

## Data Mining on Romanian Stock Market Using Neural Networks for Price Prediction

Magdalena Daniela NEMEȘ, Alexandru BUTOI  
Business Information Systems Department  
Babeș-Bolyai University, Cluj-Napoca, Romania  
nemesdaniela@gmail.com, alex.butoi@econ.ubbcluj.ro

*Predicting future prices by using time series forecasting models has become a relevant trading strategy for most stock market players. Intuition and speculation are no longer reliable as many new trading strategies based on artificial intelligence emerge. Data mining represents a good source of information, as it ensures data processing in a convenient manner. Neural networks are considered useful prediction models when designing forecasting strategies. In this paper we present a series of neural networks designed for stock exchange rates forecasting applied on three Romanian stocks traded on the Bucharest Stock Exchange (BSE). A multistep ahead strategy was used in order to predict short-time price fluctuations. Later, the findings of our study can be integrated with an intelligent multi-agent system model which uses data mining and data stream processing techniques for helping users in the decision making process of buying or selling stocks.*

**Keywords:** Data Mining, Neural Networks, MLP, Multi-agent System, Majority Voting, Time series Forecasting

### 1 Introduction

Price estimation using time series prediction models has been largely studied as being a key topic in finance for helping users in the decision making process. By developing a strong and reliable time series forecasting model, traders can adjust their trading decisions by following specific buy-sell strategies. Even though many traders rely their decision on speculation, experience and intuition, specific strategies based on data analysis are still more reliable for inexperienced users. In order to prevent random trading, a general trading method should be developed to help users in defining their position. Buying and selling stocks can be extremely difficult when changes occur unexpectedly. A general forecasting method should be included as a measure of guidance in order to prevent massive losses or poor decisions.

A large interest regarding forecasts on stock exchange market and currency exchange rates has been observed in the current literature [1-3].

Neural networks are considered a popular and useful strategy as it offers a variety of different models which can be used and adapted according to each specific problem.

Trading strategies and patterns can be observed using neural networks and can be exploited in order to obtain competitive advantages.

This paper presents extends [4], which is a short introduction to financial markets and neural network modeling. Here, we present a case study where price prediction methods are evaluated in order to find whether using neural networks can be considered an acceptable trading strategy among other trading methods.

The main purpose of this paper is to evaluate the neural network time series forecasting models by building and creating a series of neural networks and by trying to predict the future price value for three main Romanian stocks in order to find suitable learning models in the forecasting process. After computing hundreds of tests and consulting the current literature a prediction strategy was discovered. A multistep ahead strategy was evaluated and results were computed using Weka and Matlab.

### 2 Related Work

This paper focuses on evaluating time series forecasting strategies using neural networks

as main learning method. In order to evaluate prediction results a general understanding on how neural networks work is needed.

In [5], artificial neural networks are considered "universal approximators" capable of learning and adapting their structure to each particular input data. Designing an optimal network usually implies constant work based on a trial and error process.

The main idea behind neural networks is related to the way information is processed by the human brain. Having its origins in the biological nervous system, a neural network is organized in large interconnection streams between neurons that are designed to work together in solving a specific task.

Several articles on neural networks and their financial application are published every year. Well known works such as [6], [7], [8] and [9] are used as a point of reference in the literature by offering specific guidelines in the neural network development process. New models of Neural Networks are constantly presented but the most common artificial neural network is the multilayer perceptron neural network (MLP NN). The MLP has three main layers: the first layer is the input layer which contains the main information to be processed, the second layer is called the "hidden" layer where neurons create weighted connections and use the activation function in order to obtain the "output" layer which strongly depends on the activity in the hidden layer. The main output of the MLP is compared to the corresponding target output and by completing several model evaluations the error rate is reduced [10].

A single-layer perceptron (SLP) is a perceptron having only one layer of variable weights and one layer of output neurons [12]. The main algorithm behind perceptrons ensures that the learning process takes place in a finite amount of time. The perceptron model designed by Rosenblatt in 1962 has the

following form:  $f(a) = \begin{cases} -1 & \text{if } a < 0 \\ +1 & \text{if } a \geq 0 \end{cases}$  [12].

SLP are designed to divide inputs in a two-dimensional input space by means of a straight line. Because SLP can solve only simple problems, a new multi-layer percep-

tron was created in order to solve complex issues by using two or more trainable weight layers. This is how the need to develop multi-layer perceptrons emerged.

Perceptrons with more than one layer of variably weighted connections are referred to as multilayer perceptrons (MLP). A n-layer perceptron has n variable weight layers and n + 1 neuron layers with neuron layer 1 being the input layer [11]. The multi-layer perceptron neural network (MLP) has a feed forward based architecture, and is trained using the back propagation training algorithm [13]. MLPs are generally created of three main layers: an input layer, a hidden layer and an output layer. All these three levels are interconnected using a specified methodology.

