INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 15, Issue: 4, Month: August, Year: 2020
Article Number: 3910, https://doi.org/10.15837/ijccc.2020.4.3910

Analytical Modelling of Share Price Value Using Computational Intelligence Methods

M. Petrescu, M. Cuc, I. Oncioiu, A.-G. Petrescu, F.-R. Bîlcan, L. Ivan

Marius Petrescu*
Valahia University of Targoviste
130024 Targoviste, 2 Carol I Bvd, Romania
*Corresponding author: marius.petrescu@valahia.ro

Mădălina Cuc
Academia Nationala de Informatii „Mihai Viteazul”
075100 București, Odăii 20-22, Romania
madalina.cuc@animv.ro

Ionica Oncioiu
Titu Maiorescu University
040051Bucharest, 189 Calea Văcăreşti Street, Romania
ionica.oncioiu@prof.utm.ro

Anca-Gabriela Petrescu, Florentina-Raluca Bîlcan, Lucian Ivan
Valahia University of Targoviste
130024 Targoviste, 2 Carol I Bvd, Romania
anca.petrescu@valahia.ro raluca.bilcan@valahia.ro
lucian.ivan@valahia.ro

Abstract

The liberalization, volatility and high competition of financial markets are all factors that expose companies to new risks and challenges, requiring a continuous innovation of models, techniques and tools for managing share price value and other related risks. This paper provides a model to implement a subsystem that should support the management decision on the trading segment of its own shares, based on intelligent agents with regards to identifying, collecting, structuring and updating data instruments, with analysis mechanisms of the main price and volatility indicators. Our conclusion is that the use of computational intelligence methods in modeling of share price value is both a requirement and an option in managing financial-foreign exchange risks, given that companies are subject to a wide range of risks and volatility is the defining feature in which they operate.

Keywords: neural networks, intelligent systems, share price, financial daily vector, exchange risks.

1 Introduction

In the actual context, of exponential growth of data amount and computing power, of cloud computing as a result of the need for infrastructure spending decrease, at the same time with defining platform services (PaaS), software services (SaaS) and infrastructure services (IaaS), the possibility of using intelligent agents in identifying, collecting, structuring and updating data in databases is generated, the core of the transactional decision support system being represented by the specific transactional risk assessment instruments that will run on these databases [1, 7, 11].

From a practical point of view, the analysis of the efficiency of intelligent agents can be performed using neural models. Being nonlinear models, their capacity for representation is significantly more pronounced than that of linear equivalents (where such a thing exists), but a nonlinear model is not always needed in prediction.

In Romania the impact of the social factor, the absence of management efficiency measuring metrics, including for the economic, technological and last but not least ecological factors, did not enable the development of certain models which would foster the creation and evolution of some substantiation tools and decision support specific to developed economies. Currently, the roles of managerial decision substantiation and support systems consisted, at most, in their great majority, in statistical data collection, the measurement of classic indicators, alert and a set of damage control measures in case of exceeding critical performance indicators (not trading indicators). There is an almost nonexistent methodology to create indicators which would be the result of applying metrics on spatial elements given by the multitude of simulations of convergent, stable and controllable models, and this happens due to the lack of argument elements, meaning model simulations, caused by the lack of specifically built models [3, 9].

Starting from the premise of the necessity of building a decision support system specific to each national company and not a general one, customized for each company, as a result of the evolutions which would occur following the listing of these closed owned companies, the aim of this paper is to implement a subsystem to support the management decision on the trading segment of its own shares, based on intelligent agents in identifying, collecting, structuring and updating data instruments, with analysis mechanisms of the main price and volatility indicators [3, 9].

Our paper has two major goals. First, we focus on assessment the model of price prediction for share based on neural networks and the model of volatility prediction, encapsulated in an agent-based system make up a subsystem of a decision support system. Second, this research helps creating a subsystem of managerial decision assistance on the trading segment of own shares, based on modelling of share price value using computational intelligence methods.

The novelty elements of this research covered some existing gaps in the literature by highlighting aspects related to the training of a neural network to make more accurate predictions for the future share price value according to Romanian capital market conditions.

