquad (2)$$

The process $\{X_t, t = 0, \pm 1, \pm 2, \ldots\}$ is an ARIMA(p,d,q) process if $Y_t = \nabla^d X_t = (1 - B)^d X_t$ is a causal ARMA(p,q) process, that is

$$\phi_p(B)(1 - B)^d X_t = \Theta_q(B)W_t \qquad (3)$$

where $\phi_p$and Θ_q are the autoregressive and moving average polynomials respectively, B is the backward shift operator and $\{W_t\}$ is the white noise.
The problem of predicting ARIMA processes can be solved using extensions of the prediction techniques developed for ARMA processes. One of the most commonly used methods in forecasting ARMA(p,q) processes is the class of the recursive techniques for computing best linear predictors (the Durbin-Levison algorithm, the Innovations algorithm etc.). In the following we describe the recursive prediction method using the Innovation algorithm. [15]
Let $\{X_t, t = 0, \pm 1, \pm 2, \ldots\}$ be a zero mean stochastic process and $K(i,j)$ its autocorrelation function. We denote by $H_n = \overline{\text{sp}}\{X_1, X_2, \ldots, X_n\}$, $n \geq 1$ the closed linear subspace generated by $X_1, X_2, \ldots, X_n$ and let

$$\hat{X}_{n+1} = \begin{cases} 0, & n = 0 \\ P_{H_n} X_{n+1}, & n \geq 1 \end{cases} \qquad (4)$$

where $P_{H_n} X_{n+1}$ stands for the projection of X_{n+1} on H_n.
Since $\hat{X}_1 = 0$, we get

$$H_n = \overline{sp}\{X_1 - \hat{X}_1, X_2 - \hat{X}_2, \ldots X_n - \hat{X}_n\}, n \geq 1 \quad (5)$$

and

DOI: 10.12948/issn14531305/19.2.2015.04

$$\hat{X}_{n+1} = \sum_{j=1}^{n} \theta_{nj}\left(X_{n+1-j} - \hat{X}_{n+1-j}\right) \qquad (6)$$

where the coefficients $\{\theta_{nj},\ j=1,2,\ldots,n;\ v_n\}$ are given by a recursive scheme [15].

Let $v_n = \left\| X_{n+1} - \hat{X}_{n+1} \right\|^2$. If, for any $n \geq 1$, $[K(i,j)]_{i,j=1,\ldots,n}$ is a non-degenerated matrix, then we get

$$\begin{cases} v_0 = K(1,1) \\ \theta_{n,n-k} = \dfrac{K(n+1,k+1) - \sum_{j=0}^{k} \theta_{k,k-j}\theta_{n,n-j}v_j}{v_k}, 0 \leq k \leq n-1 \\ v_n = K(n+1,n+1) - \displaystyle\sum_{j=0}^{k} \left(\theta_{n,n-j}\right)^2 v_j \end{cases} \qquad (7)$$

The equation (7) gives the coefficients of the innovations, $X_j - \hat{X}_j, 1 \leq j \leq n$ in the orthogonal expansion (6), which is simple to use and, in the case of ARMA(p,q) processes, can be further simplified [15].

3 The ANN-Based Technique for Forecasting the Closing Price of a Stock

The nonlinear autoregressive network with exogenous inputs aiming to forecast the closing price of a particular stock is presented in the following.

We assume that $\mathrm{Y_t}$ is the stock closing value at the moment of time *t*. For each *t*, we denote by $\mathrm{X_t} = \left(\mathrm{X_t}(1), \mathrm{X_t}(2), \ldots, \mathrm{X_t}(n)\right)^T$ the vector whose entries are the values of the indicators significantly correlated to $\mathrm{Y_t}$, that is the correlation coefficient between $\mathrm{X_t}(i)$ and $\mathrm{Y_t}$ is greater than a certain threshold value, for $\mathrm{i} = 1,2, \ldots, n$.

The neural model used in our research is a dynamic network. The direct method was used to build the model of prediction of the stock closing value, which is described as follows.

$$\hat{\mathrm{Y}}_{(\mathrm{t+p})} = \mathrm{f}_{\mathrm{ANN}}\left(Y_t^{(d)}, X_t^{(d)}\right) \qquad (8)$$

$$Y_t^{(d)} = \{\mathrm{Y_t}, \mathrm{Y_{t-1}}, \mathrm{Y_{t-2}}, \ldots, \mathrm{Y_{t-d}}\} \qquad (9)$$

$$X_t^{(d)} = \{\mathrm{X_t}, \mathrm{X_{t-1}}, \mathrm{X_{t-2}}, \ldots, \mathrm{X_{t-d}}\} \qquad (10)$$

where $\hat{\mathrm{Y}}_{(\mathrm{t+p})}$ is the forecasted value of the stock price for the prediction period p and d is the delay expressing the number of pairs $(\mathrm{X_k}, \mathrm{Y_k}), k = t, t-1, \ldots, t-d$ used as input of the neural model. In our model, we consider $\mathrm{p} = 1$.

