Annals of Data Science (2020) 7(4):613–628
https://doi.org/10.1007/s40745-020-00305-w

Comparing Artificial Neural Network Architectures for Brazilian Stock Market Prediction

Suellen Teixeira Zavadzki de Pauli[1] · **Mariana Kleina**[1] · **Wagner Hugo Bonat**[1]

Received: 27 January 2020 / Revised: 20 June 2020 / Accepted: 25 June 2020 / Published online: 13 July 2020

Abstract

Prediction of financial time series is a great challenge for statistical models. In general, the stock market times series present high volatility due to its sensitivity to economic and political factors. Furthermore, recently, the covid-19 pandemic has caused a drastic change in the stock exchange times series. In this challenging context, several computational techniques have been proposed to improve the performance of predicting such times series. The main goal of this article is to compare the prediction performance of five neural network architectures in predicting the six most traded stocks of the official Brazilian stock exchange B3 from March 2019 to April 2020. We trained the models to predict the closing price of the next day using as inputs its own previous values. We compared the predictive performance of multiple linear regression, Elman, Jordan, radial basis function, and multilayer perceptron architectures based on the root of the mean square error. We trained all models using the training set while hyper-parameters such as the number of input variables and hidden layers were selected using the testing set. Moreover, we used the trimmed average of 100 bootstrap samples as our prediction. Thus, our approach allows us to measure the uncertainty associate with the predicted values. The results showed that for all times series, considered all architectures, except the radial basis function, the networks tunning provide suitable fit, reasonable predictions, and confidence intervals.

Keywords Artificial neural network · Forecasting · Time series · Stock market

✉ Suellen Teixeira Zavadzki de Pauli
est.suellen@gmail.com

[1] Federal University of Paraná (UFPR), Cel. Francisco H. dos Santos Avenue, 210, Curitiba, PR 81530-000, Brazil

1 Introduction

The stock market is a popular investment option for investors because of its expected high returns. In contrast, stock market prediction is a complex task to achieve with the help of artificial intelligence. It is due to stock prices depend on many factors, including trends and news in the market [1].

The prediction of financial market times series is challenging. Such information presents high volatility in the observed data, together with the uncertainty inherent in any forecast. Therefore, wrong decisions based on such predictions can have economically catastrophic consequences for individuals, institutions, and nations, as observed in the financial crises [2].

According to [3], financial institutions and banks are among those industries that have relatively complete and accurate data. Typical cases include stock investment, loan payment prediction, credit approval, bankruptcy prediction, and fraud detection. Techniques such as discriminant analysis, linear and logistic regression, entire programming, decision trees, expert systems, neural networks and dynamic models are commonly used in financial and banking applications.

It is not surprising that academia and the financial industry have invested a lot of time in proposing and comparing computational methods for analyzing available data, in order to obtain reliable predictions. Such analyses are used in the prevention of macroeconomic crises, prediction of stock market movement as well as for composing investment portfolios [2].

Since the 1950s, as computing technology was gradually used in commercial applications, many companies have developed databases to store and analyze collected data. Artificial intelligence methods were used to deal with data sets, including neural networks and decision trees. In the 90s, the term data mining came to be used, which crosses the human intervention, machine learning, mathematical modeling and databases. The investigation of the theoretical components of big data, or data science, requires interdisciplinary efforts in mathematics, sociology, economics, computer science and management science [4].

Big data has been one of the most popular topics since last several years and how to effectively conduct big data analysis is a big challenge for every field. According to [5], the process of big data analysis can be described by a general data analysis, which consists of several steps, including data acquisition and management, data access and processing, data mining and interpretation, and data applications. The modeling methods can include many techniques and the challenge reflects when and which method is appropriate. In this paper we will limit ourselves to the study of data interpretation, that still according to authors, corresponds to how to utilize rules and principles of mathematics and statistics in big data mining and interpretation. [5, 6].

According to [4], three problems are urgent to solve in order to benefit from using Big Data in science, engineering, and business applications. The first is transformation of semi- and unstructured data to structured data. The second is about complexity, uncertainty, and systematic modeling. The third, understanding of the relationship of data heterogeneity, knowledge heterogeneity, and decision heterogeneity. In this paper we focus on modeling the technique of artificial neural networks (ANN) as well as the inherent uncertainty of the model. ANN is an artificial intelligence technique for classifying, image processing and predicting the data [7].

Tkáč and Verner [8] reviewed publications where neural networks were applied in the business field, for the period from 1994 to 2015. According to the authors, in the two decades of study, artificial neural networks were widely used in several business applications. They classified the articles in 15 areas. The stocks and bonds area had the second-largest volume of publications and showed a significant increase during the study period. It was also observed that a variety of neural network models were presented in an attempt to overcome standard statistical techniques and thus obtain more assertiveness. Consequently, the popularity of new hybrid methods increased and overcame various difficulties of conventional neural networks in addition to achieving more precise results. The type of network most used, according to the authors, considering all the articles evaluated, is the feedforward multilayer and few studies have used other networks. The most common learning algorithm was the backpropagation performed by a descendent gradient search.

Artificial neural networks (ANNs) are considered flexible computational structures and universal function approximators. In general, ANNs have a high degree of assertiveness on the prediction of times series. Some advantages are that ANNs are self-adaptive, without hard initial assumptions, and can be used in nonlinear times series[9]. ANNs can learn, which occurs in an iterative process of adjustment. There are different types of networks, thus several models have been developed with variation in the learning process and network architectures [10].