The structure of the paper is as follows: Section 2 describes a brief presentation of the literature. Section 3 provides the proposed model and methodology. Section 4 analyses the empirical results and discussions implications and Section 5 drives conclusions.

2 Literature review

In literature, the first techniques specific to Computational Intelligence dates back 22 years ago and emerged from the need to resolve problems that cannot have solutions by traditional methods or when the available data is not sufficient to define a logical model leading to an algorithm that would enable one unique solution or more [2, 4, 7, 12].

Thus, there are numerous question marks regarding the metric for determining the degree of difficulty of the problems submitted to solution that would impose an exact approach specific to Artificial Intelligence or an "almost exact" one, specific to Computational Intelligence [13, 14]. Related to this, the notion of intelligence can be defined as the ability of machines to solve or assist man in solving "difficult problems". Such a situation proposes to offer alternatives to achieving an objective by using the available resources. In the case of natural intelligence, the natural selection based on competi-

tion offers the solution to the existence and evolution of the living organism, the adaptation to the changes that have occurred in the evolutionary environment being its most important characteristic. Furthermore, beyond the living organisms' struggle for survival, intelligence is not limited to biological organisms and their struggle for survival but passes to a new level, that of the possibility of choice, seen as a decision taken by a decision-maker or a factor of decision and which can be influenced by various factors [15, 16].

Generally speaking, Computational Intelligence was defined as the capacity of a system to interact with its environment, being able to take decisions based on previous and current information/data, using evolutionary fuzzy computation, neural networks and combinations of the two [8, 18, 20]. Compared to the concept of (traditional) Artificial Intelligence, Computational Intelligence does not use an exact model, and the input information for the model is incomplete, with high uncertainty and low accuracy [19, 22].

The present methodologies for the use and application of Computational Intelligence are incorporated into an elastic framework, under which new methods are added to the existing ones, with its permanent extension. It is worth noting the common aspect of this feature - elasticity - with Cloud Computing, the natural stage of development of these methodologies moving toward the technologies distributed in the Cloud.

On the other hand, other authors document the Computation Intelligence concept by describing the most applied components of this field: neural networks, a component which will be used in this study to create intelligent agents that will be implemented in the agent system, built in an analysis model of the trading risk management of shares [1, 5, 6, 17].

The study implemented in [21] used Artificial Neural Networks (ANNs) to monitor the information collection/processing systems based on simple connected elements, similar to the biological neurons, capable of collecting and processing information. These simple connected elements are distributed throughout all the areas of the neural network, processing data / information in parallel. As in the case of biological neural systems, the network's function is defined by the parameters of the elements' connection that constitute it. The parameters of the connections between neurons, expressed as weights, are the ones that store the information collected in the network. The intelligent component of the neural networks is given by network's learning/teaching feature and is achieved by adjusting the weights according to specific algorithms.

This learning is done as in the case of neurological systems through lessons/examples. Following the learning process, ANN can be configured for pattern recognition or classification. In the case of biological systems, learning/ training modifies the tensions of the synaptic connections between neurons. In artificial neural networks, learning is achieved by adjusting the weights.

The representation of a network's neurons can be regarded as a parallel processing assembly and distributed by interconnected processing modules (nodes). The input data in each node are values in the closed range [0, 1].

3 The proposed model and methodology

The proposed model and methodology are based on the works of McCulloch and Pitts [15], who first defined the computing machines as neural networks. According to these authors the Artificial Neurons (AN) can be defined as a nonlinear function from

$$f : R^n \to [0, 1] \text{ or } f : R^n \to [-1, 1]$$

in relation to the "activation function" used, where n is the number of the input values which are also called input signals [13].

At the same time, this model assumes the existence of a value threshold called excitation threshold, a situation in which the net input signal is added to the threshold value,

$$s = wx + b.$$

The value of input x is multiplied by the corresponding weight w and the result is the value of

input signal s = wx which is the same real value as x. For positive weights

$$w > 0$$

we will call them excitatory synaptic weights, and for negative

$$w < 0$$

we will call them inhibitory synaptic weights.