The considered delay has significant influence on the training set and prediction process. We use correlogram to choose the appropriate window size for our neural networks. We need to eliminate the lags where the Partial Autocorrelation Function (PACF) is statistically irrelevant [16].

The nonlinear autoregressive network with exogenous inputs (NARX) is a recurrent dynamic network, with feedback connections encompassing multiple layers of the network. The scheme of NARX is depicted in Figure 1.

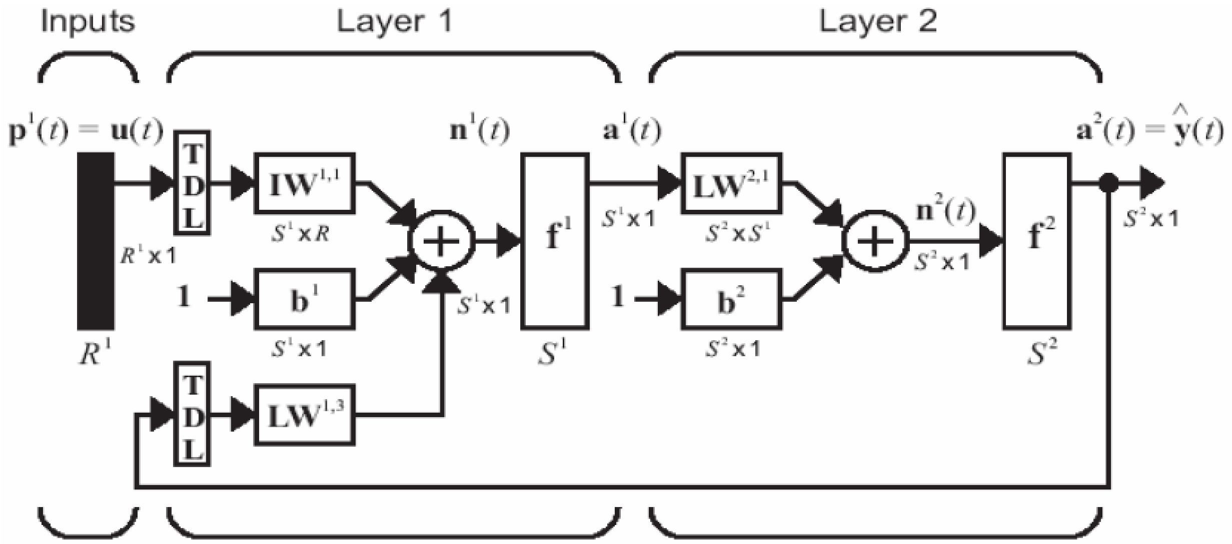


Fig. 1. The architecture of nonlinear autoregressive network with exogenous inputs (NARX)

The output of the NARX network can be considered an estimate of the output of a certain nonlinear dynamic system. Since the actual output is available during the training of the network, a series-parallel architecture is created [17], where the estimated target is replaced by the actual output. The advantages of this model are twofold. On the one hand, the inputs used in the training phase are more accurate and, on the other hand, since the resulting network has feed-forward architecture, a static backpropagation type of learning can be used.

The NARX network is used here as a predictor, the forecasting formula being given by

$$y(t) = f\Big(y(t-1), y(t-2), \dots, y\big(t - n_y\big), u(t-1), u(t-2), \dots, u(t-n_u)\Big) \quad (11)$$

where $y(t)$ is the next value of the dependent output variable y and u is externally determined variable that influences y. The "previous" values $y(t-1), y(t-2,\dots, yt-ny$ of y and $ut-1, ut-2,\dots, ut-nu$ of u are used to predict $y(t)$, the future value of y.

An example of this series-parallel network is depicted in Figure 2, where d=2, n=10 and the number of neurons in the hidden layer is 24.

The activation functions of the neurons in the hidden and output layers respectively can be defined in many ways. In our tests, we took the logistic function (12) to model the activation functions of the neurons belonging to the hidden layers, and the unit function to model the outputs of the neurons belonging to the output layers.

