Recommending Buy/Sell in Brazilian Stock Market through Long Short-Term Memory

Sandro da Silva Camargo[1] [0000−0001−8871−3950] and Gabriel Lopes Silva[1]

Programa de Pós-Graduação em Computação Aplicada (PPGCAP),
Universidade Federal do Pampa (UNIPAMPA) & EMBRAPA Pecuária Sul
Bagé, Rio Grande do Sul, Brasil
sandrocamargo@unipampa.edu.br,gabriel18.lopes@gmail.com

Abstract This work aims to evaluate the accuracy of Long Short-Term Memory Neural Networks to recommend Buy/Sell signals of some Brazilian Stock Market Blue Chips. The population of this study was composed by top 5 volume stocks, which represented nearly 40% of the total volume of Brazilian Stock Market in 2019. It was analyzed the following features: volume traded, closing and opening price, maximum and minimum price, and last five-day closing prices. Models created can forecast the next day's opening or closing price. Obtained results show that forecasting and real values have a coefficient of determination (R^2) from 0.91 to 0.99, depending on the stock.

Keywords: Variable Income, Bovespa, Time Series, Recurrent Neural Networks, Finance.

1 Introduction

The Stock Exchange is an institution which intermediates the sale and purchase of goods, such as agricultural products, raw materials and securities [9]. One of its divisions, the capital market, is responsible for the securities distribution, such as shares, debentures, short-term notes, among others. This market aims to raise capital for companies and provide profits for investors. The stock market is one of the subsets of the capital market, in which the variation of shares and the sharing of profits are the reason for investors to negotiate in this market.

In order to predict stock prices in the stock market, these prices can be analyzed as a time series, to which different techniques and methods have been proposed. However, stock market, exchange prices and time series indicators are influenced by several factors, becoming their forecasting complex [17]. Edwards et al. [6] present two techniques which tries to predict these movements. The first one is the technical analysis, which is based on the observation of charts from past periods to try to predict the future market trends. The second one is the fundamental analysis, which relies on company economic factors and observations of its transactions, in order to predict the fair price to the shares. Due to human limits to mathematical processing, computers are being used to facilitate

Received January 2023 Accepted April 2023; Published May 2023

and to increase the performance of technical analysis. Dow Theory is the broadly used by stock market technical analysts [23]. According this theory, it is possible to observe that stock market has patterns and trends. Hsu et al. [14] prove machine learning techniques are more efficient than purely observation-based economical methods. Among these machine learning techniques, Recurrent Neural Networks (RNN) has been broadly used in the task of predicting the stock market [1]. RNN can identify patterns in time series and, based on them, make predictions. So if an RNN is trained in a stock market dataset, it is expected that it can recognize these patterns and be able to predict future trends. However, the accuracy of this technique can vary according few factors as: market trend (ascending, stable or descending); country or region socioeconomic profile, country economic philosophy (capitalist, socialist, communist, among others). Moreover, other happenings as crises, wars, and public health problems may affect the performance of machine learning techniques [5, 2, 11].

Deep Learning techniques have been increasingly used to forecast different aspects of stock market [13, 14, 16, 24, 29]. These techniques presents the advantage of not only mapping linear patterns, as many other machine learning techniques, but also learn with nonlinear patterns. Among these techniques, Long Short-Term Memory (LSTM) Recurrent Neural Network are being increasingly used in stock market forecasting [18, 28, 15, 27, 12, 4]. In this context, this work aims to evaluate the LSTM Recurrent Neural Network to predict blue chips opening and closing prices in Brazilian Stock Market. In order to test this hypothesis, the top 5 volume stocks in 2019 were used.

This paper is organized as follows: Section 2 describes the methodology applied in this work. In Section 3, obtained results are presented and discussed. In Section 4, conclusions and future works are exposed.

2 Materials & Methods

This section presents classifications of this research, specifies the Dataset and details the proposed approach. Based on research classification definitions [20], this research is classified as a applied research, in the sense it will test a hypothesis to solve a real problem. In the objectives viewpoint, this research is classified as a descriptive research, which aims to describe the use of neural networks to recommend sell/buy in the Brazilian stock market. According to the results sought, it can be classified as quantitative, since it generates numerical quantifiable values, which will provide evidences to evaluate the performance of the proposed approach. According to the research design, it is experimental, since different setting for the proposed approach will be tested aiming to produce the more accurate predictions.

2.1 Dataset

A Dataset containing historical stock exchange quotations, from 2015 to 2019, was downloaded from the official Brazilian stock market[1]. According to Table 1, the data elements used are stock name, date, opening and closing prices, maximum and minimum prices, and volume traded. In order to limit the amount of stocks to this study, the top 5 volume stocks were selected: VALE3, ITUB4, BBDC4, PETR3, and ABEV3. These stocks represented in 2019, respectively, 9.86%, 9.64%, 8.66%, 5.47%, 4.75% of Brazilian Stock Market negotiations, regarding traded value.

Some published papers were used as reference to define the time period to forecasting [19, 21, 7]. They used a sliding window less than or equals to five working days to make decisions under a set of stocks. Pan et al. [19] justify that it is possible to forecast short period patterns, up to one day, when analyzing periods of less than 5 days. Data were normalized in the range between 0 and 1 because it is a requirement for LSTM technique.