The value of the net input signal is transformed by the activation function and results in the answer given by the artificial neuron. Usual activation functions are: step function (hardlim), signum function (hardlims), sigmoid transfer function (logsig), linear transfer function (purelin), hyperbolic tangent sigmoid transfer function (tansig) and the enumeration is not exhaustive.

The n-dimensional artificial neuron with correction or activation threshold can be described by a function

$$h_\beta : R^n \times R^n \to [-1, 1] \; or \; h_\beta : R^n \times R^n \to [0, 1]$$

where n is the size of the input system

$$R^n \ni x = (x_1, x_2, \; ... \, x_n)^t$$

as well as the size of the system weights

$$R^n \ni w = (w_1, w_2, \; ... \, w_n)^t$$

and scalar b will be called a correction or activation threshold and represents a factor that influences the neuron activation, with the role of increasing or decreasing the value of the input signal to the activation function.

In the model in this study, another equivalent description of the above neuron can be made by adding a new *+1* input synapse and weight equal to the activation threshold, meaning:

$$x_{n+1} = 1$$

and

$$w_{n+1} = b$$

Briefly, the net input signal and the output of the neuron representation being exemplified in Figure 1. The proposed model and methodology are based on the works of McCulloch and Pitts[15], who first defined the computing machines as neural networks. According to these authors the Artificial Neurons (AN) can be defined as a nonlinear function

$$f : R^n \to [0, 1] \; or \; f : R^n \to [-1, 1]$$

in relation to the "activation function" used, where n is the number of the input values which are also called input signals[13]. At the same time, this model assumes the existence of a value threshold called excitation threshold, a situation in which the net input signal is added to the threshold value,

$$s = wx + b$$

The value of input x is multiplied by the corresponding weight w and the result is the value of input signal

$$s = wx$$

which is the same real value as x.

For positive weights $w > 0$ we will call them excitatory synaptic weights, and for negative $w < 0$ we will call them inhibitory synaptic weights. The value of the net input signal is transformed by the activation function and results in the answer given by the artificial neuron. Usual activation functions are: step function (hardlim), signum function (hardlims), sigmoid transfer function (logsig), linear

transfer function (purelin), hyperbolic tangent sigmoid transfer function (tansig) and the enumeration is not exhaustive. The n-dimensional artificial neuron with correction or activation threshold can be described by a function

$$h_\beta : R^n \times R^n \to [-1, 1] \; or \; h_\beta : R^n \times R^n \to [0, 1]$$

where n is the size of the input system

$$R^n \ni x = (x_1, x_2, \; ... \, x_n)^t$$

as well as the size of the system weights

$$R^n \ni w = (w_1, w_2, \; ... \, w_n)^t$$

and scalar b will be called a correction or activation threshold and represents a factor that influences the neuron activation, with the role of increasing or decreasing the value of the input signal to the activation function. In the model in this study, another equivalent description of the above neuron can be made by adding a new +1 input synapse and weight equal to the activation threshold, meaning:

$$x_{n+1} = 1$$

and

$$w_{n+1} = b.$$

Briefly, the net input signal and the output of the neuron representation being exemplified in (Figure 1).

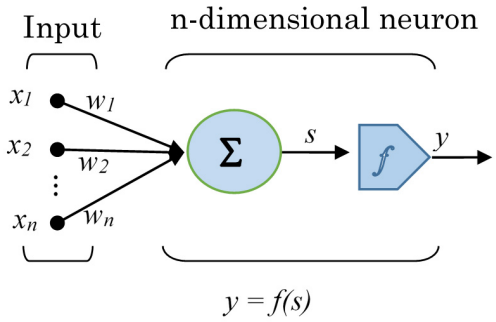

Figure 1: Form of the non-linear n-dimensional neuron model Haykin[13])

The methodology for building a neural network in this study started with the following steps:

- Analysis and quantification of the situation in terms of input and output values by specifying the requirements.
- It starts with the most parsimonious model which is also a solution to the problem.
- The values of the weights and the threshold values are determined so as to pro-duce the output results corresponding to the desired answers for all the elements in the training data set.
- The values of the weights and the threshold values are determined so as to pro-duce the output results corresponding to the desired answers for all the elements in the test data set to verify the generalization capacity.
- The stop condition relative to the test data should be checked. If this is not met, resume at step 3 for other values of weights and thresholds.
- If more than k resumes of step 3 are made and the stop condition is not verified, step 2 is resumed with the modification of the network topology (number of neurons and layers).