$$logsig(x) = \big(1 + exp(x)\big)^{-1} \quad (12)$$

DOI: 10.12948/issn14531305/19.2.2015.04

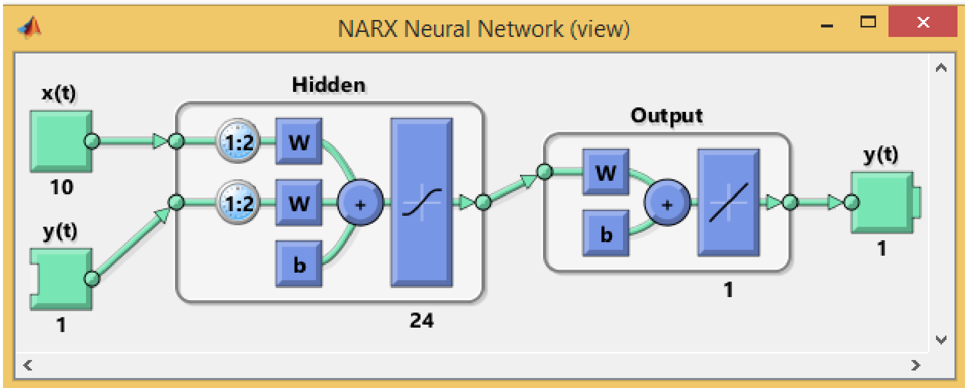


Fig. 2. Example of a series-parallel network

After the training step, the series-parallel architecture is converted into a parallel configuration, in order to perform the multi-step-ahead prediction task. The corresponding neural network architecture is presented in Figure 3.

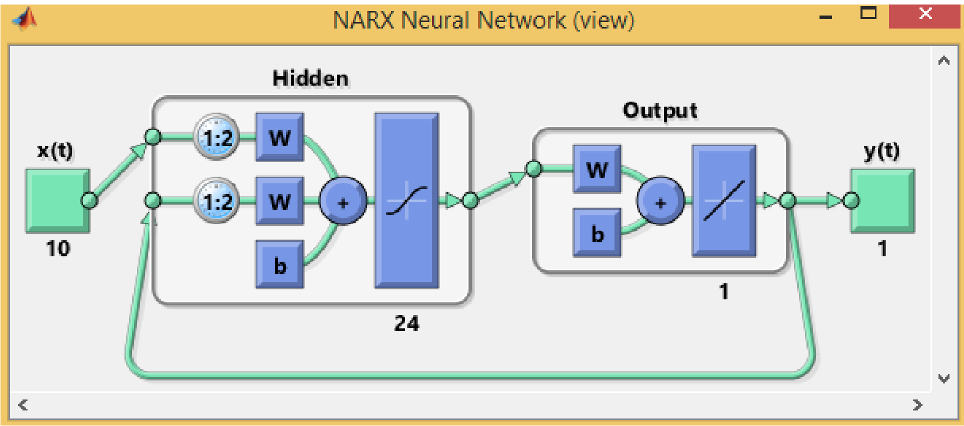


Fig. 3. Example of a parallelized network

We use the standard performance function, defined by the mean sum of squares of the network errors. The data division process is cancelled to avoid the early stopping.

The network training function used to update the weight and bias parameters corresponds to gradient descent with adaptive learning rate variant of the backpropagation algorithm. The main advanced of this method, proposed by V.P. Plagianakos et al. in [18], consists in improving the convergence rate of the learning process.

In the following we consider the class of gradient-based learning algorithms, whose general updating rule is given by

$$w^{k+1} = w^k + \eta_k d_k, \quad k = 0,1,2,\dots,n \quad (13)$$

where w^k stands for the current point, d_k is the search direction, and η_k is the steplength.

Let us denote by E the error function, defined in terms of the sum of squared differences error function over the training set. The backpropagation gradient-based algorithm with adaptive learning rate results by minimizing the error function E.

In order to provide two-point approximation to the secant equation underlying quasi-Newton methods, the learning rate defined at each epoch k is

$$\eta_k = \frac{\langle w^k - w^{k-1}, w^k - w^{k-1}\rangle}{\langle w^k - w^{k-1}, \nabla E(w^k) - \nabla E(w^{k-1})\rangle} \quad (14)$$

In this case, the gradient-based learning method possibly overshoots the optimum point or even diverges. In order to overcome

DOI: 10.12948/issn14531305/19.2.2015.04

this problem, a maximum growth factor μ is introduced, and the learning rate is computed according to the following equation

$$\lambda_k = \begin{cases} \eta_k, & \left|\frac{\eta_k}{\eta_{k-1}}\right| \leq \mu \\ \mu\, \eta_{k-1}, & \text{otherwise} \end{cases} \quad (15)$$

If the considered search direction is defined by

$$d^k = -\nabla E(w^k) \quad (16)$$

then the obtained updating rule of the backpropagation gradient-based algorithm with adaptive learning rate is [18]

$$w^{k+1} = w^k - \lambda_k \nabla E(w^k) \quad (17)$$

In our work, the number of neurons in the hidden layer is set according to the following equation [19]

$$2\sqrt{(m+2)N} \quad (18)$$

where m stands for the number of the neurons in the output layer and N is the dimension of input data.