ANNs are composed of neurons, which are organized hierarchically in layers. The first layer, called the input layer, receives information from the external environment, the final layer is the output layer and the intermediate are called hidden layers. In a feedforward network, for example, information occurs in one direction along with the connection. The connection forces in the network path are known as weights, therefore, the next layer neurons receive the weighted sum of the previous layer signals and this information goes through an activation function, which can return,for instance, a value between 0 and 1 as in the case of a widely used logistic activation function[11].

The main goal of this article is to compare the predictive performance of five neural network architectures in predicting six most traded stocks of the official Brazilian stock exchange B3 from March 2019 to April 2020. We highlight that the prediction period takes into account the deep crisis due to the covid-19 pandemic and, consequently, it is a challenging scenario for statistical prediction.

The covid-19 pandemic has hit the global economy. As the public health crisis worsens, measures are taken. Therefore, countries have responded by imposing measures of social distance, such as closing public places and reducing working hours, to reduce the chance of contracting the highly contagious virus in a social environment. As a result, there is a massive decline in commercial activity, causing unprecedented economic losses for the country [12].

In such a challenging period, aside to have an accurate prediction, we need proper modeling of the uncertainty associated with the model's prediction. Thus, we adopted the bootstrap method and, as predicted values, we take the trimmed average of one hundred bootstrap samples. The goal is to have a robust prediction for which we can provide a confidence interval. It is important to note that measuring the uncertainty in artificial neural network models is difficult and seldom done.

We compared the predictive performance of five neural networking architectures, namely: multiple linear regression (MLR), Elman, Jordan, radial basis function (RBF), and multilayer perceptron (MLP). For the last one, we considered three activation functions: logistic, tangent, and step. The historical times series, from January 2013 to April 2020 was split into three sets: training, testing, and validation. We trained the models using only the training set to predict the closing price of the next day. The input variables were the closing price from one to ten days ago in the additive and multiplicative forms. Furthermore, we used the testing set to tuned the number of input variables and hidden layers when suitable. We carry out an extensive search for the best model combining the number of input variables and hidden layers and compared the performance of all fitted models by root of the mean square error (RMSE) of the trimmed mean. Finally, we used the best model in terms of RMSE obtained based on each architecture to predict the validation set and compare the results graphically.

We review the artificial neural network architectures in Sect. 2. Section 3 describes the data set and the model applications to the data. Section 4 presents the main results and Sect. 5 discusses some concluding remarks.

2 Artificial Neural Network Architectures

In this section, we shall review the artificial neural network (ANN) architectures considered in our comparative study as well as the criteria for comparing them.

Probably, the simplest ANN architecture is the one corresponding to a multiple linear regression model, as presented in Eq. 1, where the activation function of the output layer is the identity, represented by f_o and there are no hidden layers

$$y_i = f_o(\beta_j, x_{ij}). \tag{1}$$

The independent observations of the interest variable (response) are denoted by y_i, $i = 1, \dots, n$, x_{ij} where $j = 1, \dots, p$ is a $p \times 1$ vector of explanatory/input variables. Similarly, β is a $p \times 1$ vector of unknown regression coefficients or weights[13]. Figure 1 presents the architecture of a multiple linear regression model.

The parameter estimation for this model is usually performed using maximum likelihood method. The model has a simple structure which makes it possible to estimate the model's parameters analytically. The other models in the sequence need numeric methods for parameter estimation.

Probably, the most popular ANN is the multilayer perceptron (MLP) network that learns from error correction. The propagation rule of this network occours through the internal product of the weighted inputs, adding the bias term. It should be considered that the output of the previous layer is the current layer input[14]. MLP training takes place in three stages: feedforward, backpropagation, and weights adjustment. In

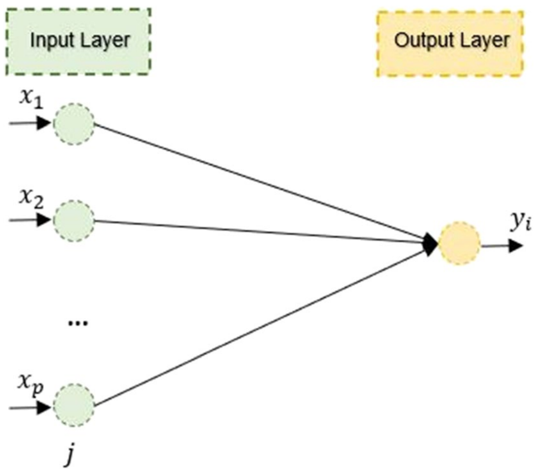

Fig. 1 Multiple linear regression model architecture

the first, the pattern training with the inputs occurs. In the second, the error correction backpropagation occurs, and finally, in the third, the weights are adjusted[15]. The MLP structure is presented in Fig. 2. In this example, there is a hidden layer, as used in our data analysis. It has a single output neuron, as our application predicts one day ahead.

In the feedforward phase, for each $k = 1, 2 \ldots, m$ hidden layer neuron there are p input variables x_j, where $j = 1, 2 \ldots, p$ and their respective synaptic weights w_{jk} which in turn results in the linear combination as in Eqs. 2 and 3. The bias term w_0k, increases or decreases the net input of the activation function and permits the displacement in the hyperplane of the origin on the abscissa axis[14].