- If more than p resumes of step 2 are made and the stop condition is not verified, step 1 is resumed and the variables representing the problem are re-quantified.

Furthermore, the neural networks involve the transformed transmission of input-to-output (forward) signal values and the methodology for carrying out the learning process defines the learning algorithm and operationalizes this process by the weights' modification function (synapses) of the neural network until it reaches the desired objective.

However, the primary element of neural intelligence, the learning, consists in transforming the weights $(w_1, w_2, \ldots w_n)$ of the processing elements from the input to the output.

The transformation of a connection's weight or more exactly of a processing element depends on the importance of its role in minimizing the difference between the output and the desired result. The importance degree or the power of a connection will be expressed depending on the weight wij during the learning process. Within this learning process there can be learning sequences based on accurate data subsets in the inaccurate data sets and certain conditions in uncertain conditions sets, and these sequences use a mathematic model to adjust weights. Such a learning sequence using the data and conditions previously described is called a learning rule.

In the prediction of share price, the problem of training a neural network is described through the minimization of a criterion function. When its value is less than a threshold called error, we say that the training is done with a given error threshold. The rule for updating weights is under the form:

$$w_{k+1} = w_k + \frac{c}{2\,(d_k - y_k)}$$

where:

w_{k+1} is the weights' vector at step $k+1$;

w_k is the weights' vector at step k;

d_k is the desired output vector at step k ;

y_k is the obtained output vector at step k ;

x_k is the input vector at step k ;

c is the correction constant $(0 < c \leq 1)$.

Simultaneously, the network's convergence and performance will be calculated based on the testing data, this being the stopping condition for the whole training and parametrization process.

The data used in this paper for Romanian companies are taken from the Bucharest Stock Exchange data portal where they are made publicly available [23]. The neural network performing the price prediction for companies listed will be built, trained, validated and tested according to the following steps. Using the MATLAB program, we will perform the numerical and the graphical simulations that will represent all the stages. The analysis interval comprises a period between 2013 and 2016, with an aggregate of 688 recordings for the data series of the variable – daily closing price. Also, the series of values for the daily traded volume was also stored in the database. Of the 688 values 488 were retained for the network training data and 100 values for validation and testing.

The endogenous variable considered is the share price (daily closing) and the exogenous variable considered is the trading volume. The data used are for the daily closing quotas and are represented in Figure 2.

As a rule, overfitting appears when a model is more flexible than it should be or when it is excessively complex, having too many parameters in relation to the number of observations.

4 Experimental results

Following the proposed model and methodology presented above, taking into consideration the neural network multi-perceptron with the endogenous variable given by daily closings and the exogenous variable given by the volume of open-loop architecture transactions, the predictive performance measures will be R^n and MSE. The result of this analysis was given in Figure 3.

The training data (488) will generate the calculus of the gradient and weights of the network. The validation data (100) will be used for stopping the training as a protection to the "overspecialization".