The learning method used by MLPs is considered to be a supervised method because the output values are already known to the MLP. The MLP method has two characteristics [14]: (1) it has nonlinear processing elements (PEs) with a nonlinearity that must be smooth; (2) massive interconnectivity determines strong relationships between layers (any element of a given layer feeds all the elements of the next layer).

When working with time-series data, using recurrent networks can be useful as their design is compact and can reduce the number of required input nodes. An example of a classic recurrent network is the Elman network where nodes from the hidden layer have feedback connections to themselves. The classic training method for these networks was proposed by Elman in 1990 and consists of using the back-propagation algorithm combined with treating each of the feedback inputs from the context layer as an additional input to the network at the next time step [15]. Currently there are several variations for this specific network used in time series prediction presented in [16-19]. However using recurrent models can become time consuming if the number of layers increases or the used dataset is large.

In 2011, Guresen, et. al. [14] evaluates the effectiveness of different neural network models by comparing the Mean Square Error (MSE) and Mean Absolute Deviate of

NASDAQ Stock Exchange index obtained on different ANN such as multi-layer perceptron (MLP), dynamic artificial neural network (DAN2) and the hybrid neural networks (GARCH). The study concluded that MLP was the best used architecture when forecasting stock movements. Even though many hybrid models have been developed there are still traditional ANN models that offer accurate predictions.

Our previous study on stock forecasting presented in [4] introduces several models, mainly implemented in the Weka Time Series Forecasting Module:

- *Multilayer Perceptron Regressor* (MLPReg) which is an extension of the MLP that uses an optimization class in order to minimize the squared error plus a quadratic penalty;
- *RBF Regressor* is a neural network which implements radial basis function networks in classification problems and differs from the traditional RBF network by using the BFGS method for minimizing squared error [20];
- *Linear Regression* – widely studied within statistics, represents a good alternative when dealing with numeric attributes and price forecasting to express outputs;
- *Gaussian Processes* is a general regression method structured as a collection of random variables which have (consistent) joint Gaussian distributions [21];
- *Sequential minimal optimization (SMO)* is an improved algorithm for addressing regression shortcomings using support vector machines [22].

In the previous paper [4] a general multi-agent system using different prediction methods was presented aiming to help users in the decision making process using a Majority Voting System (MVS). The system tackles with the problem of strong dynamics of trading data. An intelligent, autonomous multi-agent system can provide powerful trading support, facilitating the decision making process. The autonomous system consists of several forecasting agents with different learning strategies for short-term price pre-

diction outputting multiple predicted values. A confidence index based on the majority voting technique will aggregate the output values offering real-time trading support. The prediction performance of the system is determined by comparing the predicted values with the real market values of the same period.

### 3 Measuring performance

In order to evaluate the performance of a neural network, a set of different metrics can be used and compared. The literature offers many evaluation alternatives and each one can help in making decisions. The most common error measures are the following [12]:

- mean absolute error (MAE) -  

$$MAE_j = \frac{1}{N} \sum_{i=1}^N |\bar{Y}_i - Y_i| ;$$
- mean squared error (MSE) -  

$$MSE = \frac{1}{N} \sum_{i=1}^N (\bar{Y}_i - Y_i)^2, \text{ where } \bar{Y}_i \text{ is the}$$

predicted value and  $Y_i$  is the observed value for a number of N observations;

- root mean squared error (RMSE) -  

$$RMSE = \sum \sqrt{\frac{1}{N} \sum_{i=1}^N (\bar{Y}_i - Y_i)^2}, \text{ where } \bar{Y}_i \text{ is}$$

the predicted value and  $Y_i$  is the observed value for a number of N observations.

A very common criteria is using the MAE or the MSE as a means of establishing the best performance. Even though these evaluation metrics are largely used, there are a series of drawbacks regarding to their use. Both MAE and MSE can lead to poor generalization due to the used learning datasets. By including evaluation metrics of risk-adjusted profit, the obtained results can be considered more reliable.

When evaluating the MLPs performance, the used dataset is extremely important. Moreover, the training dataset must be representative for the whole dataset and should be evaluated several times to produce different training and validation datasets.

There is no general rule about the number of hidden layers a MLP neural network should have. The number and the size of these layers is usually determined heuristically and should be adapted according to each situation. By having a large hidden layer, the amount of data needed to evaluate the model increases.

#### 4 Methodology

This paper focuses on time series forecasting strategies using neural networks as the main learning method.

In order to evaluate the prediction performance, three different stocks were analyzed during the period of January 2012 to February 2013 using different input indices extracted from the technical and fundamental analysis. The scope of this evaluation was to train the neural network in order to facilitate output evaluation for the next month (March 2013). The selected stocks were BRD, TLV and SIF1 as being representative for the Romanian stock market. To build the model, the opening and closing price of the stock were considered in order to train the network. Results were computed using Weka Time Series Analysis and Forecasting module [20] and Matlab.