$$y_i = f_o(\beta_j, z_{.k}), \tag{2}$$

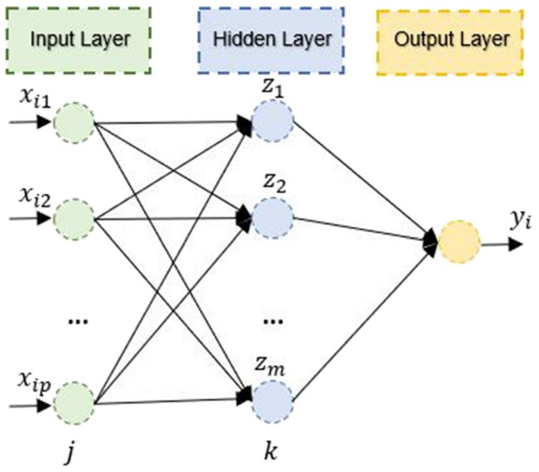

Fig. 2 MLP architecture with an output layer neuron

$$z_{.k} = f_h(w_{jk}, x_{ij}). \tag{3}$$

The neuron k output signal is given by $f_h(w_{jk}, x_{ij})$, where $f_h(.)$ corresponds to an activation function. In our analysis, we considered three types of activation functions. The logistic (Eq. 4), tangent (Eq. 5) and, step functions (Eq. 6) [16].

$$f_h(w_{jk}, x_{ij}) = \frac{1}{1 + e^{-w_{jk}x_{ij}}}, \tag{4}$$

$$f_h(w_{jk}, x_{ij}) = \frac{e^{w_{jk}x_{ij}} - e^{-w_{jk}x_{ij}}}{e^{w_{jk}x_{ij}} + e^{-w_{jk}x_{ij}}}, \tag{5}$$

$$f_h(w_{jk}, x_{ij}) = \begin{cases} 1, & \text{if } w_{jk}x_{ij} > 0, \\ 0, & \text{if } w_{jk}x_{ij} \leq 0. \end{cases} \tag{6}$$

A similar process occurs for the output layer, however, the activation functions need not be the same. In our analysis, we used the identity function in the output layer.

After a given input set passes through the network, there is a computed output and the difference between the network output and the actual one corresponds to the error. The ultimate goal is to find the set of weights that ensures that, for each set of inputs, the output produced by the network is as close as possible to the desired output[11]. The model fitting is done, generally, by a gradient method (see, Equation 7), where η is the learning rate parameter and E is some error function[17].

$$\Delta w_{jk} = \eta \frac{\partial E}{\partial w_{jk}}. \tag{7}$$

The Elman neural network (ENN) was proposed in 1990 by Elman [18]. It is a recurrent dynamic network whose structure includes a context layer. Recurrent neural networks are a set of feedforward neural networks, which have recurring edges that span adjacent time steps, thus introducing a notion of time for the model[19]. The input layer is divided into two parts: the input neurons as in the MLP and the context neurons, which have a copy of the activation of the previously hidden layer neurons[20]. The Equations 8 and 9 show the calculations, where $z_{.k}^{(t-1)}$ refers to $z_{.k}$ in the previous iteration and $w_{z_{.k}k}$ is the respective weights

$$y_i = f_o(\beta_j, z_{.k}), \tag{8}$$

$$z_{.k} = f_h(w_{jk}, x_{ij} + z_{.k}^{(t-1)} w_{z_{.k}k}). \tag{9}$$

For this network, input and context neurons activate hidden neurons, and hidden neurons advance to activate output neurons, which also feed to activate context neurons, so the context layer allows the ENN to adapt to dynamic characteristics[21]. The great advantage of ENN is that context nodes can be used to memorize previous activation of hidden layer nodes[22]. Figure 3 shows the ENN architecture, where

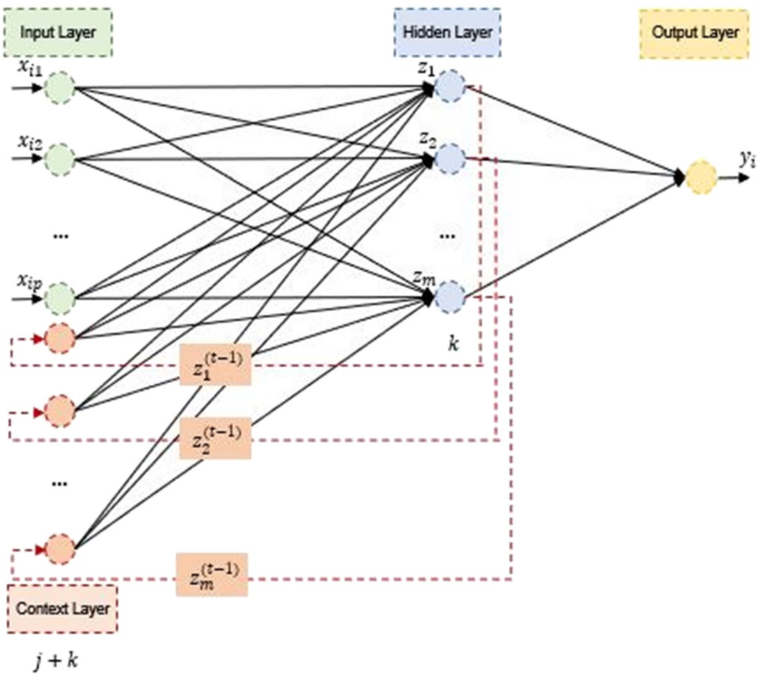

Fig. 3 Elman neural network architecture

z^{-1} consists of a unit draw, corresponding to the context layer. Model fitting is done using the gradient method.

Similarly, the Jordan neural network (JNN) was proposed in 1986 by Jordan[23]. It resembles a feedforward network with a single hidden layer, the difference is in adding the context layer. These units are feed from the output layer, see Fig. 4 [19]. There is an unit delay, corresponding to the context layer and represented by z^{-1}. Model fitting is done, generally, by a gradient method.