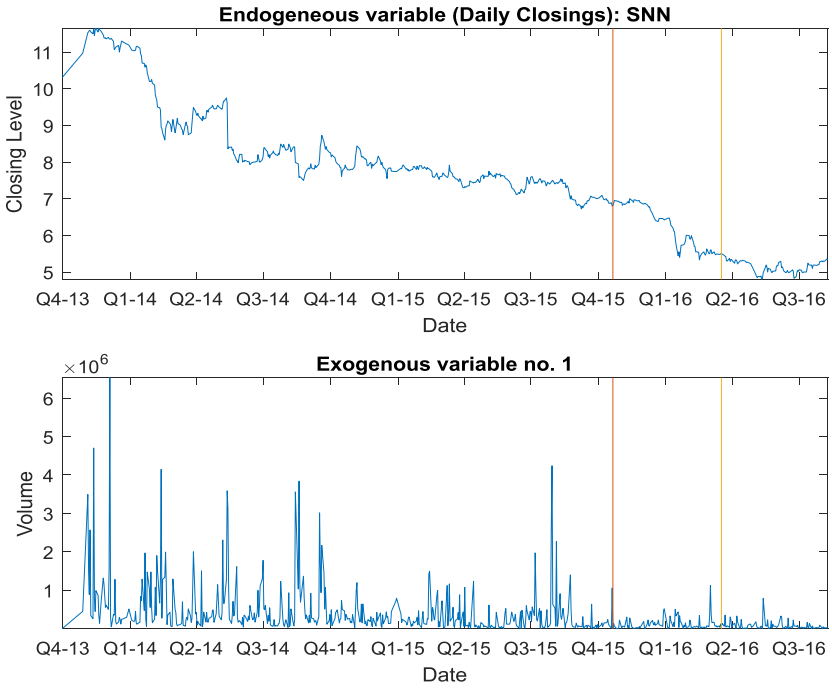

Figure 2: Representation of the variables (endogenous and exogenous))

For the neural network presented, the training data set is used to the calculus of the network gradient and weights without stopping the process.

The process stopping will be performed through the validation data after error evaluation. This error decreases during the initial training stage, just as the error obtained at the error evaluation for the validation data. However, when the network starts overfitting the data (the model overspecialization effect), the error evaluated for the validation data usually starts increasing. When the error for the validation subset increases for a specified number of iterations, the training process is stopped and the returned weights are those corresponding to the minimum validation error.

A supplementary check of the performances of model can be made using the error histogram Figure 4.

The histogram can show the aberrant values, that are values for which the fitting is much "worse" than for most of the other data.

The Figure 5 display the autocorrelation function (ACF) of the errors for the training data for the

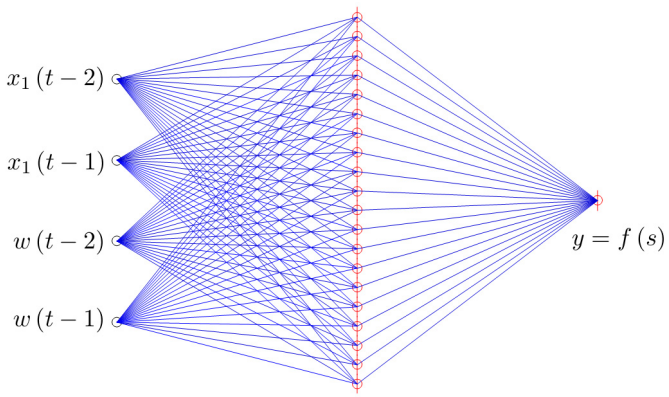

Figure 3: The neural network for model

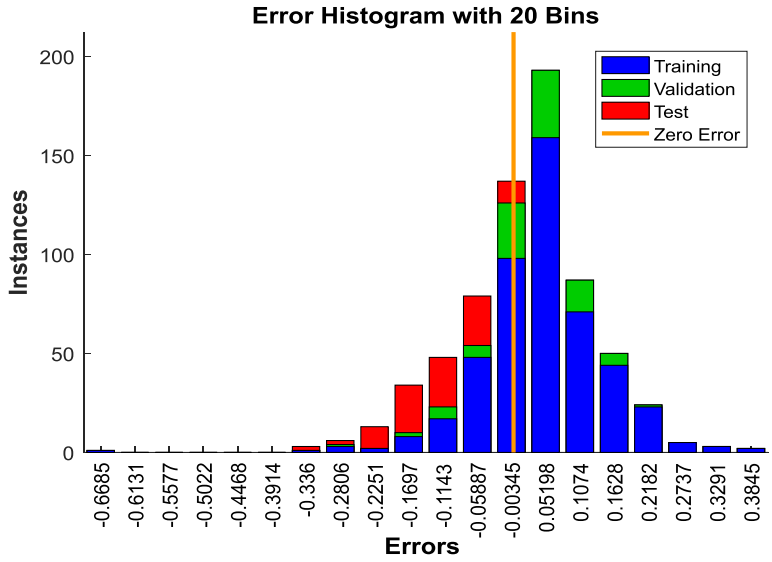

Figure 4: Error histogram for the training, validation and testing data for model

model.