In the Weka Time Series Analysis and Forecasting module a Multilayer Perceptron network was computed and used as prediction model. In Matlab a NARX neural network was used in order to complete the multiple steps ahead predictions [23]. Inputs and targets were normalized into the range  $[-1,1]$ . In the multistep ahead prediction the predicted value serves as input data for the new predicted value. In this way the actual obtained error is transmitted from one step to the other determining a less accurate prediction.

#### 5 Experimental results

##### BRD Symbol Predictions

Computed results for the multistep ahead prediction using Weka Time Series Analysis and

**Table 1.** Prediction results for BRD using Weka Forecasting tool and Matlab

Measure of Error	Weka	Matlab
Mean absolute error (MAE)	0.067868	0.042539
Root mean squared error (RMSE)	0.080129	0.065131
Mean squared error (MSE)	0.006420	0.004242

Forecasting module and Matlab can be observed in Table 1 and we can observe that Matlab results were much better than the ones obtained by Weka.

##### *Weka*

The neural network built in Weka contained information regarding the opening and the closing trading values. In order to ensure better results the data was normalized in the range of  $[-1,1]$ . A general MLP algorithm was used as learning method. The network was built using a learning rate of 0.3 and a training time of 500. After testing the network the forecasting tool was used in order to predict future values for March 2013. The general MSE suggests that each predicted value was a specific error which is transmitted from one step to the other as future predictions depend on the computed value of the previous prediction. A one step ahead methodology tested on the same dataset suggests better results than the multistep ahead prediction. Even though the MSE value is acceptable we observed that the predicted values do not follow the actual trading course of the BRD stock.

##### *Matlab*

A neural network was computed with 20 neurons on the hidden layer and a delay value of 2. The training algorithm used to compute the network, was the Levenberg–Marquardt algorithm (LMA) which is widely known and used for minimizing nonlinear functions. In our case the minimization is made on the MSE. The literature considers that a proportion of 75% train, 15% validation and 15% test is a good practice. We used these values

in our evaluation but users can adapt them according to each specific problem. The error histogram computed after the training process of the neural network presents the distribution of errors during the training

of the network with three main datasets: the training, the validation and the test set. Most errors are concentrated around zero which is indicated by the orange line as the Fig. 1 presents.

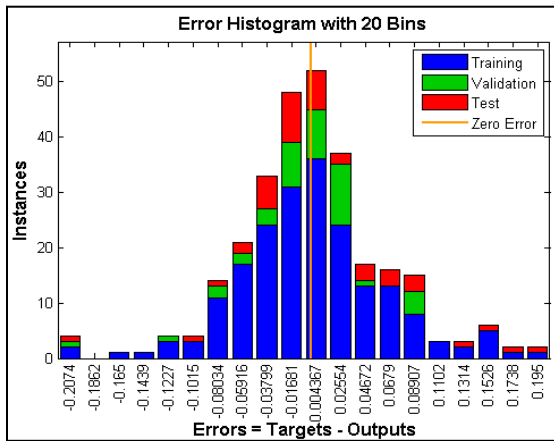


Fig. 1. The error histogram for BRD

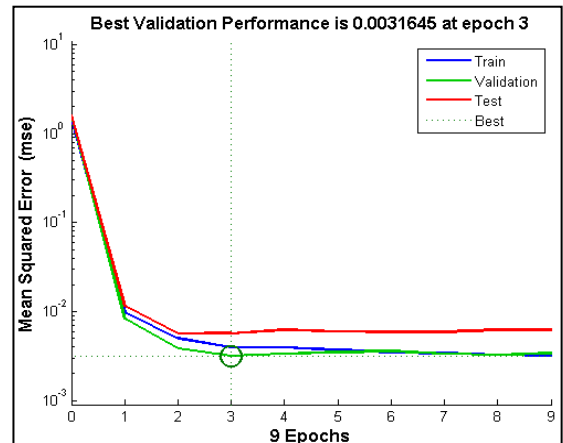


Fig. 2. MSE evolution for BRD

The MSE of the created neural network is 0.0031654 and was obtained at the 3rd epoch. The general MSE drop can be observed in Fig. 2 as the learning process determines a reduction in the network error. The training process stops when the validation error stops decreasing. The blue line in-

dicates the MSE for the training dataset, the green line for the Validation set and the red line indicates the evolution of the MSE for the test set. All three suggest a good performance of the neural network because by training the network, a constant drop of the MSE has been established.