Table 1: Dataset Description

Feature	Description
Stock	Stock Name
Date	Day/Month/Year
Opening	Stock Opening Price
Closing	Stock Closing Price
Maximum	Higher Price
Minimum	Lower Price
Volume	Amount of stocks traded

2.2 Experimental Environment

The *google colab environment*[2] was used to run the experiments reported in this paper. Hardware resources available involves 12.7GB of System RAM, a Tesla T4 GPU with 15GB of RAM and 78.2 GB of Disk. Regarding software resources, *Python* language[3], version 3.9.16, was used to develop the models. Tensorflow library[4], version 2.11.0, was used to due to being an open source platform for machine learning. *Pandas* library[5], version 1.4.4, was used for data analysis and data manipulation. *Numpy* library[6], version 1.22.4, provides a series of functions

[1] https://www.b3.com.br/pt_br/market-data-e-indices/servicos-de-dados/market-data/historico/mercado-a-vista/cotacoes-historicas/
[2] https://colab.research.google.com
[3] https://www.python.org/
[4] https://www.tensorflow.org/
[5] https://pandas.pydata.org/
[6] https://numpy.org/

to facilitate and speed up the process of calculations between vectors and matrices, which are useful in the model training process. *Keras* library was used because it allows creating the RNN through a simple setting hyperparameters, avoiding the implementation of data structures, learning functions, activation, among others.

2.3 Proposed Approaches

Different LSTM models were trained and tested. Both approaches presented in this work, Figure 1, received the same input values, namely: trading volume, opening price, closing price, maximum price, and minimum price, which are represented with a yellow background in the figure. All these values refer to the last five business days. The only difference between approach 1 and approach 2 is the forecasting target. Approach 1 aims to predict the closing price, which is represented with a blue background. Otherwise, approach 2 aims to predict the opening price, which is represented with a green background.

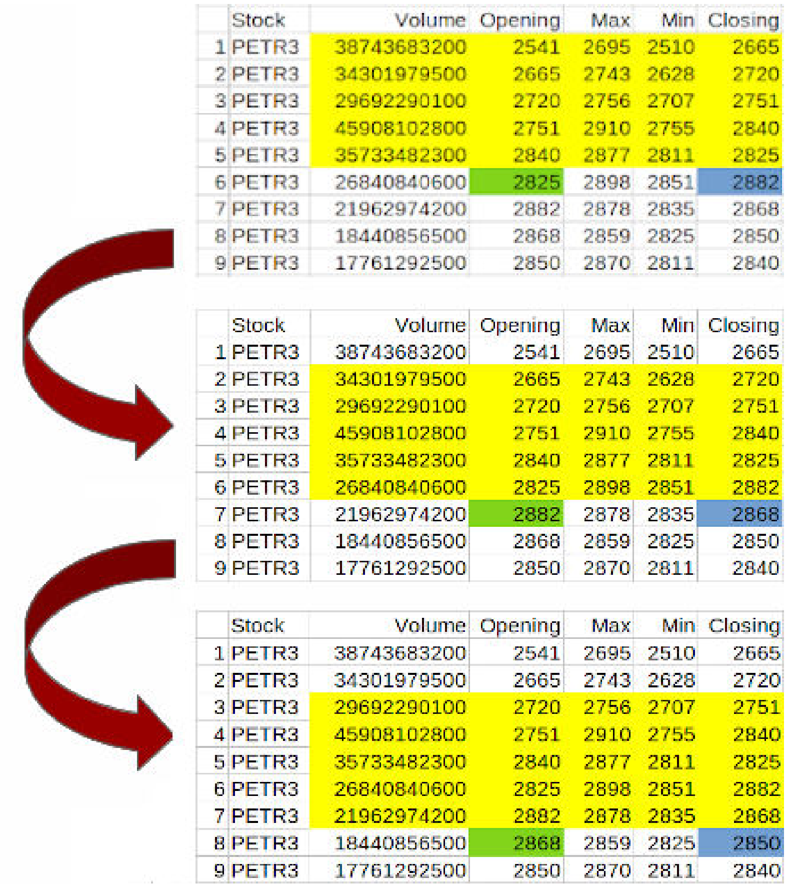
Figure 1 presents the training process schema. The curved red arrows represent the steps for training with the sliding window. To each step, the window slides one day forward in order to select the inputs, which are the independent variable emphasized with a yellow background, and the desired output, which is the dependent variable emphasized with a green background. To each interaction, five days are used as input and the sixth day opening or closing value is used as the output to be forecast.

2.4 Hyperparameters Tuning

Algorithm hyperparameters must be defined by the user as an external element to the model and whose value cannot be estimated from data [10]. According Reimers and others [22], tuning the hyperparameters has a fundamental importance in the network performance. However, this tuning process is a computationally expensive task, due to the amount of possible hyperparameters, like: amount of layers, amount of neurons in each layer, setting activation function, dropout tuning, and loss function. In order to overcome this problem, some heuristics were developed[8]. Moreover, the range of possible values to each hyperparameter is also important. Bergstra and others [3] used three heuristics to optimize hyperparameters. Two of them are sequential and based on greedy search algorithms and grid search. The remaining heuristics uses a brute force random search strategy.