Finally, the radial basis functions neural networks (RBF) were proposed by Broomhead and Lowe in 1988[24]. Are feedforward networks whose structure is similar to an MLP, the main difference being that RBF has a hidden layer with nodes called RBF units. The hidden unit measures the distance between an input data vector and the center of its RBF. They are usually composed of three layers: an input layer, a single hidden layer whose processing is nonlinear, and an output layer, for details, see [25] and [26]. The activation function is not sigmoid, but radially symmetric often the Gaussian distributions[27].

The output neuron consists of the weighted sum of the hidden layer outputs, as shown in Eqs. 10 and 11, where $x \epsilon \mathfrak{R}^{n\times 1}$ is the input layer vector, $\phi_k(.)$ is the radial base function and c_k represents the RBF centers in the input vector space, ||. || denotes the Euclidean norm[25].

$$y_i = f(\beta_j, \phi_k(x_{ij}, c_k)), \tag{10}$$

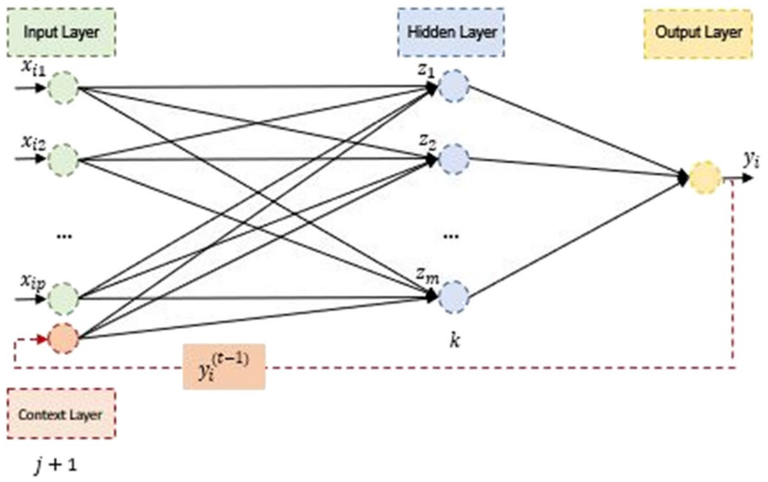

Fig. 4 Jordan neural network architecture

$$\phi_k(x_{ij}, c_k) = \phi_k(\| x_{ij} - c_k \|). \tag{11}$$

A Gaussian function is typically used as the hidden layer radial basis function. Each neuron in the hidden layer is associated with a kernel function $\phi_k(.)$ characterized by a center c_k and a width σ_k^2 as in Eq. 12. The weights are optimized using the least mean square algorithm with RBF unit centers already defined[26].

$$\phi_k(x_{ij}, c_k) = \exp(-1/2\sigma_k^2(\| x_{ij} - c_k \|^2)). \tag{12}$$

3 Design of the Comparative Study

To compare the neural network architectures, we use data of the six most traded stocks in the covid-19 pandemic period from March 16 to 27, 2020. The data set is available on the official Brazilian stock exchange B3 website. The first week evaluated was from 16 to 20 March 2020 and the second one from 23 to 27 March 2020. The frequency that each stock appeared in the evaluated period is presented in Table 1.

The six selected stocks are part of large sectors of the Brazilian economy, as shown in Table 2, where the trading codes are also presented. So, we could say that we have a representative set of stocks.

Figure 5 presents the times series plot from January 2013 to April 2020 for the six stocks considered. The data set consists of 88 months of a daily closing price considering Monday to Friday, when the market is opened.

As signaled by dashes in the Fig. 5, the data were split into three sets: training (70%), test (15%) and validation (15%). We considered and compared a bunch of model configurations to forecast the stocks the next day. Five different neural

Table 1 The six most traded Brazilian stocks from 16 to 27 March 2020

Stock	Week 1	Week 2	Total
ISHARES BOVA	5	5	10
ITAÚ UNIBANCO	5	5	10
PETROBRÁS	5	5	10
VALE	5	5	10
B3	3	2	5
BRADESCO	2	1	3

Table 2 Selected stocks, trading codes and sectors

Stock	Code	Main activity
ISHARES BOVA	BOVA11	Bovespa index
ITAÚ UNIBANCO	ITUB4	Financial/financial intermediaries/banks
PETROBRÁS	PETR4	Oil/gas and energy
VALE	VALE3	Basic materials/mining/metallic minerals
B3	B3SA3	Financial/miscellaneous financial services
BRADESCO	BBDC4	Financial/financial intermediaries/banks

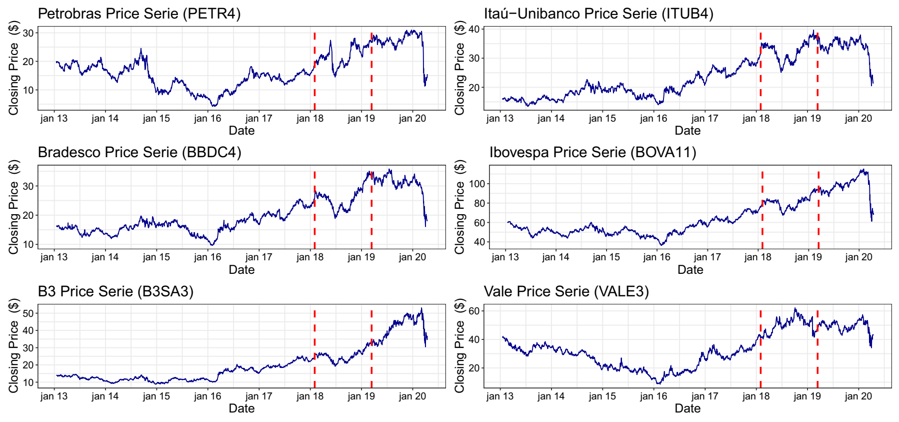

Fig. 5 Times series plots of the stocks series selected for the analysis

networks were applied, one of them, the MLP neural network was trained using three different activation functions, which resulted in a total of seven networks types, as shown in Fig. 6.