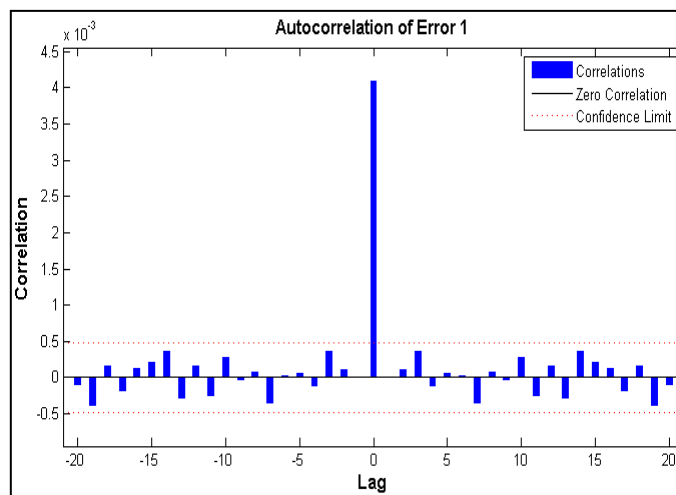


Fig. 3. The error autocorrelation for BRD

The error autocorrelation figure presents the way prediction errors are related in time. The value presented at zero is the MSE. The correlations except the zero ones tend to fall in the 95% confidence interval which suggests that the neural network was a good performance. The predicted values for BRD using

this method suggests that first computed predictions are close to the actual value as can be seen in Fig. 4. Predicted values for BRD As predicted values are further away from the first predicted value, the predicted values start to become more inaccurate and the MAE value increases.

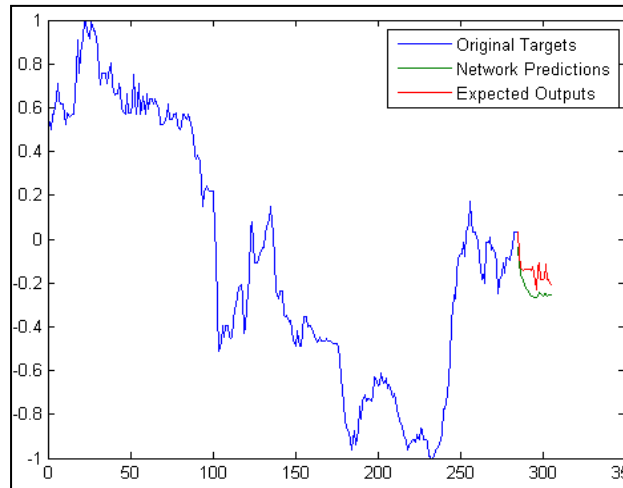


Fig. 4. Predicted values for BRD

**SIF1 Symbol Predictions**

Computed results for SIF1 using Weka Time Series Analysis and Forecasting module and Matlab can be observed in the following table:

**Table 2.** Prediction results for SIF1 using Weka Forecasting tool and Matlab

Measure of Error	Weka	Matlab
Mean absolute error (MAE)	0.095602	0.077452
Root mean squared error (RMSE)	0.107448	0.102246
Mean squared error (MSE)	0.011545	0.010454

**Weka**

The same methodology was applied for SIF1. The learning algorithm was MLP and the prediction was made on 21 days representing the number of trading days of March 2013. The neural network used was the MLP neural network at a learning rate of 0.3 and a training time of 500. Used data was normalized in the range of [-1,1]. Computed results are less accurate than the obtained results using NARX neural network.

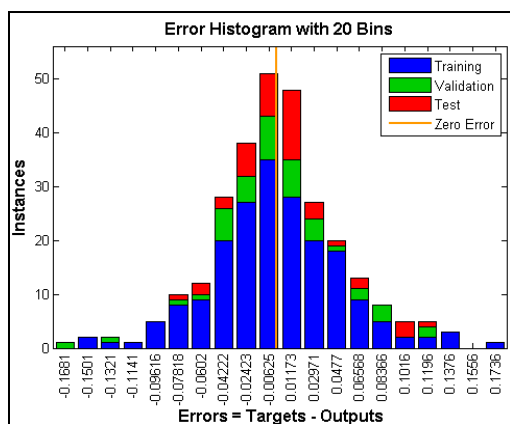


Fig 5. The error histogram for SIF1

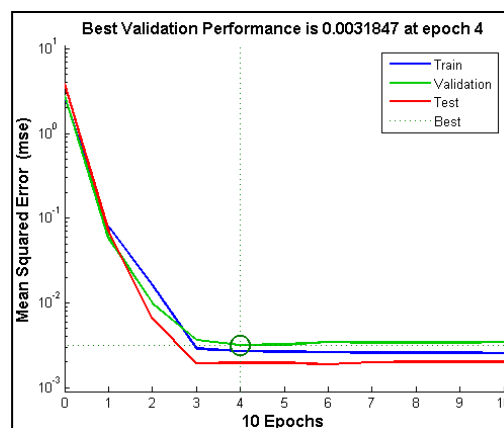


Fig. 6. MSE evolution for SIF1

**Matlab**

Computed results show that the distribution of errors is concentrated near 0. The obtained MSE of the network is 0.0031847 and was obtained at the 4th epoch. Even though many

tests were performed no major improvement was detected in the obtained values. The final created network has 20 neurons on the hidden layer and uses the Levenberg–Marquardt algorithm as learning method.

***TLV Symbol Predictions***

Computed results for TLV using Weka Forecasting tool and Matlab can be observed in Table 3. Matlab results are not as good as the ones obtained with the Weka Forecasting tool.