4 Experimental Results

We tested the proposed model on 300 samples dataset. The samples are historical weekly observations of a set of variables *S*, between 3/1/2009 and 11/30/2014. The set *S* contains the opening, closing, highest and lowest price respectively of SNP stock from Bucharest Stock Exchange, and seven indicators obtained from technical and fundamental analysis of the stock market.

The correlogram shows that for all variables PACF function drops immediately after the 2nd lag. This means that window size for all variables could be set to 2.

In our tests, we used 200 samples for training purposes and 100 unseen yet samples for data forecasting.

The neural network parameters are determined based on the following process:

REPEAT
1. Initialize the parameters of the NN.
2. Train the NN using the set of training samples in 6000 epochs.
UNTIL the overall forecasting error computed on the already trained data in terms of MSE measure is less than a certain threshold value.

In our tests, the threshold value is set to 10^{-3}.

If we denote by $T = (T(1), T(2), \dots, T(nr))$ the vector of target values and by $P = (P(1), P(2), \dots, P(nr))$ the vector whose entries correspond to the predicted values, the MSE error measure is defined by

$$MSE(T,P) = \frac{1}{nr}\sum_{i=1}^{nr}\big(T(i) - P(i)\big)^2 \quad (19)$$

The results obtained using the above mentioned technique are reported in the following. The overall forecasting error computed on the already trained data prediction is 0.00035. The regression coefficient computed on the already trained data and the data fitting are presented in Figure 4. The network predictions versus actual data in case of already trained samples are illustrated in Figure 5. The overall forecasting error computed on the new data prediction is 0.0012.The network predictions versus actual data in case of new samples are illustrated in Figure 6.

DOI: 10.12948/issn14531305/19.2.2015.04

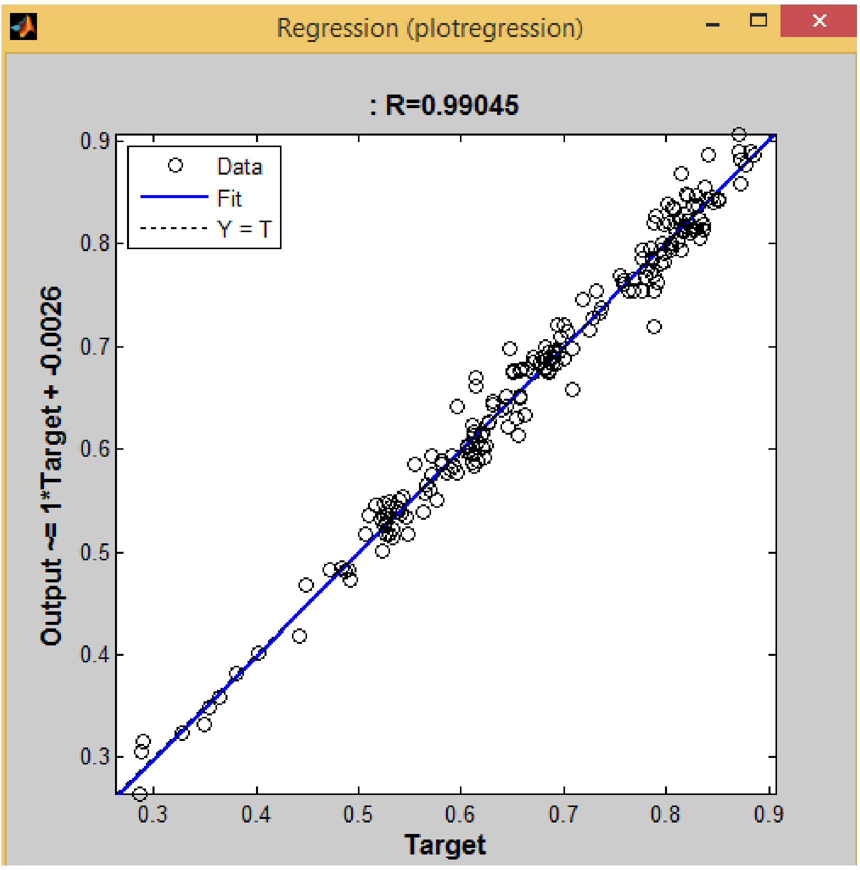


Fig. 4. The regression coefficient and data fitting in case of already trained samples