Grid-search is the process of finding the optimal hyperparameters of a model which results in the most accurate forecast. The range of possible values to each hyperparameter must be previously defined. Regarding neural networks, one model is trained with each possible combination of hyperparameters. The accuracy of each model is evaluated. After this process, the hyperparameters combination which represents the most accurate model is known.

Fig. 1: Forecasting the closing (Approach 1) or the opening value (Approach 2) for the next day [25].



	Stock	Volume	Opening	Max	Min	Closing
1	PETR3	38743683200	2541	2695	2510	2665
2	PETR3	34301979500	2665	2743	2628	2720
3	PETR3	29692290100	2720	2756	2707	2751
4	PETR3	45908102800	2751	2910	2755	2840
5	PETR3	35733482300	2840	2877	2811	2825
6	PETR3	26840840600	2825	2898	2851	2882
7	PETR3	21962974200	2882	2878	2835	2868
8	PETR3	18440856500	2868	2859	2825	2850
9	PETR3	17761292500	2850	2870	2811	2840

	Stock	Volume	Opening	Max	Min	Closing
1	PETR3	38743683200	2541	2695	2510	2665
2	PETR3	34301979500	2665	2743	2628	2720
3	PETR3	29692290100	2720	2756	2707	2751
4	PETR3	45908102800	2751	2910	2755	2840
5	PETR3	35733482300	2840	2877	2811	2825
6	PETR3	26840840600	2825	2898	2851	2882
7	PETR3	21962974200	2882	2878	2835	2868
8	PETR3	18440856500	2868	2859	2825	2850
9	PETR3	17761292500	2850	2870	2811	2840

	Stock	Volume	Opening	Max	Min	Closing
1	PETR3	38743683200	2541	2695	2510	2665
2	PETR3	34301979500	2665	2743	2628	2720
3	PETR3	29692290100	2720	2756	2707	2751
4	PETR3	45908102800	2751	2910	2755	2840
5	PETR3	35733482300	2840	2877	2811	2825
6	PETR3	26840840600	2825	2898	2851	2882
7	PETR3	21962974200	2882	2878	2835	2868
8	PETR3	18440856500	2868	2859	2825	2850
9	PETR3	17761292500	2850	2870	2811	2840

3 Results & Discussion

This section aims to present how the proposed method was implemented, as well as performed tests.

3.1 Training & Test

In order to perform the training and test of both proposals, dataset was split with holdout method. Samples of years from 2015 to 2018 were used for training, while samples of year 2019 were used for model testing. Some statistics were extracted that indicate the complexity of forecast a given stock. Table 2 presents stock price standard deviations, for training and test sets, and R^2. Standard deviations allows to understand how different are the variations in training and test sets. The more similar are the Standard Deviations in the sets, the more similar are the stock variations and, theoretically, the more accurate can be the models when predicting prices in the test set. In the other hand, if the standard deviation in test set is significantly higher then in training set, it is expected a lower accuracy of the model in the test set. In order to identify what is a minimum acceptable accuracy for the models, a baseline was proposed. Baseline R^2 represents the determination coefficient when comparing the price to be forecast with the closing price in the next day. When this value is close to 1, less variations happens between prices in adjacent days. So, R^2 is the baseline to be overcome by the models.

Table 2: Stock variations in training and test sets [25].

	Training Set	Test Set	Baseline
Stock	Standard Deviation	Standard Deviation	Baseline R^2
VALE3	0.0271	0.0158	0.8153
ITUB4	0.0166	0.0147	0.8842
BBDC4	0.0177	0.0149	0.9627
PETR3	0.0283	0.0149	0.8702
ABEV3	0.0119	0.0144	0.8935

After selecting the best hyperparameters, which is the subject of the next subsection, the more accurate LSTM network is selected as the final model. In order evaluate the generalization capabilities of the models, the test set, which contains data from 2019, is used to forecast the closing price of the next day (Approach 1) or the opening price of the next day (Approach 2). The results provided by the network are compared to the real values, which actually occurred. These comparisons are performed using Coefficient of Determination (R^2), Root Mean Squared Error (RMSE), and Mean Squared Error (MSE) metrics.

3.2 Hyperparameters Selection

In this work, the following hyperparameters were tuned: amount of layers (4 or 5), amount of LSTM units in each layer ([8, 16, 16, 32], [8, 16, 32, 32], [8, 16, 32, 64], [16, 32, 64, 128], [32, 32, 32, 32], [32, 64, 64, 128], [8, 8, 16, 32, 32], [8, 16, 32, 32, 64], [8, 16, 32, 64, 128], [16, 16, 32, 32, 64], [16, 16, 32, 64, 128], and [16, 32, 32, 64, 128]), activation functions (Rectified Linear Unit (ReLU) and hyperbolic tangent), and dropout (0, 0.2, 0.3, and 0.5). This tuning process was done through the grid search technique, explained in Subsection 2.4. Dropout values were used between 20% and 50%, only in the last layers, as suggested by Srivastava [26]. Regarding activation functions, tanh and ReLu were adopted as suggested by Greff [10]. The works developed by Greff [10] and Reimers and Gurevych [22] were taken as a basis for choosing the number of neurons and layers to be tested. In order to evaluate which set of parameters obtained the best performance, RMSE and R^2 were used to decide what were the best hyperparameters.