For each neural network type, we considered twenty possibilities for the input variables. The input variables were the previous closing price of the series from one to ten days ago, each with two possibilities: "additive" and "multiplicative" forms. The "additive" form corresponds to the standard format of neural networks, while

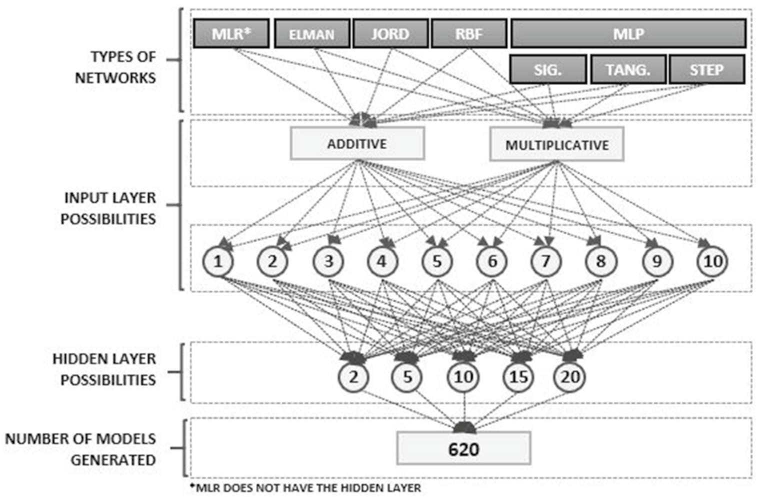
TYPES OF NETWORKS
MLR*
ELMAN
JORD
RBF
MLP
SIG.
TANG.
STEP
INPUT LAYER POSSIBILITIES
ADDITIVE
MULTIPLICATIVE
1
2
3
4
5
6
7
8
9
10
HIDDEN LAYER POSSIBILITIES
2
5
10
15
20
NUMBER OF MODELS GENERATED
620
*MLR DOES NOT HAVE THE HIDDEN LAYER

Fig. 6 Configurations of the neural network models considered

the “multiplicative” form, in addition to the main variable effects, the input layer also includes all possible iterative effects between them. Finally, we considered models with 2, 5, 10, 15, and 20 neurons, except in the MLR model, which has no hidden layer. Combining all these possibilities, we obtained 620 models that were evaluated for each of the six stocks. Figure 6 summarizes the considered models.

The computational implementation was done in the statistical software `R`[28]. The data sets and `R` code are available as a supplementary material for this paper. The data sets were obtained using the function `getSymbols()` from the `quantmod` [29] package, whose data come from Yahoo! Finance. For fitting the neural network models, we use the package `RSNNS` [30] which includes the networks MLP, Elman, Jordan and RBF. In the MLR model, the fitting was done by the maximum likelihood method as implemented in the `lm()` function, for more detail see [31]. For more details about ANNs learning by the `RSNNS` package see [16].

For each of the 620 model configurations, a 95% confidence interval was obtained based on 100 bootstrap samples. To have a robust predicted value we used the trimmed average of the 100 bootstrap samples. All models were trained based only on the training set. The test set was used to tune the model hyperparameters, i.e. number of input and hidden layers. The root of the mean square error (RMSE) obtained by predicting trimmed mean of 100 bootstrap samples was used as a comparative measurement.

4 Results

In this section, we present the main results obtained in the comparative study described in Sect. 3. Figures 7 and 8 show the RMSE for all models and times series considered.