**Table 3.** Prediction results for TLV using Weka Forecasting tool and Matlab

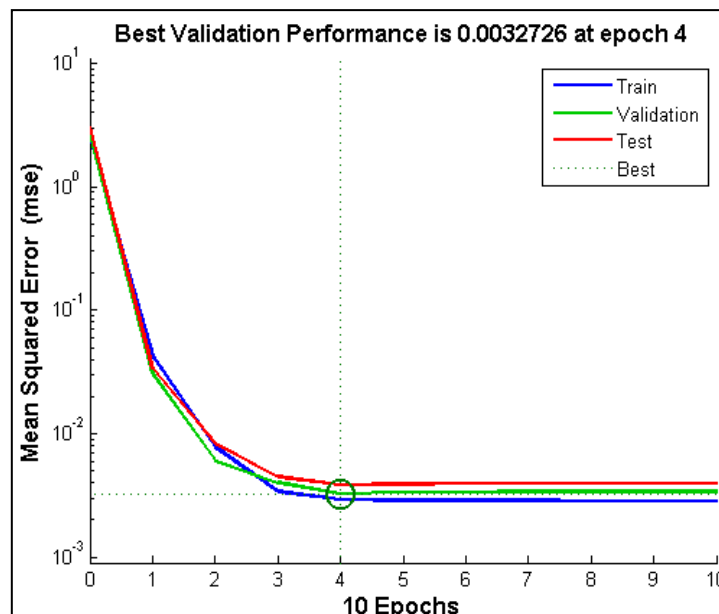
Measure of Error	Value Weka	Value Matlab
Mean absolute error (MAE)	0.040360	0.046114
Root mean squared error (RMSE)	0.048316	0.056152
Mean squared error (MSE)	0.002334	0.003153

***Weka***

A MLP neural network was used at a learning rate of 0.3, a momentum value of 0.2 and a training time of 500. Input data was normalized in the range of [-1,1]. After testing and training the network, computed results for the 21-day prediction show a MSE of 0.002334 which was obtained by comparing the actual values with the predicted values of the same period of time.

***Matlab***

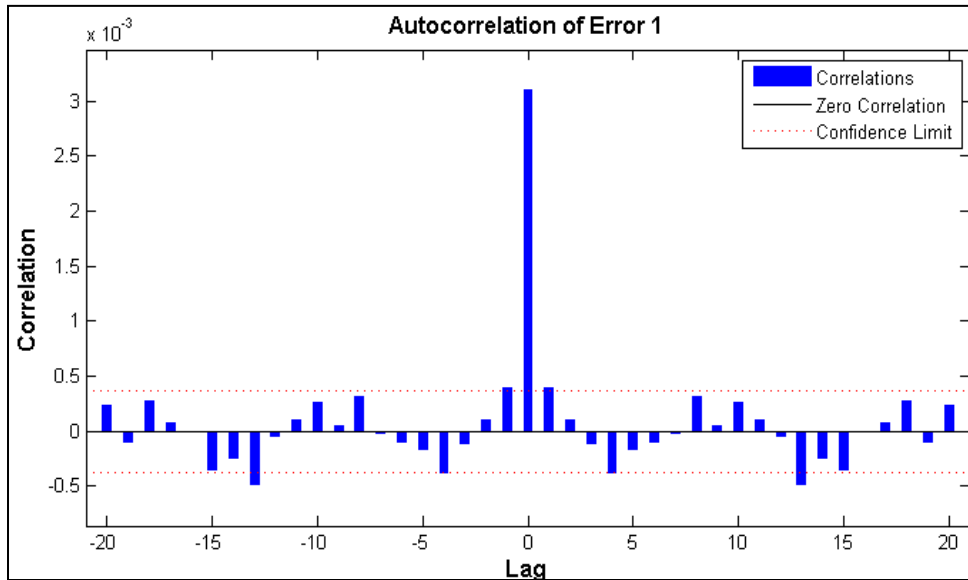
The same type of neural network was computed for TLV as previously presented for BRD and SIF1. We can see that by starting the learning process the MSE for the train, test and validation datasets decreased rapidly as the network adapted the outputs to the included inputs. The best validation MSE was obtained at epoch 4 with a value of 0.0032726 presented in Figure 7.