Grid search process consumes a huge amount of time, because many models must be trained. This way, some decisions were taken to reduce the scope of this hyperparameters search. In this sense, the following hyperparameters were constant: 16 to batch size, 100 to amount of epochs. Grid search was done to approach 1 using only PETR3 stock samples. This stock was selected because it has the highest standard deviation in the training set, according Table 2. We expect the higher standard deviation implies in higher variability and greatest amount of different patterns in the training set. Other stocks and approach 2 were tested only using the best hyperparameters found in the first scenario.

The hyperparameters which reach the top 5 accuracies can be found in Table 3. In the first column, Layer, the amount of layers in the architecture is declared. In the second column, named Neurons, the amount of LSTM units in each layer was presented. In the third column, Dropout, the Dropout ratio used in each layer was informed. In the forth column, there is the activation function. In the fifth column, the RMSE metric was presented. The sixth column contains the R^2 metric. As presented in Table 2, the R^2 accuracy baseline for PETR3 is 0.8702. All hyperparameters in top 5 ranking overcome the baseline. Due to time constraints, just the top 2 hyperparameters were evaluated.

Table 3: Top 5 hyperparameters found in the Grid Search based on PETR3 stock

Layers	Neurons	Dropout by layer	Activation Function	RMSE	R^2
4	[8, 16, 32, 64]	[0, 0, 0.2, 0.5]	tanh	0.0445	0.9550
5	[16, 32, 32, 64, 128]	[0, 0, 0.2, 0.3, 0.5]	tanh	0.0455	0.9530
5	[8, 16, 32, 64, 128]	[0, 0, 0, 0.2, 0.2]	relu	0.0458	0.9523
5	[8, 16, 32, 64, 128]	[0, 0, 0.2, 0.2, 0.2]	relu	0.0462	0.9515
5	[8, 16, 32, 32, 64]	[0, 0, 0, 0.2, 0.2]	relu	0.0462	0.9514

The network architecture for approach 1 is illustrated in Fig. 2. The final network contains four layers of LSTM units, where each layer contains respectively 8, 16, 32 and 64 LSTM units. Within the same layer, the units are connected to their adjacent, and the outputs of the units of one layer are completely connected with the inputs of the next layer. The network has dropout of 0%, 0%, 20%, and 50% to each layer, respectively. As a regression task, it means, the network output must be a single number, the network has a single neuron in the output layer. Activation function for this network is the hyperbolic tangent.

The network architecture for approach 2 is presented in Fig. 3. It has five layers of LSTM units, where each layer contains respectively 16, 32, 32, 64 and 128 LSTM units. The network has dropout for each layer is 0%, 0%, 20%, 30%, and 50% respectively. Activation function for this network is the hyperbolic tangent.

Fig. 2: Network Architecture for Approach 1.

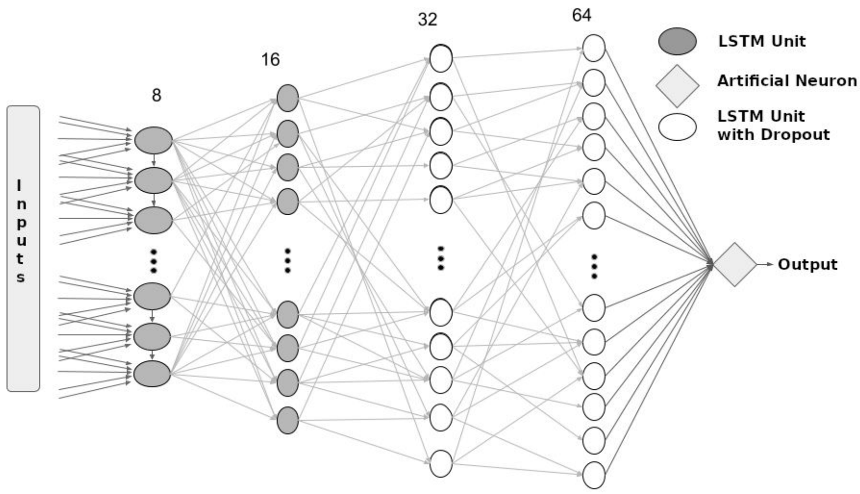
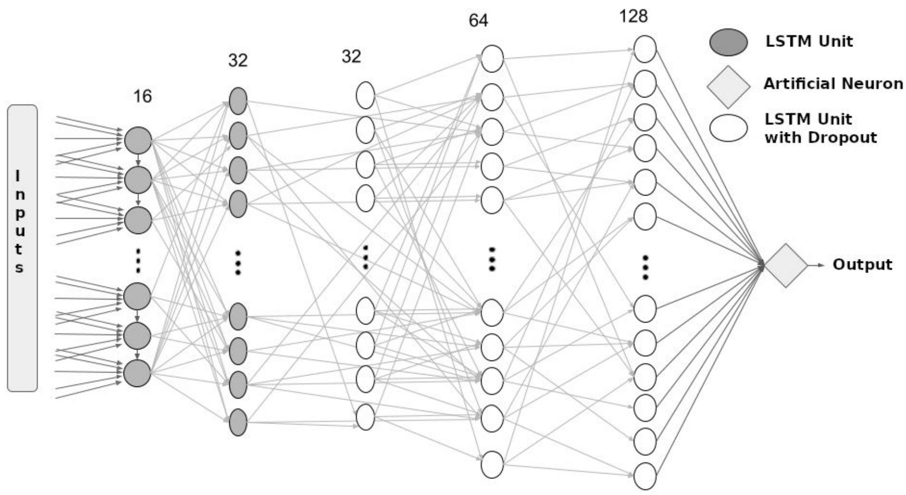