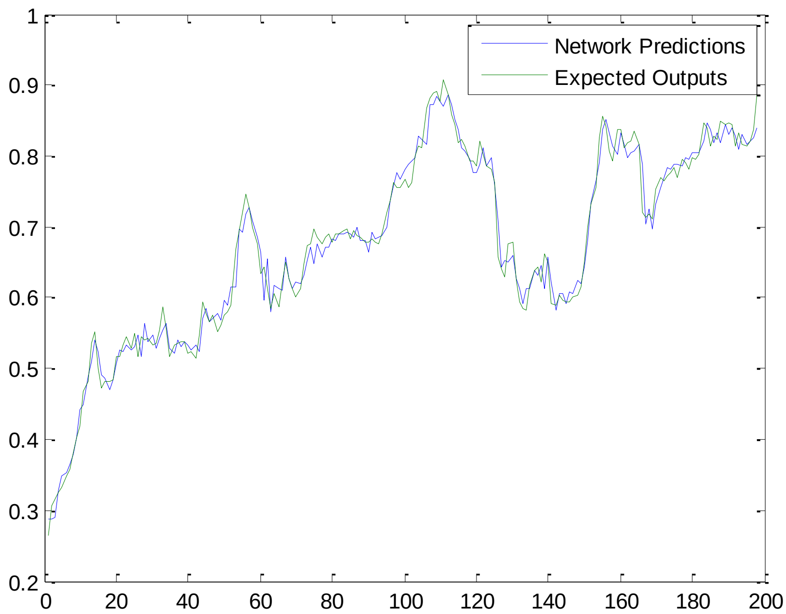


Fig. 5. Predictions versus actual data in case of already trained samples

DOI: 10.12948/issn14531305/19.2.2015.04

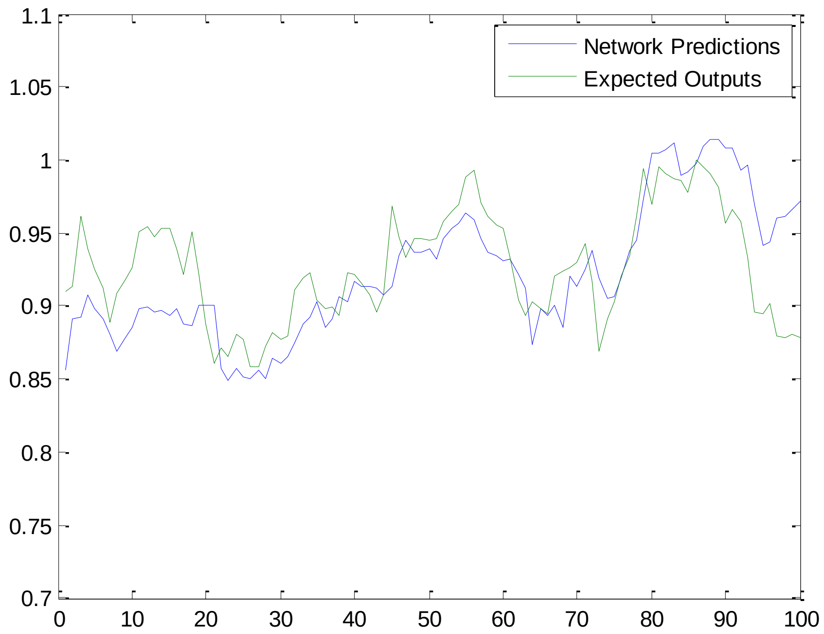


Fig. 6. The network predictions versus actual data in case of new samples

The error histogram in case of new data set is depicted in Figure 7.

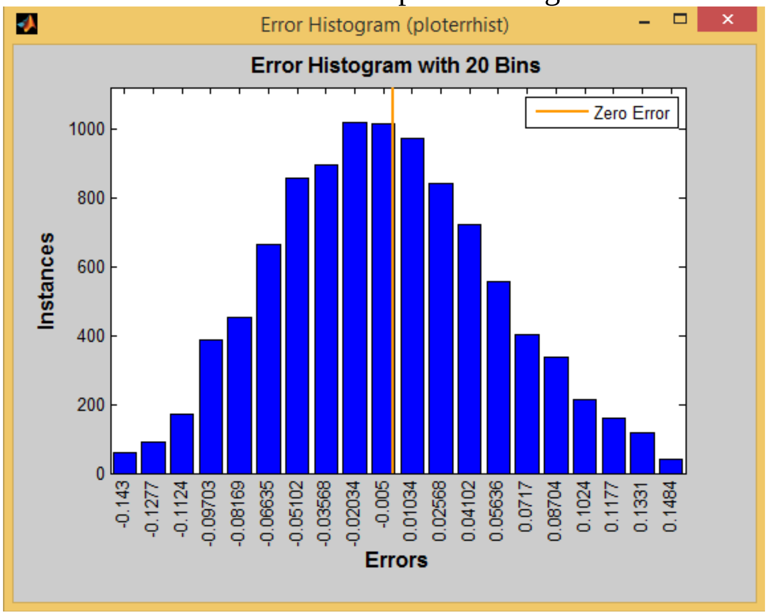
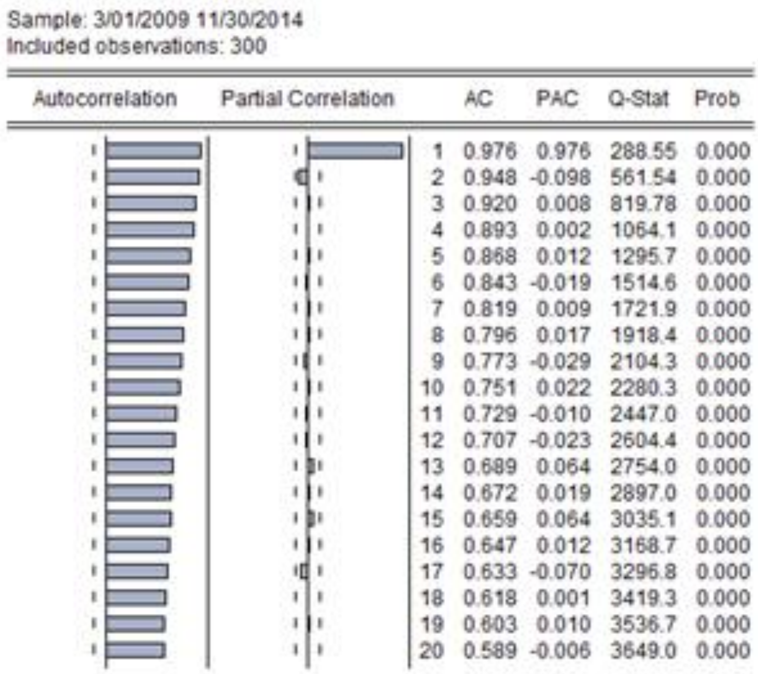