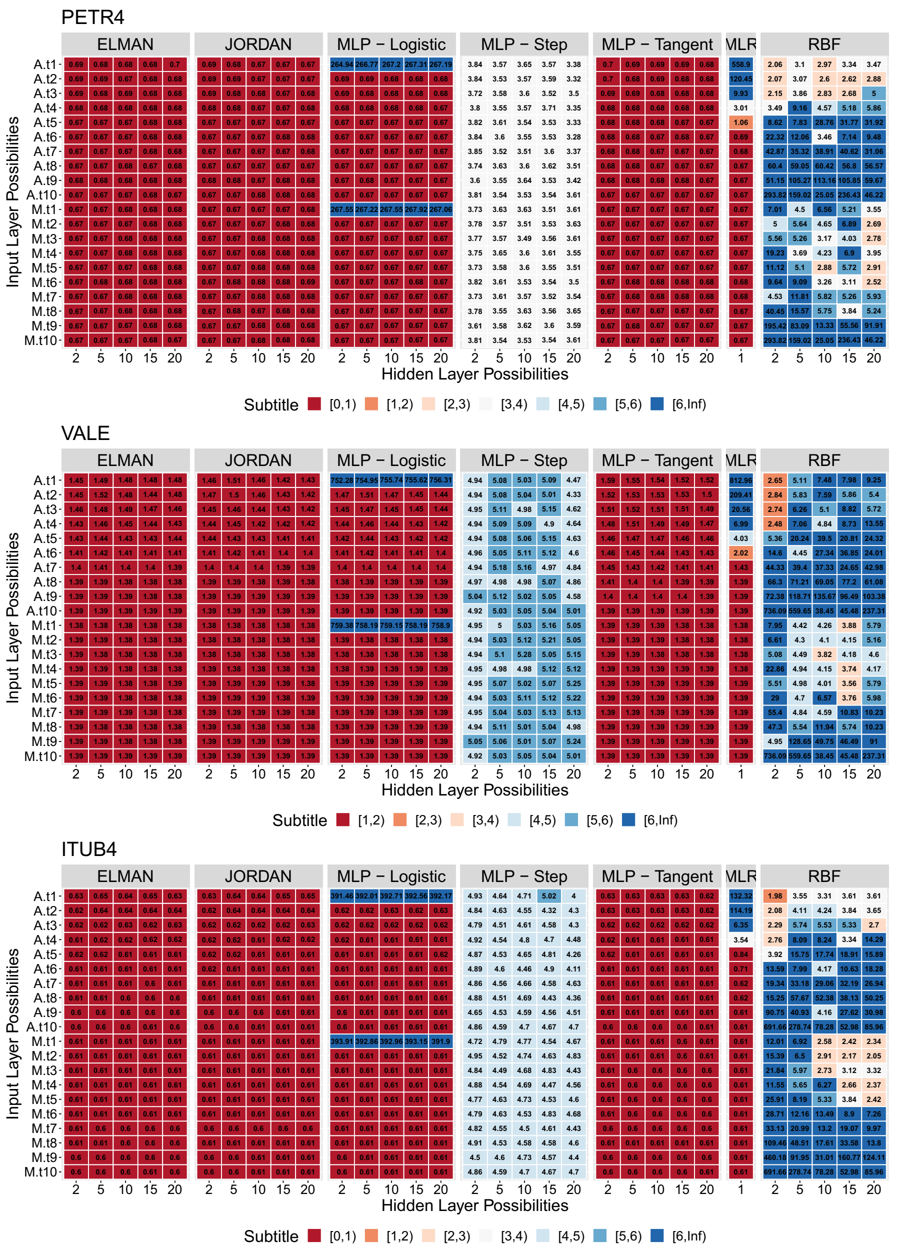

Fig. 7 RMSE of all model configurations for PETR4, VALE and ITUB4

The results presented in Figs. 7 and 8 show that overall the approaches MLP-step and RBF did not provide a suitable fit for all the times series analyzed. Concerning the action function used for specifying the MPL networks, we note that the sigmoid logistics did not fit well in some cases, on the other hand, the MLP-tangent fits well

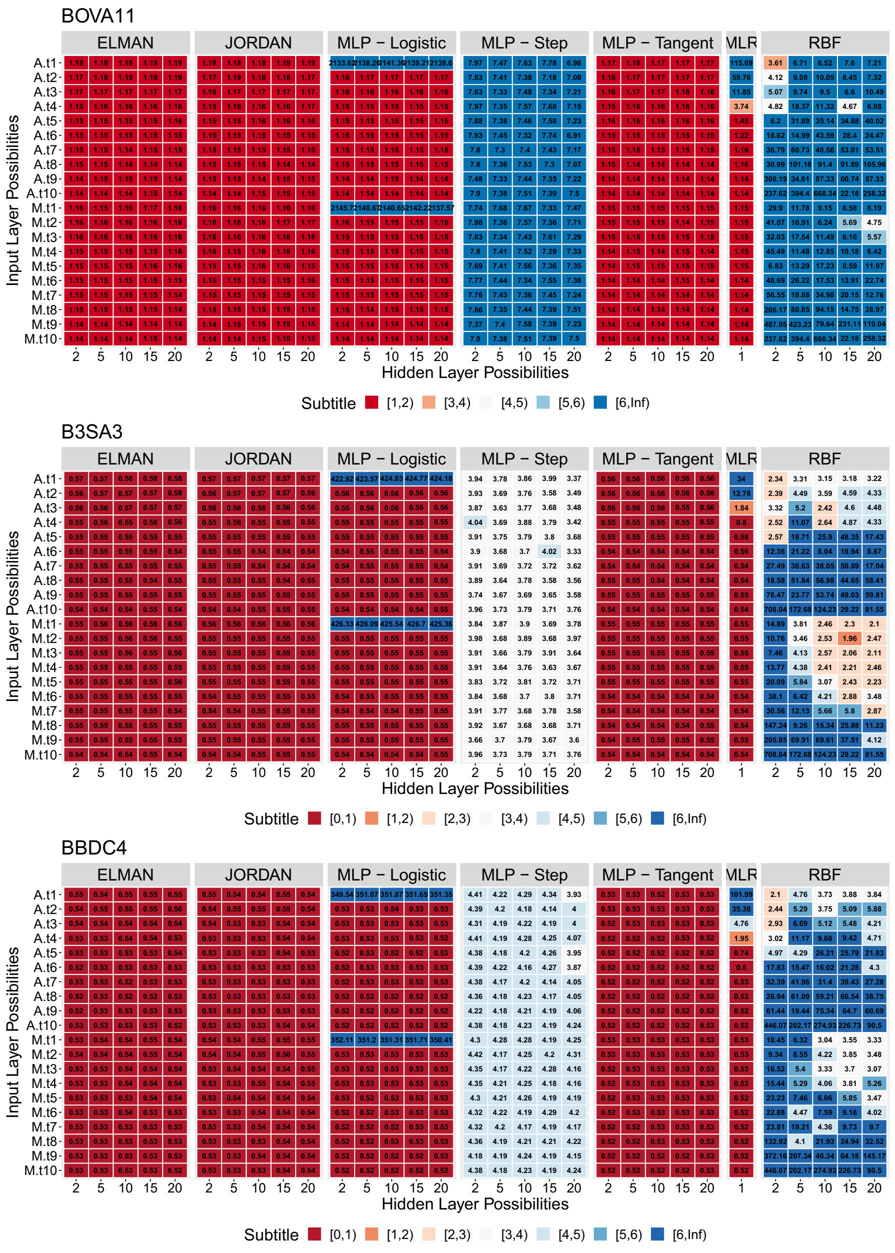

Fig. 8 RMSE of all model configurations for BOVA11, B3SA3 and BBDC4 stock

for virtually all series and model specifications. Thus, we decided to continue to describe the results focusing on the ELMAN, JORDAN, MLP-tangent, and MLR approaches. Table 3 presents the RMSE for the best fit obtained based on each neural network architectures.