Fig. 3: Network Architecture for Approach 2.



3.3 Experimental Results

Using the network architectures presented in Subsection 2.4, the training and test proposed in Subsection 3.1 were performed. The best results can be found in Table 4 and they are graphically represented by the Figures(4, 5, 6, 7, 8). The Hyperbolic Tangent (tanh) activation function was used in all reported results.

Table 4: Testing results

Stock	Target	Neurons by Layer	Dropout	RMSE	MSE	R^2
PETR	Closing	[8, 16, 32, 64]	[0, 0, 0.2, 0.5]	0.0517	0.0026	0.9392
PETR	Opening	[8, 16, 32, 64]	[0, 0, 0.2, 0.5]	0.0575	0.0033	0.9119
VALE3	Closing	[8, 16, 32, 64]	[0, 0, 0.2, 0.5]	0.0504	0.0025	0.9363
VALE3	Opening	[16, 32, 32, 64, 128]	[0, 0, 0.2, 0.3, 0.5]	0.0496	0.0024	0.9311
ITUB4	Closing	[16, 32, 32, 64, 128]	[0, 0, 0.2, 0.3, 0.5]	0.0393	0.0015	0.9599
ITUB4	Opening	[8, 16, 32, 64]	[0, 0, 0.2, 0.5]	0.0528	0.0027	0.9433
BBDC	Closing	[16, 32, 32, 64, 128]	[0, 0, 0.2, 0.3, 0.5]	0.0311	0.0009	0.9871
BBDC	Opening	[16, 32, 32, 64, 128]	[0, 0, 0.2, 0.3, 0.5]	0.0274	0.0007	0.9902
ABEV	Closing	[16, 32, 32, 64, 128]	[0, 0, 0.2, 0.3, 0.5]	0.0402	0.0016	0.9598
ABEV	Opening	[8, 16, 32, 64]	[0, 0, 0.2, 0.5]	0.0434	0.0018	0.9405

Best results were reached with models trained and tested in BBDC stocks (Fig. 5) with R^2=0.99 in the opening price forecasting and R^2=0.98 in the closing price forecasting. Moreover, all models overcome the baselines presented in Table 2. It seems possible that there is a relationship between the stock prices standard deviation and the models accuracy. However the amount of stocks analyzed in this work does not allow conclusions with statistical significance.

Aiming to investigate if created models are able to forecast the price of other stocks that were not trained for, all models were trained with a stock and tested with other stocks. Forecasting results for opening prices are found in Table 6 and closing prices in Table 7.

3.4 Baseline Comparison

The best results in predicting the closing price were compared with the baseline, previously presented in Table 2. The improvement obtained is presented in Table 5. The best improvement was obtained in stock VALE which had the lowest baseline and also the lowest Test R^2. This observation suggest VALE as the hardest stock to predict in analysed context but the LSTM leads to an improvement around 15% regarding baseline. In the other hand, the lowest improvement was obtained in stock BBDC which had the highest baseline and also the highest Test R^2. This evidence suggest BBDC as the easiest stock to be predicted in analysed context. In this case, the LSTM model leads to an improvement around 2.5% regarding baseline.

4 Conclusions

In this work, a Long Short-Term Memory Recurrent Neural Network was proposed for recommending negotiations in Brazilian stock market. Decisions which led to the construction of the method were explained, as well as how the data was obtained and manipulated. Training for model building and test for model evaluation were explained. Hyperparameters tuning was examined and the results obtained by the proposed approach were presented and discussed.

Stock	Test R^2	Baseline	Improvement
PETR	0.9392	0.8702	0.0690
VALE	0.9363	0.8153	0.1210
ITUB	0.9599	0.8842	0.0757
BBDC	0.9871	0.9627	0.0244
ABEV	0.9598	0.8935	0.0663

Table 5: Improvements regarding Baseline

Obtained results show that forecasting and real values have a coefficient of determination (R^2) from 0.91 to 0.99, depending on the stock. In this sense, proposed approach consists in a viable proposal for predicting the behavior of stocks in the financial market. It is important to mention, the code and dataset used in this work are available at `https://github.com/Gabrielllopes/B3-LSTM`.