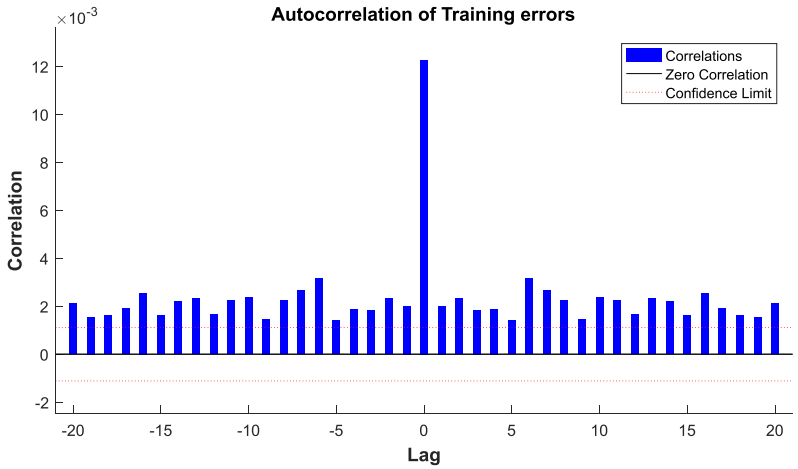

Figure 5: The autocorrelation function (ACF) of training data

The Figure 6 and Figure 7 display the autocorrelation function (ACF) of the errors for the validation and testing data for model, visualizing the way the prediction errors are autocorrelated in time. For a perfect predictive model there should be only one value differing from zero of the autocorrelation function and it should appear for a zero-time lag. This would mean that the predictive errors were totally uncorrelated with each other (the white noise).

In our case there is a slight autocorrelation for the short time lags, the rest of the correlations being approximately within the limits of the trust interval of 95 around zero, therefore the model is adequate. The result of the network predictions is shown in Figure 8. Also, from Figure 8 one may notice that between the observed values and the predicted ones there are very small differences, difficult to seize at the resolution of graphs presentation.