Fig. 7. The error histogram in case of new samples

We developed a comparative analysis of the neural network-based approaches against the well-known ARIMA forecasting method.

First, we used Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) to establish whether the time series are stationary or not. In case of stationary time series, ACF decays rapidly. Since the computed values of ACF indicated that the function decays very slowly, we concluded that the considered time series are non-stationary. The corresponding correlogram is depicted in Figure 8.

DOI: 10.12948/issn14531305/19.2.2015.04

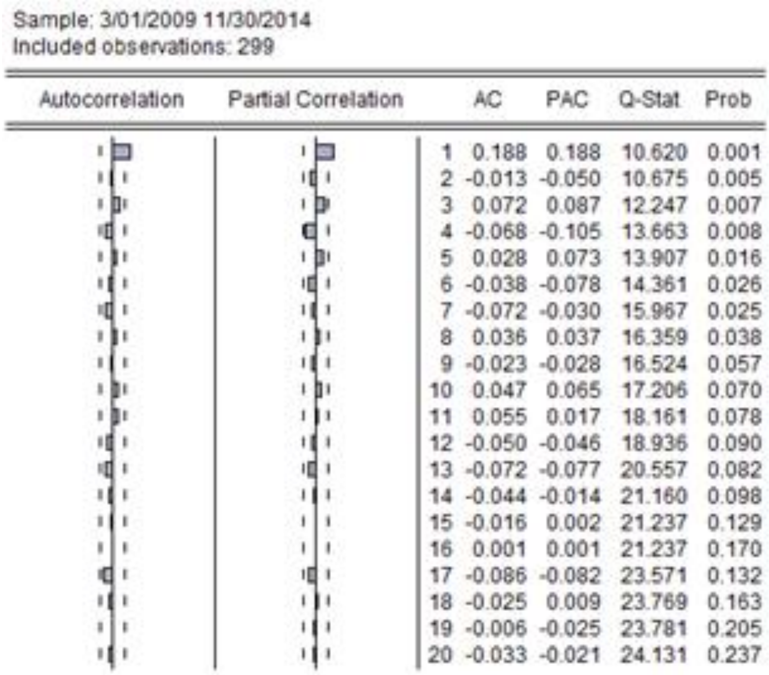


Sample: 3/01/2009 11/30/2014
Included observations: 300

Lag	AC	PAC	Q-Stat	Prob
1	0.976	0.976	288.55	0.000
2	0.948	-0.098	561.54	0.000
3	0.920	0.008	819.78	0.000
4	0.893	0.002	1064.1	0.000
5	0.868	0.012	1295.7	0.000
6	0.843	-0.019	1514.6	0.000
7	0.819	0.009	1721.9	0.000
8	0.796	0.017	1918.4	0.000
9	0.773	-0.029	2104.3	0.000
10	0.751	0.022	2280.3	0.000
11	0.729	-0.010	2447.0	0.000
12	0.707	-0.023	2604.4	0.000
13	0.689	0.064	2754.0	0.000
14	0.672	0.019	2897.0	0.000
15	0.659	0.064	3035.1	0.000
16	0.647	0.012	3168.7	0.000
17	0.633	-0.070	3296.8	0.000
18	0.618	0.001	3419.3	0.000
19	0.603	0.010	3536.7	0.000
20	0.589	-0.006	3649.0	0.000

Fig. 8. The correlogram of the available data

In order to tune the differencing parameter of the ARIMA model, the first order and the second order differenced series respectively have been computed. The corresponding correlogram of the first order differenced series is presented in Figure 9. Since the values of ACF in case of using the first order differenced series are quite small, we concluded that the differencing parameter of ARIMA model should be set to the value 1.