During this study, some possibilities for future works were identified. These possibilities could lead to improvements of the proposed approach, however they are computing intensive and need more time or computational resources to be pursuit. The possibilities found were:

– Fitting hyperparameters for each stock independently: in this work, hyperparameters fitting was done throught grid search using just PETR3 stock. Once defined, same hyperparameters were used to other stocks. In this sense, it can be investigated if a different set of hyperparameters to each stock can optimize the model accuracy.
– Expanding the amount of searched hyperparameters, inserting other hyperparameters such as batch size and learning rate.
– Expanding the range to each hyperparameter to encompass a broader search space.
– Regarding the tests, it can be used other stocks to validate the models proposed in this work, either using other Brazilian stocks or using stocks from other markets.

References

1. Atsalakis, G.S., Valavanis, K.P.: Surveying stock market forecasting techniques - part II: Soft computing methods. Expert Systems with Applications **36**(3), 5932–5941 (2009)
2. Bak, P., Paczuski, M., Shubik, M.: Price variations in a stock market with many agents. Physica A: Statistical Mechanics and its Applications **246**(3), 430–453 (1997). https://doi.org/https://doi.org/10.1016/S0378-4371(97)00401-9, `https://www.sciencedirect.com/science/article/pii/S0378437197004019`
3. Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for hyper-parameter optimization. In: Proceedings of the 24th International Conference on Neural Information Processing Systems. p. 2546–2554. NIPS'11, Curran Associates Inc., Red Hook, NY, USA (2011)

4. Bhandari, H.N., Rimal, B., Pokhrel, N.R., Rimal, R., Dahal, K.R., Khatri, R.K.: Predicting stock market index using LSTM. Machine Learning with Applications **9**, 100320 (2022). https://doi.org/https://doi.org/10.1016/j.mlwa.2022.100320, `https://www.sciencedirect.com/science/article/pii/S2666827022000378`
5. Dase, R., Pawar, D.: Application of artificial neural network for stock market predictions: A review of literature. International Journal of Machine Intelligence **2**(2), 14–17 (2010)
6. Edwards, R.D., Magee, J., Bassetti, W.C.: Technical analysis of stock trends. CRC press (2018), `https://books.google.com.br/books?id=MDgPEAAAQBAJ`
7. Farley, A.: 20 Rules for the Master Swing Trader. McGraw-Hill (2000)
8. Feurer, M., Springenberg, J.T., Hutter, F.: Initializing bayesian hyperparameter optimization via meta-learning. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
9. Gomes, F.R.: A bolsa de valores brasileira como fonte de informações financeiras. Perspectivas em ciência da informação **2**(2) (1997)
10. Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J.: LSTM: A search space odyssey. IEEE transactions on neural networks and learning systems **28**(10), 2222–2232 (2016)
11. Grima, S., Caruana, L.: The effect of the financial crisis on emerging markets. a comparative analysis of the stock market situation before and after. European Research Studies Journal **XX**, 427–453 (2017)
12. Hasan, M.M., Roy, P., Sarkar, S., Khan, M.M.: Stock market predictionweb service using deep learning by lstm. In: 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC). pp. 0180–0183 (2021). https://doi.org/10.1109/CCWC51732.2021.9375835
13. Hiransha, M., Gopalakrishnan, E., Menon, V.K., Soman, K.: NSE stock market prediction using deep-learning models. Procedia computer science **132**, 1351–1362 (2018)
14. Hsu, M.W., Lessmann, S., Sung, M.C., Ma, T., Johnson, J.E.: Bridging the divide in financial market forecasting: machine learners vs. financial economists. Expert Systems with Applications **61**, 215–234 (2016)
15. Li, Y., Li, L., Zhao, X., Ma, T., Zou, Y., Chen, M.: An attention-based LSTM model for stock price trend prediction using limit order books. Journal of Physics: Conference Series **1575**(1), 012124 (jun 2020). https://doi.org/10.1088/1742-6596/1575/1/012124
16. Menon, V.K., Vasireddy, N.C., Jami, S.A., Pedamallu, V.T.N., Sureshkumar, V., Soman, K.: Bulk price forecasting using spark over NSE data set. In: International Conference on Data Mining and Big Data. pp. 137–146. Springer (2016)
17. Mishkin, F.S.: The economics of money, banking and financial markets. Addison Wesley Longman (2004)
18. Moghar, A., Hamiche, M.: Stock market prediction using LSTM recurrent neural network. Procedia Computer Science **170**, 1168–1173 (2020). https://doi.org/https://doi.org/10.1016/j.procs.2020.03.049, `https://www.sciencedirect.com/science/article/pii/S1877050920304865`, the 11th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops
19. Pan, H.P.: A swingtum theory of intelligent finance for swing trading and momentum trading. In: Pan, H.P., Sornette, D., (Eds.), K. (eds.) Intelligent Finance - A Convergence of Financial Mathematics with Technical and Fundamental Analysis, pp. 451–475. University of Ballarat, Melbourne (2004)