Table 3 RMSE, number of input and hidden layers for the best fit based on each neural network architecture by times series. M stands for multiplicative and A for additive. tX stands for the number of lagged input variables

	PETR4				ITUB4			
	Test	Validation	Input	Hidden	Test	Validation	Input	Hidden
ELMAN	0.67	0.83	M.t5	2	**0.60**	**0.69**	**A.t4**	**20**
JORDAN	0.67	0.83	M.t5	2	0.60	0.71	A.t4	20
MLP-TANG.	0.67	0.83	A.t10	2	0.60	0.69	A.t7	20
MLR	**0.67**	**0.79**	**A.t8**	**1**	0.61	0.69	A.t4	1
RBF	2.06	4.75	M.t10	2	1.98	3.46	M.t10	2
	BBDC4				BOVA11			
ELMAN	0.52	0.70	M.t3	15	1.14	2.09	M.t3	15
JORDAN	0.53	0.71	M.t5	2	1.14	2.09	M.t3	15
MLP-TANG.	0.52	0.73	M.t6	2	1.14	2.11	A.t1	5
MLR	**0.52**	**0.71**	**M.t1**	**1**	**1.14**	**2.10**	**A.t2**	**1**
RBF	2.10	2.40	M.t10	2	3.61	6.28	M.t10	2
	B3SA3				VALE3			
ELMAN	**0.54**	**1.33**	**A.t1**	**2**	1.38	1.38	A.t9	15
JORDAN	0.54	1.33	M.t4	20	1.38	1.40	A.t10	20
MLP-TANG.	0.54	1.36	M.t5	15	1.38	1.41	A.t9	20
MLR	0.54	1.33	A.t4	1	**1.38**	**1.40**	**A.t8**	**1**
RBF	1.96	3.43	A.t9	15	2.48	1.83	M.t7	2

Bold represents the best solution for each stock

The results show that BOVA11, B3SA3, and VALE3 were the most difficult for forecasting, i.e. they presented larger RMSE than the other times series considered. Moreover, for B3SA3, despite the small RMSE for the test set it double for the validation set for all considered models. The Elman, Jordan, MLP, and MLR networks presented quite similar RMSE for all times series analyzed. Finally, it is interesting to note that, although the simplest one, the MLR provides the best or pretty similar to best fit for all times series analyzed.

Figure 9 presents the times series plot for the validation set along with the predicted values and 95% confidence intervals. The models capture the swing in the data even in March when the times series present a deep decrease due to the covid-19 pandemic. It is interesting to note that the confidence intervals are quite narrow, which probably shows that the uncertainty is underestimated for all times series.

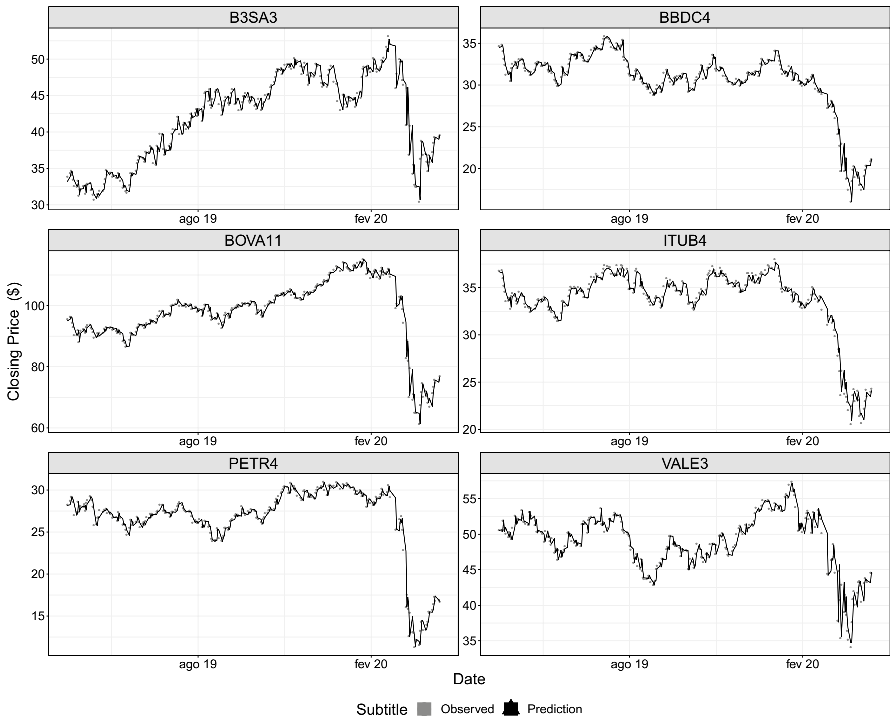

Fig. 9 Observed and predicted values based on the best model by times series—validation data set

5 Discussion

The closing price forecasting is valuable information in the financial market and a challenging task even more in the middle of a pandemic. In this paper, we compared the predictive performance of five neural network architectures to predict the six most traded stocks of the official Brazilian stock exchange B3 from March 2019 to April 2020. By combining different input variables and hidden layers we trained 620 models and selected the best to predict each times series based on the RMSE. The selected models provided reasonable prediction and confidence intervals, even for the pandemic period. Thus, our results suggest that such models could be a valuable tool for the financial market.