Sample: 3/01/2009 11/30/2014
Included observations: 299

Lag	AC	PAC	Q-Stat	Prob
1	0.188	0.188	10.620	0.001
2	-0.013	-0.050	10.675	0.005
3	0.072	0.087	12.247	0.007
4	-0.068	-0.105	13.663	0.008
5	0.028	0.073	13.907	0.016
6	-0.038	-0.078	14.361	0.026
7	-0.072	-0.030	15.967	0.025
8	0.036	0.037	16.359	0.038
9	-0.023	-0.028	16.524	0.057
10	0.047	0.065	17.206	0.070
11	0.055	0.017	18.161	0.078
12	-0.050	-0.046	18.936	0.090
13	-0.072	-0.077	20.557	0.082
14	-0.044	-0.014	21.160	0.098
15	-0.016	0.002	21.237	0.129
16	0.001	0.001	21.237	0.170
17	-0.086	-0.082	23.571	0.132
18	-0.025	0.009	23.769	0.163
19	-0.006	-0.025	23.781	0.205
20	-0.033	-0.021	24.131	0.237

Fig. 9. The correlogram of the first order differenced series

The parameters of ARIMA model related to AR(p) and MA(q) processes were tuned based on the following criteria: relatively small values of BIC (Bayesian Information Criterion), relatively high values of adjusted R^2 (coefficient of determination) and

DOI: 10.12948/issn14531305/19.2.2015.04

relatively small standard error of regression (SER). The results of our tests are summarized in Table 1. According to these results, the best model from the point of view of the above mentioned criteria is ARIMA(1,1,1) model. We concluded that the best fitted models are ARIMA(1,1,0) and ARIMA(1,1,1).

Table 1. ARIMA models

ARIMA model	BIC	Adjusted R^2	SER
(1,1,0)	-5.292201	0.987351	0.015247
(1,1,1)	-5.547453	0.990408	0.013278
(0,1,1)	-2.283686	0.754100	0.068656
(0,1,0)	-1.017242	0.108715	0.130709

The overall forecasting error computed on the new data prediction is 0.0077 in case of using ARIMA(1,1,0) model, and 0.0096 in case of using ARIMA(1,1,1) model. The results of forecasting are illustrated in Figure 10.

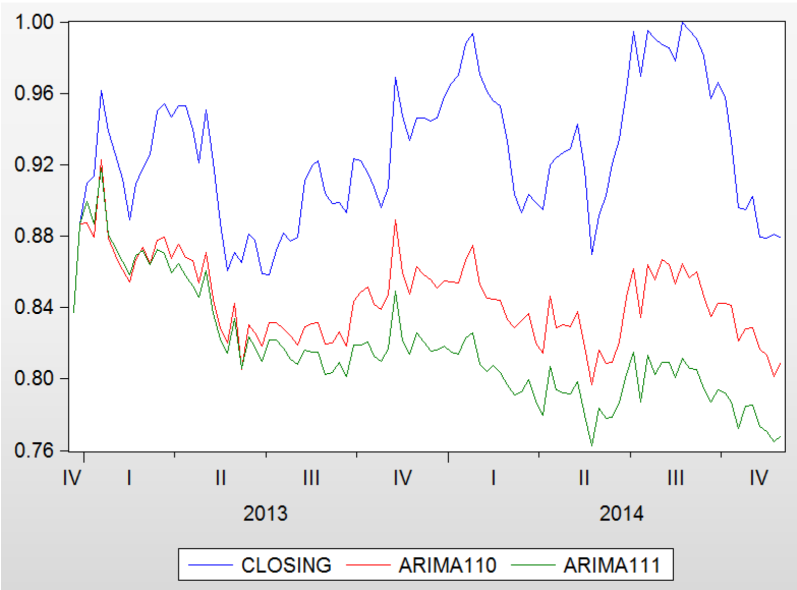


Fig. 10. Predicted values of ARIMA(1,1,0) and ARIMA(1,1,1) models versus actual data

5 Conclusions

The research reported in this paper focuses on a comparative analysis of NARX neural network against standard ARIMA models. The study was developed on a dataset consisting in 300 historical weekly observations of a set of variables, between 3/1/2009 and 11/30/2014. The results obtained using the proposed neural approach proved better results from the point of view of MSE measure. The obtained results are encouraging and entail future work toward extending the study in case of using alternative neural models.

Acknowledgement

A shorter version of this paper was presented at the 14th International Conference on Informatics in Economy (IE 2015), May 1-3, 2015.