**Fig. 7.** MSE evolution for TLV

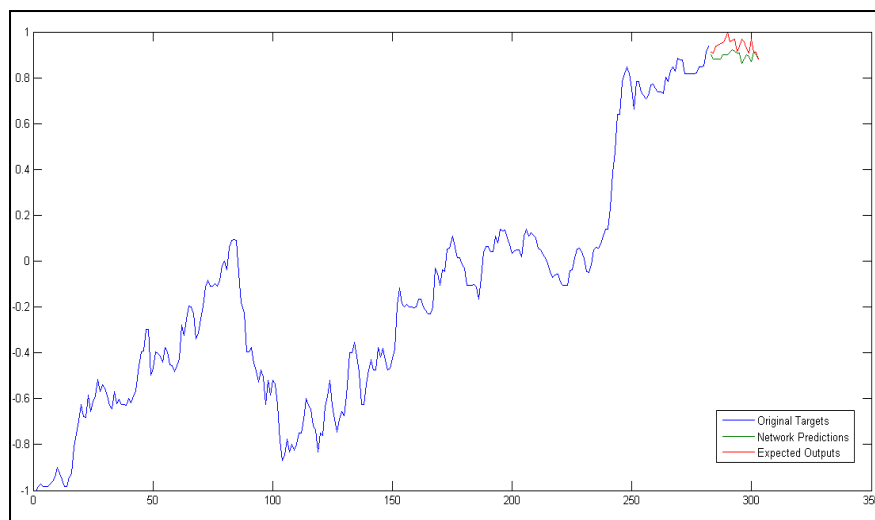
The autocorrelation of errors (Figure 8) during the training process of the neural network suggest that most of the computed values are situated in the 95% confidence interval. There are however some exceptions which

suggest that the created neural network can be improved by retraining it or by increasing the number of neurons in the hidden layer. Several tests were computed but no major improvement was obtained.



**Fig. 8.** The error autocorrelation for TLV

The actual and the predicted values can be observed in the chart below:



**Fig. 9.** Predicted values for TLV

## 6 Discussion

The overall obtained results suggest that Matlab results were more accurate than Weka obtained results. Multiple steps ahead predictions usually offer less accurate results than one step ahead predictions. However it is difficult to consider that investors can solely rely on the predicted values. It is important to see the general help that these tools offer users in order to predict future values but other important factors have to be taken into consideration when making decisions. A general model cannot be created from the ob-

tained results as each stock has its own particularities and can be influenced by multiple factors which are not always included in the prediction model. In order to tackle with this, the previous paper [4] presented an adaptive multi-agent system which deals with the following problem that every prediction method needs a specific setup for achieving the best performance. We proposed a multi-agent model for dynamic computation of the best time length of the required training dataset for every new symbol built on 5 types of agents which coordinates and cooperates



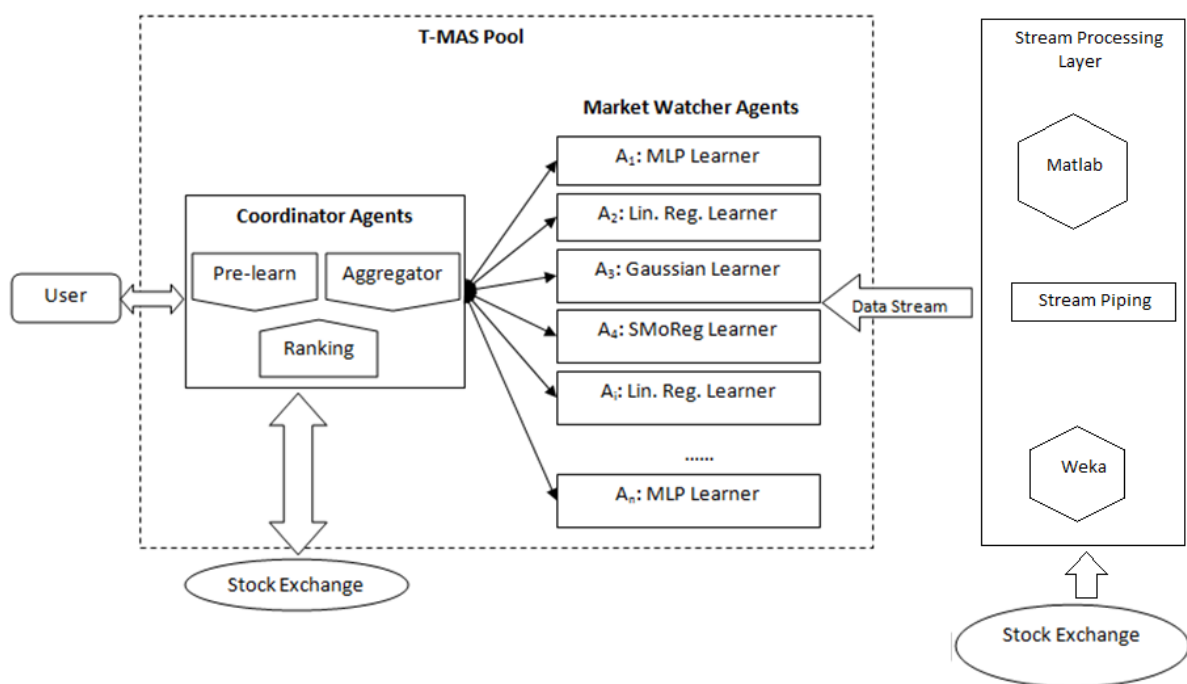
with each other in order to obtain the best performance in the prediction process:

- (1) The market watcher agent implements one of the six studied learning and prediction strategies with its parameters tuned for best performance on each traded symbol.
- (2) Coordinator Agents will organize, coordinate and supervise the market watchers in order to obtain maximum global performance and accuracy of the predictions.
- (3) Pre-learn coordinator agents will enable the pre-learn process of the agents for high performance learning and fine tuning of each agent for maximum performance. They will enable the update process of the learning parameters and strategy when agent predictions have increasing errors due to data change variations.