20. Prodanov, C.C., de Freitas, E.C.: Metodologia do Trabalho Científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico - 2ª Edição. Editora Feevale (2013), https://books.google.com.br/books?id=zUDsAQAAQBAJ
21. Raschke, L.B., Connors, L.A.: Street Smarts: High Probability Short-Term Trading Strategies. M. Gordon Publishing Group (1996)
22. Reimers, N., Gurevych, I.: Optimal hyperparameters for deep LSTM-networks for sequence labeling tasks. arXiv preprint arXiv:1707.06799 (2017)
23. Rhea, R.: The Dow Theory. Igal Meirovich (2013), https://books.google.com.br/books?id=FO65ngEACAAJ
24. Sezer, O.B., Ozbayoglu, M., Dogdu, E.: A deep neural-network based stock trading system based on evolutionary optimized technical analysis parameters. Procedia computer science **114**, 473–480 (2017)
25. Silva, G.L.: Aplicando redes neurais recorrentes na predição de preços de ações da bolsa de valores. (2020), monografia (Bacharel em Engenharia de Computação), UNIPAMPA (Universidade Federal do Pampa), Bagé, Brazil
26. Srivastava, N.: Improving neural networks with dropout. University of Toronto **182**(566), 7 (2013)
27. Wen, Y., Lin, P., Nie, X.: Research of stock price prediction based on PCA-LSTM model. IOP Conference Series: Materials Science and Engineering **790**(1), 012109 (mar 2020). https://doi.org/10.1088/1757-899x/790/1/012109
28. Yadav, A., Jha, C.K., Sharan, A.: Optimizing LSTM for time series prediction in indian stock market. Procedia Computer Science **167**, 2091–2100 (2020). https://doi.org/https://doi.org/10.1016/j.procs.2020.03.257, international Conference on Computational Intelligence and Data Science
29. Zhang, G.P.: Time series forecasting using a hybrid arima and neural network model. Neurocomputing **50**, 159–175 (2003)

Appendix A Forecasting Results

Fig. 4: ABEV3 Forecasting using data from the year 2019.

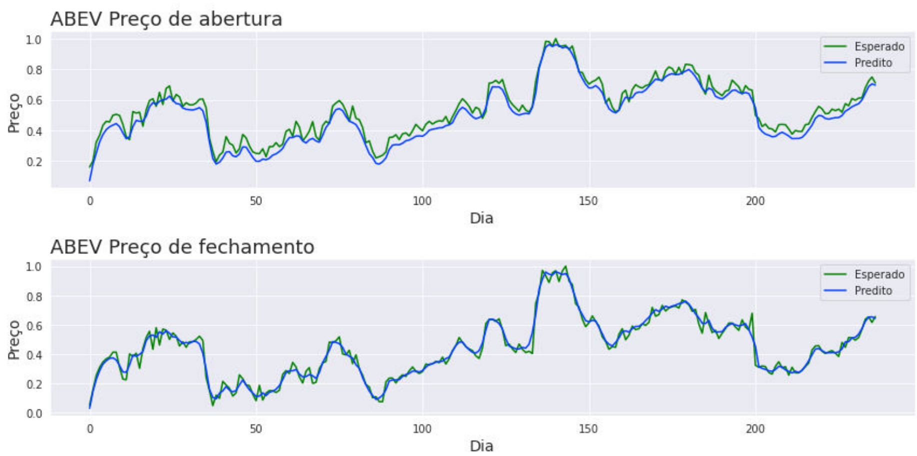


Fig. 5: BBDC4 Forecasting using data from the year 2019.

Fig. 6: ITUB4 Forecasting using data from the year 2019.

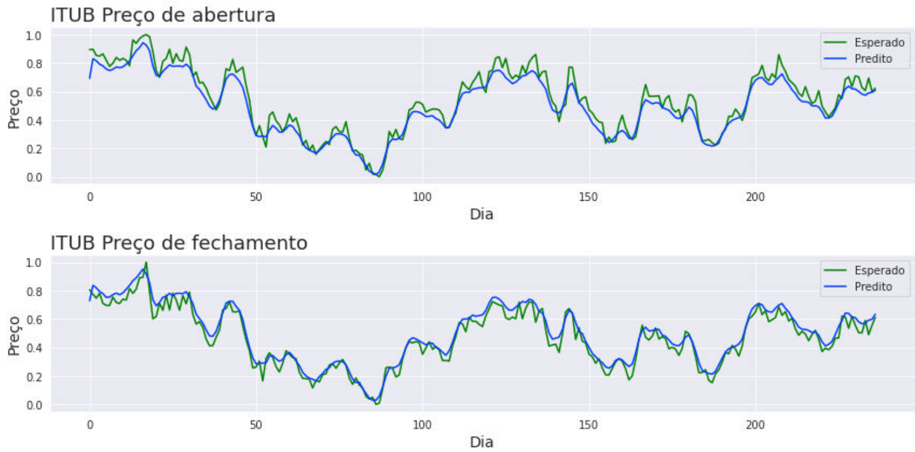


Fig. 7: PETR3 Forecasting using data from the year 2019.