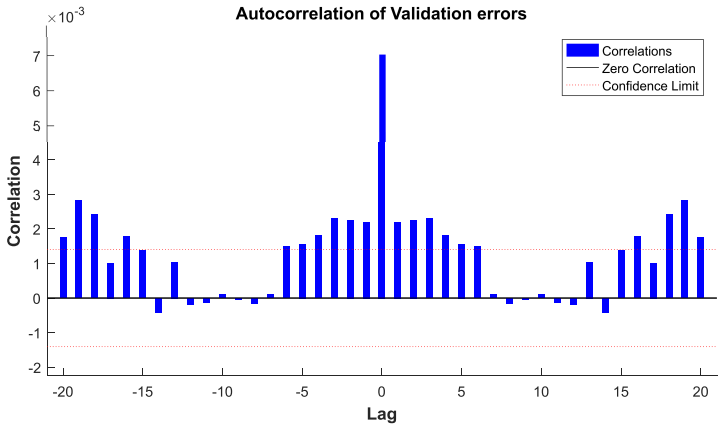

Figure 6: The autocorrelation function (ACF) of validation data

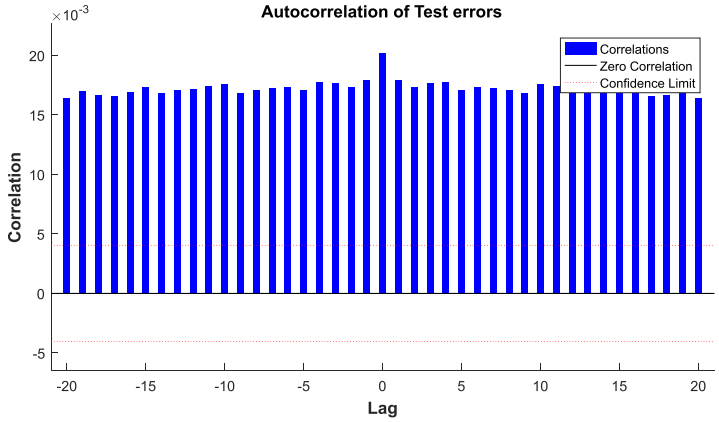

Figure 7: The autocorrelation function (ACF) of testing data

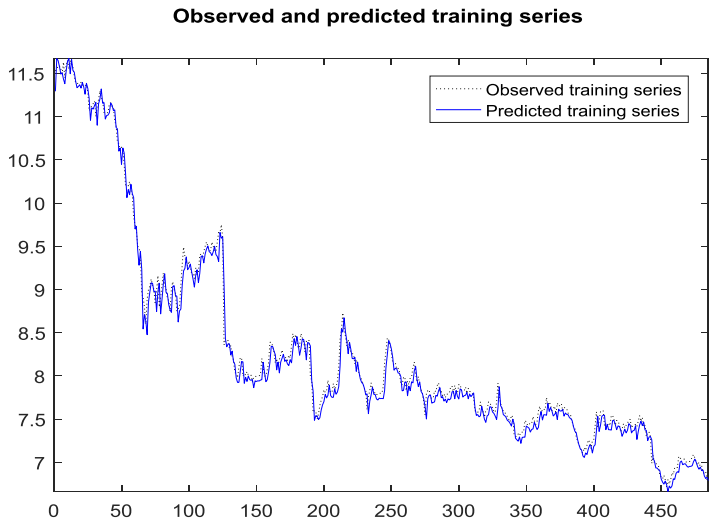

Figure 8: Graph of the observed, respectively predicted time series for the three data sets (training, validation, testing) of model

5 Conclusion and future work

In this paper, the findings reveal that the neural networks are able to extract semantic contexts from unstructured Big Data, they create and classify models and recognize complex patterns. While validating the "learning", the neural network becomes an "expert" in the date subset it was trained on, being then used to predict other new situations of the type "what if".

Therefore, the remarkable calculus power of the parallel distributed structure of a neural network is cumulated with the generalization capacity, respectively, the capacity to produce output rational values (in a reasonable convergence range) for unknown input values within the training process. These two qualities of the neural networks make possible finding reasonable solutions for complex problems operating in uncertain environments with inaccurate data, sometimes impossible to be solved by classical methods, based on exact mathematical models.

In addition, the results proved that for every training data, the network modifies its synaptic weights to minimize the difference between the desired response and the properly response of the network. The network training is done until it reaches a status when the synaptic weights do not modify significantly. In this way, the neural network builds a mapping of input data at output data from the capital market in Romania.

In conclusion, one can say that the strength of neural networks is their ability to accurately predict outcomes for share price value. In tests of accuracy performed in the present study, compared with other approaches, neural networks were able to get very good scores. Hence, using neural networks for the share price prediction of some companies is a good alternative to the traditional technical analysis used by the trading platforms.

For the future work, the study can be further explored in many direction as follows: when designing the neural network evaluation and error correction tools are necessary in order to be sure that the network is error tolerant, and in order to profit from the benefits of adaptability, the network's time constants must be long enough for the system to ignore the fake troubles.

Author contributions

The authors contributed equally to this work.

Conflict of interest

The authors declare no conflict of interest.

References

[1] Alberola, J. M.; Such, J. M.; Botti, V.; Espinosa, A.; Garcia-Fornes, A. (2013). A Scalable Multiagent Platform for Large Systems, *Computer Science and Information Systems*, 10, 51-77, 2013.

[2] Allen, F., Karjalainen R., 1999 Using Genetic Algorithms to Find Technical Trading Rules *Journal of Financial Economics*, 51, 245-271, 1999.

[3] Bahna, M.; Cepoi, C.-O.; Dumitrescu, B. A.; Damian, V. (2018). Estimating the price impact of market orders on the Bucharest Stock Exchange, *Romanian Journal of Economic Forecasting*, XXI(4), 120-133, 2018.

[4] Benoudjit, N.; Verleysen, M. (2003). On the kernel widths in radial-basis function networks, *Neural Processing Letters*, 18(2), 139-154, 2003.

[5] Boden, M. (2002). *A guide to recurrent neural networks and backpropagation*, The Dallas Project, 2002.