(4) Ranking coordinator agents will track the performance of every market watcher and, based on its performance history.

(5) The aggregator agents use a weighted majority voting model in order to reconcile the predicted values provided by all agents. The ranking indexes will constitute the voting power of each agent in the predicted values reconciliation. The global prediction for a specific time in the future will be calculated as weighted mean of all predicted values.

Integration of the evaluated learning processes into the agent-based framework will be achieved using an extra piping data stream processing layer. The data stream layer will have as the main data source trading data services providing real time stock data. Stream processing is suitable for trading data analyses due to the continuous specific of the trading process.



**Fig. 10.** Integration of Data Mining Tools with T-MAS system

A data stream is represented by a constant sequence of tuples generated and processed in real time [24]. Recently real-time processing of large data streams has become a major issue in performance evaluation processes. General solutions in large data stream processing have become essential when developing real time trading systems.

Because stream processing needs to be highly responsive, data stream environments has different requirements from the traditional software. The general requirements that need to be taken into consideration when working with data streams are the following [25]:

- each data should be processed individually and used only once;

- the application should use a limited amount of memory;
- results need to be presented in a limited amount of time;
- be ready to predict at any time.

In our model, integration of different datamining tools like Matlab and Weka used for price predictions are integrated with T-MAS system through a stream processing layer using data pipes. The pre-learn stage and the trading data will be computed using the third-party tools in order to avoid implementing them from scratch inside the agents. The main data stream that integrates trading data sources and T-MAS system are:

- Stock Exchange data services and the stream processing layer – where trading data is continuously collected and processed using implemented neural networks provided by the integrated tools;
- Stream processing layer with market watcher agents which use the stream processing layer as data source for providing predictions. The results and data output provided by the neural networks in the stream processing layer are used as input by market watchers for providing a prediction regarding the future prices;
- Stream processing layer with the pre-learn agents which will recalibrate the learning models implemented and integrated in the stream processing layer for maximum performance.

## 7 Conclusions and Future Development

General results show that using time series neural networks for forecasting on Romanian Stock Exchange is feasible but requires specific implementation methodology for best results [4]. Even though computed results for the multiple step ahead prediction was better when using NARX neural networks, the MLP neural network results can also be considered relevant and should be taken into consideration when developing trading strategies. It is relevant to understand the limitations imposed by these models and to define new decision criteria when buying or selling stocks. Relevant factors such as rapid chang-

es in the global economy, natural disasters or global political issues can determine significant changes in the stock market price which makes the prediction process even more difficult.

As a general observation, it is important to observe the minimum particular time span for selecting historical training data sets in order to ensure better results. Future development will focus on determining this optimum timespan for each analyzed stock in order to improve final results [4]. Because a pre-learn stage is needed to find this optimum time span for each learning process, the actual model evaluation is time consuming and requires a well organized set of experiments.

Because of the markets dynamics a need to determine the optimum training value for each dataset and to reevaluate the neural network in a continuous way emerges. In order to solve this issue stream processing techniques will be studied and analyzed to simulate real time trading systems. Time series neural networks use historical data in order to predict future values. This can be successfully applied in a trading system by using real time prediction models based on the continuous evaluation of data streams. This could help users in delimiting real time bids by observing the predicted values of the traded stock.

In order to ensure better results a multi-agent system with several prediction methodologies can be designed to ensure a more accurate evaluation of the predicted stock price. In one of my previous works I have presented a multi-agent system that could be successfully adapted for this current situation in order to facilitate the decision making process. The created system needs to be capable of self-organization and self-coordination and the main decision strategy can focus on weighted majority voting methods [4]. The best prediction method will help end users in determining the actual buy-sell strategy. The system can be integrated with existing prediction tools using a stream processing layer as a bind.

The actual limitations of the model are still yet to be found and solved. Many successful trading systems are limited by environmental and unexpected variables but continue to be successfully used by traders on different stock exchange markets.

### Acknowledgement

This paper was supported by CNCS Young Researchers Teams Project - TE 316.