References

[1] E.F. Fama, *Efficient capital markets: A review of theory and empirical work*, The Journal of Finance, 25 (2) (1970), pp. 383–417.

[2] C. Cocianu, L. State, and P. Vlamos, *Neural Implementation of a Class of PCA Learning Algorithms*, Economic Computation and Economic Cybernetics

DOI: 10.12948/issn14531305/19.2.2015.04

Studies and Research, Vol. 43, No 3/2009, pp. 141-154, 2009

[3] Y. Chen, B. Yang, J. Dong, and A. Abraham, *Time-series forecasting using flexible neural tree model*, Information Sciences, vol. 174, no. 3-4, pp. 219–235, 2005.

[4] F. Giordano, M. La Rocca, and C. Perna, *Forecasting nonlinear time series with neural network sieve bootstrap*, Computational Statistics and Data Analysis, vol. 51, no. 8, pp. 3871–3884, 2007.

[5] C.T. Lin, H.Y. Yeh, *Empirical of the Taiwan stock index option price forecasting model – Applied artificial neural network*, Applied Economics, 41 (15) (2009), pp. 1965–1972.

[6] Z. Liao, J. Wang, *Forecasting model of global stock index by stochastic time effective neural network*, Expert Systems with Applications, 37 (1) (2009), pp. 834–841.

[7] M.M. Mohamed, *Forecasting stock exchange movements using neural networks: empirical evidence from Kuwait*, Expert Systems with Applications, vol. 27, no. 9, pp.6302–6309, 2010.

[8] X. Cai, G. Lai, X. Lin, *Forecasting large scale conditional volatility and covariance using neural network on GPU*, The Journal of Supercomputing., 63 (2013), pp. 490–507

[9]A. Meyler, G. Kenny and T. Quinn, "*Forecasting Irish Inflation using ARIMA Models"*, Central Bank of Ireland Research Department, Technical Paper, 3/RT/1998.

[10] J. Contreras, R. Espinola, F.J. Nogales, and A.J. Conejo, *"ARIMA models to predict Next Day Electricity Prices"*, IFEE Transactions on power system, Vol. 18, No. 3, 2003, pp 1014-1020.

[11] V. Ediger, S. Akar, *ARIMA forecasting of primary energy demand by fuel in Turkey*, Energy Policy, 35 (2007), pp. 1701–1708

[12] Datta, K. (2011) "*ARIMA Forecasting of Inflation in the Bangladesh Economy"*, The IUP Journal of Bank Management, Vol. X, No. 4, pp. 7-15.

[13] Hussein Ali Al-Zeaud. (2011), *Modelling and Forecasting Volatility Using ARIMA Models*, European Journal of Economics, Finance and Administrative Studies. (35): 109-125.

[14] P. Box, G.M. Jenkins, Time series analysis: Forecasting and control, Holden-day Inc., San Francisco, CA (1976).

[15] P.J. Brockwell, R.A.Davis, *Time Seires: Theory and Methods*, 2nd Edition, Springer Series in Statistics, Springer-Verlag, 1991

[16] D.N. Gujarati, *Basic econometrics*, McGraw-Hill, New York (2003).

[17] Narendra, Kumpati S., Kannan Parthasarathy, *Learning Automata Approach to Hierarchical Multi objective Analysis*, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20, No. 1, January/February 1991, pp. 263–272.

[18] V.P. Plagianakos, D.G. Sotiropoulos, and M.N. Vrahatis, *An Improved Backpropagation Method with Adaptive Learning Rate*, University of Patras, Department of Mathematics,

Division of Computational Mathematics & Informatics, GR-265 00, Patras, Greece (1998).

[19] Fagner A. de Oliveira, Cristiane N. Nobre, Luis E. Zarate *Applying Artificial Neural Networks to prediction of stock price and improvement of the directional prediction index – Case study of PETR4, Petrobras, Brazil*, Journal of Expert Systems with Applications, 40, (2013), 7596–7606.

Catalina-Lucia COCIANU, Professor, PhD, currently working with Academy of Economic Studies, Faculty of Cybernetics, Statistics and Informatics, Department of Informatics in Economy. Competence areas:

DOI: 10.12948/issn14531305/19.2.2015.04

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.

Hakob GRIGORYAN has graduated the Faculty of Cybernetics of the State Engineering University of Armenia (Polytechnic) in 2011. In 2014 he graduated the Faculty of Informatics and Mathematics of the University of Bucharest with a specialization in Database and Web Technologies. At the present, he is earning his PhD degree in Economic Informatics at Bucharest University of Economic Studies, coordinated by Professor Catalina-Lucia Cocianu. His PhD thesis is “Machine Learning-Based Techniques for Financial Data Analysis and Forecasting Purposes”.

DOI: 10.12948/issn14531305/19.2.2015.04