[6] Bordini, R.H.; Braubach, L.; Dastani, M.; Seghrouchni, A.E.F.; Gomez-Sanz, J.J.; Leite, J.; O'Hare, G.; Pokahr, A.; Ricci, A. (2006). A Survey of Programming Languages and Platforms for Multi-Agent Systems, *Informatica*, 30(1), 33–44, 2006.

[7] Dodd, O.; Gilbert, A. (2016). The Impact of Cross-Listing on the Home Market's Information Environment and Stock Price Efficiency, *Financial Review*, 51(3), 299–328, 2016.

[8] Dzitac, S.; Felea, I.; Dzitac, I. et al. (2008). An application of neuro-fuzzy modelling to prediction of some incidence in an electrical energy distribution center, *International Journal of Computers Communications & Control*, 3(S), 287-292, 2008.

[9] Georgescu, V. (2010). Robustly Forecasting the Bucharest Stock Exchange Bet Index Through a Novel Computational Intelligence Approach, *Economic Computation and Economic Cybernetics Studies and Research*, 44(3), 23-42, 2010.

[10] Georgescu, V. (2011). An Econometric Insight into Predicting Bucharest Stock Exchange Mean-Return and Volatility–Return Processes, *Economic Computation and Economic Cybernetics Studies and Research*, 3, 25-42, 2011..

[11] Grigore, L. Șt.; Soloi, A.; Tiron, O.; Răcuciu, C. I. (2013). Fundamentals of Autonomous Robot Classes with a System of Stabilization of the Gripping Mechanism, *Advanced Materials Research*, 646(1), 164-170, 2013.

[12] Haykin, S. (1994). *Neural networks: a comprehensive foundation*, Prentice Hall PTR, 1994.

[13] Haykin, S. (2009). *Neural networks and Learning machines*, Prentice Hall Publishing, 2009.

[14] Kaplan, D.; Glass, L. (2012). *Understanding nonlinear dynamics*, Springer Science & Business Media, 2012.

[15] McCulloch, W.; Pitts, W. (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity, *Bulletin of Mathematical Biophysics*, 5, 115-133, 1943.

[16] Rehar, T.; Ogrizek, B.; Leber, M.; Pisnic, A.; Buchmeister, B. (2017). Product Lifecycle Forecasting Using System's Indicators, *International Journal of Simulation Modelling*, 16(1), 45-57, 2017.

[17] Sun, Y.F.; Zhang M.L., Chen, S.; Shi, X.H. (2018). A Financial Embedded Vector Model and Its Applications to Time Series Forecasting, *International Journal of Computers Communications & Control*, 13(5), 881-894, 2018.

[18] Tsai, C.W.; Lai, C.F.; Chiang, M.C.; Yang, L.T. (2013). Data mining for internet of things: A survey, *IEEE Communications Surveys & Tutorials*, 16(1), 77-97, 2013.

[19] Tse, Y.; Tsui, A. (2002). A Multivariate GARCH Model with Time-Varying Correlations, *Journal of Business and Economic Statistics*, 20, 351-362, 2002.

[20] Vapnik, V. (2006). *Estimation of Dependences Based on Empirical Data*, Springer, 2006.

[21] Wang, SC. (2003). *Artificial Neural Network. In: Interdisciplinary Computing in Java Programming. The Springer International Series in Engineering and Computer Science*, Springer, Boston, MA., 2003.

[22] Weiss, G. (2000). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, Cambridge, Massachusetts London, England, 2000.

[23] [Online]. Bucharest Stock Exchange. Available: http://www.bvb.ro/, Accessed on 22 January 2020.

Journal's webpage: http://univagora.ro/jour/index.php/ijccc/

This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).
https://publicationethics.org/members/international-journal-computers-communications-and-control

Cite this paper as:

Petrescu, M.; Cuc, M.; Oncioiu, I.; Petrescu, A.-G.; Bîlcan, F.-R.; Ivan, L. (2020). Analytical Modelling of Share Price Value Using Computational Intelligence Methods, *International Journal of Computers Communications & Control*, 15(4), 3910, 2020.
https://doi.org/10.15837/ijccc.2020.4.3910, 2020.