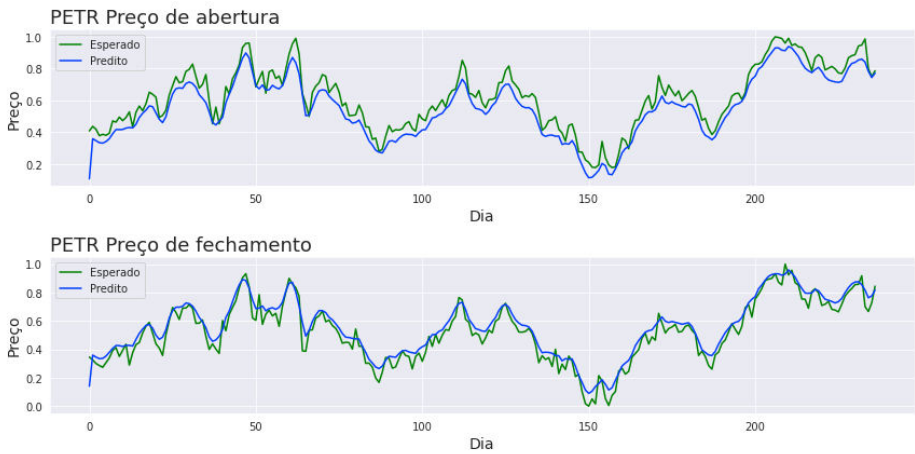


Fig. 8: VALE3 Forecasting using data from the year 2019.

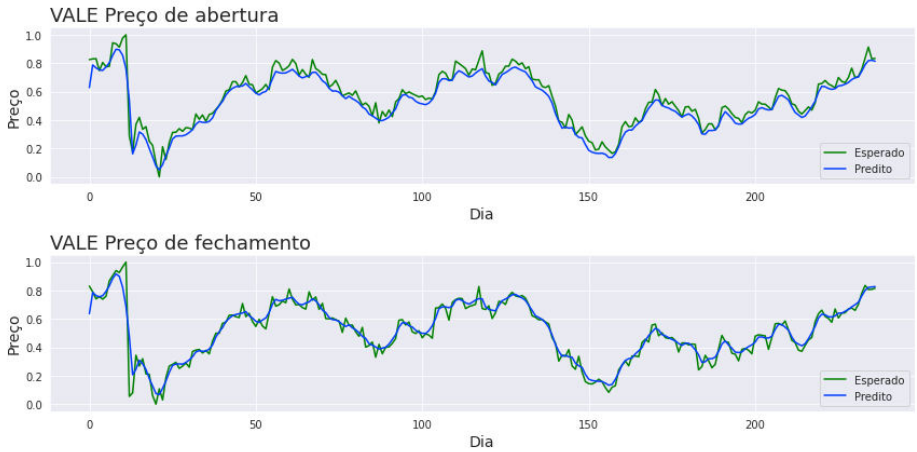


Table 6: Cross-forecasting for opening price using one stock for training and other stocks for testing.

Training	Test	RMSE	MSE	R^2
PETR3	VALE3	0.0498	0.0024	0.9306
PETR3	ITUB4	0.0468	0.0021	0.9555
PETR3	BBDC4	0.0274	0.0007	0.9902
PETR3	ABEV3	0.0543	0.0029	0.9067
VALE3	PETR3	0.0678	0.0046	0.8775
VALE3	ITUB4	0.0530	0.0028	0.9429
VALE3	BBDC4	0.0267	0.0007	0.9907
VALE3	ABEV3	0.0614	0.0037	0.8810
ITUB4	PETR3	0.0602	0.0036	0.9034
ITUB4	VALE3	0.0534	0.0028	0.9202
ITUB4	BBDC4	0.0281	0.0007	0.9897
ITUB4	ABEV3	0.0593	0.0035	0.8889
BBDC4	PETR3	0.0718	0.0051	0.8627
BBDC4	VALE3	0.0518	0.0026	0.9248
BBDC4	ITUB4	0.0707	0.0050	0.8987
BBDC4	ABEV3	0.0682	0.0046	0.8532
ABEV3	PETR3	0.0502	0.0025	0.9329
ABEV3	VALE3	0.0450	0.0020	0.9433
ABEV3	ITUB4	0.0468	0.0021	0.9556
ABEV3	BBDC4	0.0277	0.0007	0.9900

Table 7: Cross-forecasting for closing price using one stock for training and other stocks for testing.

Training	Test	RMSE	MSE	R^2
PETR3	VALE3	0.0524	0.0027	0.9313
PETR3	ITUB4	0.0484	0.0023	0.9394
PETR3	BBDC4	0.0330	0.0010	0.9854
PETR3	ABEV3	0.0461	0.0021	0.9472
VALE3	PETR3	0.0478	0.0022	0.9481
VALE3	ITUB4	0.0397	0.0015	0.9591
VALE3	BBDC4	0.0302	0.0009	0.9878
VALE3	ABEV3	0.0429	0.0018	0.9543
ITUB4	PETR3	0.0559	0.0031	0.9289
ITUB4	VALE3	0.0490	0.0024	0.9400
ITUB4	BBDC4	0.0324	0.0010	0.9859
ITUB4	ABEV3	0.0481	0.0023	0.9424
BBDC4	PETR3	0.0451	0.0020	0.9537
BBDC4	VALE3	0.0503	0.0025	0.9366
BBDC4	ITUB4	0.0386	0.0014	0.9613
BBDC4	ABEV3	0.0368	0.0013	0.9663
ABEV3	PETR3	0.0489	0.0023	0.9457
ABEV3	VALE3	0.0497	0.0024	0.9383
ABEV3	ITUB4	0.0404	0.0016	0.9577
ABEV3	BBDC4	0.0328	0.0010	0.9856