### References

- [1] T. Koskela, M. Lehtokangas, J. Saarinen and K. Kaski, "Time series prediction with multilayer perceptron, FIR and Elman neural networks" In Proceedings of the World Congress on Neural Networks, September 1996, pp. 491-496.
- [2] M. Ghiassi and S. Nangoy, "A dynamic artificial neural network model for forecasting nonlinear processes", *Computers and Industrial Engineering*, Vol. 57, August 2009, pp. 287-297.
- [3] M. Khashei and M. Bijari, "An artificial neural network (p,d,q) model for time series forecasting", *Expert Systems with Applications*, Vol. 37, January 2010, pp. 479-489.
- [4] D. Nemes, "Agent-Based Data Mining Stock Evolution Forecasting". *International Conference on Informatics in Economy*, Bucharest: ASE Publishing House 2013, pp. 657-661
- [5] S. Kiranyaz, e. "Evolutionary artificial neural networks by multi-dimensional particle swarm optimization". *Neural Networks*, 2009, pp. 1448-1462.
- [6] L. S. Yu "A neural-network-based nonlinear metamodeling approach to financial time series forecasting". *Applied Soft Computing* 9.2, 2009, pp. 563-574.
- [7] Z. Liao a. "Forecasting model of global stock index by stochastic time effective neural network". *Expert Systems with Applications* 37.1, 2010, pp. 834-841.
- [8] Y. Zhang a. Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network. *Expert systems with applications* 36.5, 2009, pp.8849-8854.
- [9] P.-C. Chang e. "A neural network with a case based dynamic window for stock trading prediction". *Expert Systems with Applications* 36.3, 2009, 6889-6898.
- [10] C.-F. Tsai, a.-W. "Using neural network ensembles for bankruptcy prediction and credit scoring". *Expert Systems with Applications* 34.4, pp. 2639-2649.
- [11] D. Kriesel, "A Brief Introduction to Neural Networks", 2007, [Online] Available at [http://www.dkriesel.com/\\_media/science/neuronalenetze-en-zeta2-1col-dkrieselcom.pdf](http://www.dkriesel.com/_media/science/neuronalenetze-en-zeta2-1col-dkrieselcom.pdf)
- [12] M. B. Ersin, "Improving forecasts of GARCH family models with the artificial neural networks: An application to the daily returns in Istanbul Stock Exchange". *Expert Systems with Applications*, Volume 36, Issue 4, May 2009, pp. 7355-7362.
- [13] A. Brabazon a. "Biologically inspired algorithms for financial modelling", Vol. 20. Berlin: Springer, 2006, pp.18-22, 130-133, 144-148, 153, 162-169.
- [14] G. K. Erkam. "Using artificial neural network models in stock market index prediction". *Expert Systems with Applications*, Volume 38, Issue 8, August 2011, pp. 10389-10397.
- [15] J. Elman. "Finding structures in time". *Cognitive Science* 14, 1990, pp. 179-211.
- [16] M. A.-F. Zolfaghari. "Chaotic time series prediction with residual analysis method using hybrid Elman–NARX neural networks". *Neurocomputing*, Volume 73, Issues 13–15, 2010, pp. 2540-2553.
- [17] M. K. Bijari. "An artificial neural network (p,d,q) model for timeseries forecasting", *Expert Systems with Applications*, Volume 37, Issue 1, January 2010, pp. 479-489
- [18] R. C. Zhang. "Cooperative convolution of Elman recurrent neural networks for chaotic time series prediction. *Neurocomputing*", Volume 86, June 2012, pp. 116-123.
- [19] H. Tan. "The Application of Improved Elman Neural Network in the Exchange Rate Time Series". *Artificial Intelligence and Computational Intelligence (AICI)*,

- 2010 International Conference on, vol.3, October 2010, pp. 440-443.
- [20] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. Witten, "The WEKA Data Mining Software: An Update", SIGKDD Explorations, Vol. 11, 2009.
- [21] C. E. Rasmussen, "Gaussian Processes in Machine Learning", MIT Press, 2003
- [22] S. S. Keerthi, Bhattacharyya C. and Murthy K.R.K., "Improvements to the SMO algorithm for SVM regression", IEEE Transactions 11.5, 2000, pp. 1188-1193.
- [23] MathWorks Documentation Center. (n.d.). Time Series Prediction. Retrieved June 23, 2013 [Online], Available at: <http://www.mathworks.com/help/nnet/gs/time-series-prediction.html>
- [24] M. Stonebraker, Ç. Uğur, and Z. Stan "The 8 requirements of real-time stream processing". ACM SIGMOD Record 34.4, 2005, pp. 42-47
- [25] Bifet A., R. Kirkby, P. Kranen, P. Reutemann, "Massive Online Analysis, Manual", [Online], May 2011, Available at <http://www.cs.waikato.ac.nz/~abifet/MOA/Manual.pdf>



**Daniela NEMEȘ** has graduated from the Faculty of Economics and Business Administration in 2013. She holds a bachelor diploma in Finance and Economics as well as in Business Information Systems and she currently is a second year Master student in the same department.



**Alexandru BUTOI** is a PhD student since October 2012, in the domain of Cybernetics and Statistics at Faculty of Economics and Business Administration, "Babeș-Bolyai" University of Cluj-Napoca. His main research interests are Web technologies and Mobile technology applied in business and lately, Cloud Computing, focusing on data security and privacy in Cloud Computing. He also worked on development of agent-based cloud resource negotiation protocols and developed a secure protocol for cloud storage environments.

vironments.