Probably, the main difference of our approach concerning the common computational methods used for forecasting in the literature is the use of the bootstrap method to provide robust predicted values and measure the uncertainty associated with the predictions. The great advantage of such methods is their simplicity in the sense that they need only the times series for training. Of course, they are limited tools, as they rely in the past for prediction. Consequently, we suggest to use them only as a support tool.

Concerning the comparison of the five neural network architectures, our results showed that the MLR although the simplest one may be tuned to provide results quite similar to more complicated models such as the MLP, ELMAN, and JORDAN neural networks, at least for the times series considered. The great advantage of the MLR model is that it is possible to estimate the parameters analytically, besides that, it is easier to interpret the MLR than the others neural networks. This model has a simple configuration and is cheap computationally, which made it the most satisfactory model for our analyses.

As future work, we suggest to consider other architectures such as convolutional and recurrent neural networks and apply the framework for other times series. Moreover, the evaluation of other techniques to compute confidence intervals for the predicted values are highly demanded.

References

1. Saini A, Sharma A (2019) Predicting the unpredictable: an application of machine learning algorithms in Indian stock market. Ann Data Sci 6:1–9
2. Podsiadlo M, Rybinski H (2016) Financial time series forecasting using rough sets with time-weighted rule voting. Expert Syst Appl 66:219–233
3. Shi Y, Tian Y, Kou G, Peng Y, Li J (2011) Optimization based data mining: theory and applications. Springer, Berlin
4. Shi Y (2014) Big data: history, current status, and challenges going forward. Bridge 44(4):6–11
5. Xu Z, Shi Y (2015) Exploring big data analysis: fundamental scientific problems. Ann Data Sci 2(4):363–372
6. Olson DL, Shi Y, Shi Y (2007) Introduction to business data mining, vol 10. McGraw-Hill/Irwin Englewood Cliffs, New York
7. Aziz R, Verma C, Srivastava N (2018) Artificial neural network classification of high dimensional data with novel optimization approach of dimension reduction. Ann Data Sci 5(4):615–635
8. Tkáč M, Verner R (2016) Artificial neural networks in business: two decades of research. Appl Soft Comput 38:788–804
9. Khashei M, Bijari M (2010) An artificial neural network (p, d, q) model for timeseries forecasting. Expert Syst Appl 37(1):479–489
10. Braga AP, Carvalho ACPLF, Ludermir TB (2000) Redes neurais artificiais: teoria e aplicações, 2nd edn. Editora, Rio de Janeiro
11. Cooper JC (1999) Artificial neural networks versus multivariate statistics: an application from economics. J Appl Stat 26(8):909–921
12. Liu Y, Gu Z, Xia S, Shi B, Zhou XN, Shi Y, Liu J (2020) What are the underlying transmission patterns of covid-19 outbreak?—An age-specific social contact characterization. EClinicalMedicine 18:100354
13. Ertaş H, Toker S, Kaçıranlar S (2015) Robust two parameter ridge m-estimator for linear regression. J Appl Stat 42(7):1490–1502
14. Haykin S (2001) Redes neurais: princípios e práticas. Trad. Paulo Martins Engel, 2 edn. Bookman, Porto Alegre
15. Fausett LV et al (1994) Fundamentals of neural networks: architectures, algorithms, and applications, vol 3. Prentice-Hall, Englewood Cliffs
16. Zell Aea (1998) http://www.ra.cs.uni-tuebingen.de/SNNS/welcome.html
17. Haykin SS et al. (2009) Neural networks and learning machines/Simon Haykin
18. Elman JL (1990) Finding structure in time. Cogn Sci 14(2):179–211
19. Lipton ZC, Berkowitz J, Elkan C (2015) A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019 (2015)
20. Toha SF, Tokhi MO (2008) MLP and Elman recurrent neural network modelling for the TRMS. In: 2008 7th IEEE international conference on cybernetic intelligent systems, pp 1–6. IEEE

21. Yu C, Li Y, Zhang M (2017) An improved wavelet transform using singular spectrum analysis for wind speed forecasting based on elman neural network. Energy Convers Manag 148:895–904
22. Chong S, Rui S, Jie L, Xiaoming Z, Jun T, Yunbo S, Jun L, Huiliang C (2016) Temperature drift modeling of mems gyroscope based on genetic-Elman neural network. Mech Syst Signal Process 72:897–905
23. Jordan M (1986) Finding structure in time. In: Proceedings of the eighth annual conference of the cognitive science society, pp 531–546
24. Broomhead DS, Lowe D (1988) Radial basis functions, multi-variable functional interpolation and adaptive networks. Technical Report, Royal Signals and Radar Establishment Malvern (United Kingdom)
25. Wu J, Long J, Liu M (2015) Evolving rbf neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm. Neurocomputing 148:136–142
26. Vahabli E, Rahmati S (2016) Application of an rbf neural network for fdm parts' surface roughness prediction for enhancing surface quality. Int J Precis Eng Manuf 17(12):1589–1603
27. Bergmeir C, Benítez JM (2012) Neural networks in R using the Stuttgart neural network simulator: RSNNS. J Stat Softw 46(7):1–26
28. R Core Team (2019) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/
29. Ryan JA, Ulrich JM, Thielen W, Teetor P, Bronder S, Ulrich MJM (2020) Package 'quantmod'
30. Bergmeir CN, Benítez Sánchez JM et al (2012) Neural networks in R using the Stuttgart neural network simulator: RSNNS. American Statistical Association, Alexandria
31. Pawitan Y (2001) In all likelihood: statistical modelling and inference using likelihood. Oxford University Press, Oxford